



**EBook Gratuito**

# APPENDIMENTO

---

# CSS

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#CSS**

# Sommario

|  |           |
|--|-----------|
| Di.....  | 1         |
| <b>Capitolo 1: Iniziare con i CSS</b> .....                                    | <b>2</b>  |
| Osservazioni.....  | 2         |
| Versioni.....  | 2         |
| Examples.....  | 2         |
| Foglio di stile esterno.....   | 2         |
| <b>Esempio</b> .....   | <b>2</b>  |
| Stili interni.....   | 4         |
| Stili in linea.....  | 4         |
| Regola @import CSS (una delle regole at-CSS).....                              | 5         |
| Come usare @import.....  | 5         |
| Cambiare CSS con JavaScript.....   | 5         |
| JavaScript puro.....   | 5         |
| jQuery.....  | 6         |
| <b>Guarda anche</b> .....  | <b>6</b>  |
| Elenchi di stili con CSS.....  | 6         |
| <b>Capitolo 2: Adattamento e posizionamento dell'oggetto</b> .....             | <b>8</b>  |
| Osservazioni.....  | 8         |
| Examples.....  | 8         |
| oggetto-fit.....   | 8         |
| <b>Capitolo 3: animazioni</b> .....  | <b>11</b> |
| Sintassi.....  | 11        |
| Parametri.....   | 11        |
| Examples.....  | 11        |
| Animazioni con la proprietà di transizione.....                                | 11        |
| <b>Esempio</b> .....   | <b>11</b> |
| <b>Compatibilità tra browser</b> .....   | <b>12</b> |
| Aumentare le prestazioni dell'animazione usando l'attributo `will-change`..... | 13        |
| Animazioni con fotogrammi chiave.....  | 13        |

|  |           |
|--|-----------|
| Esempio di base.....   | 13        |
| Compatibilità cross-browser.....   | 15        |
| Esempi di sintassi.....  | 15        |
| <b>Capitolo 4: Blocca i contesti di formattazione.....</b>                   | <b>17</b> |
| Osservazioni.....  | 17        |
| Examples.....  | 17        |
| Utilizzando la proprietà di overflow con un valore diverso da visibile.....  | 17        |
| <b>Capitolo 5: box-ombra.....</b>  | <b>19</b> |
| Sintassi.....  | 19        |
| Parametri.....   | 19        |
| Osservazioni.....  | 19        |
| Examples.....  | 19        |
| drop shadow.....   | 19        |
| ombra esterna.....   | 20        |
| ombreggiatura di solo livello inferiore utilizzando uno pseudo-elemento..... | 20        |
| più ombre.....   | 21        |
| <b>Capitolo 6: Cascading e Specificità.....</b>                              | <b>23</b> |
| Osservazioni.....  | 23        |
| Examples.....  | 23        |
| Cascading.....   | 23        |
| <b>Ordine di caricamento CSS.....</b>  | <b>23</b> |
| <b>Come vengono risolti i conflitti?.....</b>                                | <b>23</b> |
| Esempio 1 - Regole di specificità.....                                       | 23        |
| Esempio 2 - Regole in cascata con selettori identici.....                    | 24        |
| Esempio 3: regole a cascata dopo le regole di specificità.....               | 24        |
| Un'ultima nota.....  | 24        |
| La! Dichiarazione importante.....  | 25        |
| Calcolo della specificità del selettore.....                                 | 25        |
| Esempio 1: specificità delle varie sequenze di selettori.....                | 26        |
| Esempio 2: come viene utilizzata la specificità dal browser.....             | 26        |
| Esempio 3: come manipolare la specificità.....                               | 27        |
| !important Dichiarazioni di stile !important e in linea.....                 | 28        |

|   |           |
|---|-----------|
| Un'ultima nota.....   | 28        |
| Esempio di specificità più complesso.....                                       | 28        |
| <b>Capitolo 7: Centraggio.....</b>  | <b>30</b> |
| Examples.....   | 30        |
| Usando la trasformazione CSS.....   | 30        |
| COMPATIBILITÀ DEL BROWSER CROSS.....  | 30        |
| MAGGIORI INFORMAZIONI.....  | 31        |
| Utilizzando Flexbox.....  | 31        |
| Usando la posizione: assoluta.....  | 32        |
| Tecnica dell'elemento fantasma (hack di Micha Czerow).....                      | 33        |
| Utilizzando testo allineato.....  | 33        |
| Centraggio in relazione ad un altro oggetto.....                                | 34        |
| Allineare verticalmente qualsiasi cosa con 3 linee di codice.....               | 35        |
| Allinea verticalmente un'immagine all'interno div.....                          | 35        |
| Centrata orizzontale e verticale usando la disposizione della tabella.....      | 36        |
| Utilizzo di calc ().....  | 36        |
| Allinea verticalmente gli elementi di altezza dinamica.....                     | 37        |
| Utilizzando l'altezza della linea.....  | 38        |
| Centrata verticale e orizzontale senza preoccuparsi di altezza o larghezza..... | 38        |
| Il contenitore esterno.....   | 38        |
| Il contenitore interno.....   | 38        |
| La scatola dei contenuti.....   | 38        |
| dimostrazione.....  | 39        |
| Centrata con dimensioni fisse.....  | 39        |
| Centrata orizzontale con larghezza fissa.....                                   | 40        |
| Centrata verticale con altezza fissa.....                                       | 40        |
| Utilizzo del margine: 0 auto;.....  | 41        |
| <b>Capitolo 8: Centrata verticale.....</b>                                      | <b>43</b> |
| Osservazioni.....   | 43        |
| Examples.....   | 43        |
| Centrata con display: tabella.....  | 43        |
| Centrata con Trasforma.....   | 43        |

|   |           |
|---|-----------|
| Centraggio con Flexbox.....                       | 44        |
| Centrata del testo con l'altezza della linea..... | 44        |
| Centrata con posizione: assoluta.....             | 45        |
| Centrata con pseudo elemento.....                 | 46        |
| <b>Capitolo 9: colonne.....</b>                   | <b>47</b> |
| Sintassi.....                                     | 47        |
| Examples.....                                     | 47        |
| Esempio semplice (numero di colonne).....         | 47        |
| Larghezza della colonna.....                      | 48        |
| <b>Capitolo 10: Colonne multiple.....</b>         | <b>50</b> |
| introduzione.....                                 | 50        |
| Osservazioni.....                                 | 50        |
| Examples.....                                     | 50        |
| Esempio di base.....                              | 50        |
| Crea più colonne.....                             | 51        |
| <b>Capitolo 11: Colori.....</b>                   | <b>52</b> |
| Sintassi.....                                     | 52        |
| Examples.....                                     | 52        |
| Parole chiave di colore.....                      | 52        |
| <b>Parole chiave di colore.....</b>               | <b>52</b> |
| Valore esadecimale.....                           | 59        |
| sfondo.....                                       | 59        |
| Sintassi.....                                     | 60        |
| rgb () Notazione.....                             | 60        |
| Sintassi.....                                     | 61        |
| hsl () Notazione.....                             | 61        |
| <b>Sintassi.....</b>                              | <b>61</b> |
| <b>Gli appunti.....</b>                           | <b>62</b> |
| currentColor.....                                 | 62        |
| Utilizzare nello stesso elemento.....             | 62        |
| Ereditato dall'elemento padre.....                | 62        |

|   |           |
|---|-----------|
| rgba () Notazione.....  | 63        |
| Sintassi.....   | 64        |
| hsla () Notazione.....  | 64        |
| <b>Sintassi.....</b>  | <b>64</b> |
| <b>Capitolo 12: Commenti.....</b>                               | <b>65</b> |
| Sintassi.....   | 65        |
| Osservazioni.....   | 65        |
| Examples.....   | 65        |
| Linea singola.....  | 65        |
| Linea multipla.....   | 65        |
| <b>Capitolo 13: Confine.....</b>                                | <b>66</b> |
| Sintassi.....   | 66        |
| Osservazioni.....   | 66        |
| Examples.....   | 68        |
| border-radius.....  | 68        |
| stile del bordo.....  | 69        |
| confine (stenografia).....                                      | 70        |
| border-immagine.....  | 70        |
| border-[left   right   top   bottom].....                       | 71        |
| border-collapse.....  | 71        |
| Bordi multipli.....   | 71        |
| Creare un bordo multicolore usando l'immagine del bordo.....    | 73        |
| <b>CSS.....</b>   | <b>73</b> |
| <b>HTML.....</b>  | <b>73</b> |
| <b>Capitolo 14: contatori.....</b>                              | <b>75</b> |
| Sintassi.....   | 75        |
| Parametri.....  | 75        |
| Osservazioni.....   | 75        |
| Examples.....   | 76        |
| Applicando lo stile dei numeri romani all'uscita contatore..... | 76        |
| <b>CSS.....</b>   | <b>76</b> |

|   |           |
|---|-----------|
| <b>HTML</b> .....   | <b>76</b> |
| Numera ogni oggetto usando il contatore CSS.....                                | 76        |
| <b>CSS</b> .....  | <b>76</b> |
| <b>HTML</b> .....   | <b>77</b> |
| Implementazione della numerazione multilivello utilizzando i contatori CSS..... | 77        |
| <b>CSS</b> .....  | <b>77</b> |
| <b>HTML</b> .....   | <b>77</b> |
| <b>Capitolo 15: Contesto impilabile</b> .....                                   | <b>79</b> |
| Examples.....   | 79        |
| Contesto impilabile.....  | 79        |
| <b>Capitolo 16: Controllo del layout</b> .....                                  | <b>83</b> |
| Sintassi.....   | 83        |
| Parametri.....  | 83        |
| Examples.....   | 84        |
| La proprietà di visualizzazione.....  | 84        |
| <b>In linea</b> .....   | <b>84</b> |
| <b>Bloccare</b> .....   | <b>84</b> |
| <b>Blocco in linea</b> .....  | <b>84</b> |
| <b>nessuna</b> .....  | <b>86</b> |
| Per ottenere la vecchia struttura della tabella utilizzando div.....            | 87        |
| <b>Capitolo 17: CSS Image Sprites</b> .....                                     | <b>88</b> |
| Sintassi.....   | 88        |
| Osservazioni.....   | 88        |
| Examples.....   | 88        |
| Un'implementazione di base.....   | 88        |
| <b>Capitolo 18: CSS Object Model (CSSOM)</b> .....                              | <b>90</b> |
| Osservazioni.....   | 90        |
| Examples.....   | 90        |
| introduzione.....   | 90        |
| Aggiunta di una regola per l'immagine di sfondo tramite CSSOM.....              | 90        |

|   |            |
|---|------------|
| <b>Capitolo 19: Corsore Styling</b> .....           | <b>92</b>  |
| Sintassi.....                                       | 92         |
| Examples.....                                       | 92         |
| Cambiare il tipo di cursore.....                    | 92         |
| pointer-events.....                                 | 93         |
| caret-color.....                                    | 93         |
| <b>Capitolo 20: Elenca stili</b> .....              | <b>94</b>  |
| Sintassi.....                                       | 94         |
| Parametri.....                                      | 94         |
| Osservazioni.....                                   | 94         |
| Examples.....                                       | 94         |
| Tipo di pallottola o numerazione.....               | 94         |
| Posizione di proiettile.....                        | 95         |
| Rimozione di elenchi puntati / numeri.....          | 95         |
| <b>Capitolo 21: Eredità</b> .....                   | <b>97</b>  |
| Sintassi.....                                       | 97         |
| Examples.....                                       | 97         |
| Ereditarietà automatica.....                        | 97         |
| Ereditarietà forzata.....                           | 97         |
| <b>Capitolo 22: Forme ad elementi singoli</b> ..... | <b>99</b>  |
| Examples.....                                       | 99         |
| Piazza.....   | 99         |
| triangoli.....                                      | 99         |
| Bursts.....   | 103        |
| Cerchi ed ellissi.....                              | 104        |
| <b>Cerchio</b> .....                                | <b>104</b> |
| <b>Ellisse</b> .....                                | <b>105</b> |
| trapezio.....                                       | 105        |
| Cubo.....   | 106        |
| Piramide.....                                       | 107        |
| <b>Capitolo 23: Forme per galleggianti</b> .....    | <b>109</b> |

|  |            |
|--|------------|
| Sintassi.....  | 109        |
| Parametri.....   | 109        |
| Osservazioni.....  | 109        |
| Examples.....  | 109        |
| Forma all'esterno con forma base - cerchio ().....   | 109        |
| Margine di forma.....  | 111        |
| <b>Capitolo 24: funzioni.....</b>  | <b>113</b> |
| Sintassi.....  | 113        |
| Osservazioni.....  | 113        |
| Examples.....  | 113        |
| funzione calc ().....  | 113        |
| funzione attr ().....  | 114        |
| funzione linear-gradient ().....   | 114        |
| funzione radial-gradient ().....   | 114        |
| funzione var ().....   | 114        |
| <b>Capitolo 25: Galleggianti.....</b>  | <b>116</b> |
| Sintassi.....  | 116        |
| Osservazioni.....  | 116        |
| Examples.....  | 116        |
| Galleggiare un'immagine all'interno del testo.....   | 116        |
| Semplice layout a due colonne a larghezza fissa.....                                       | 117        |
| Semplice layout a tre colonne a larghezza fissa.....                                       | 118        |
| Layout pigro / avido a due colonne.....  | 119        |
| chiara proprietà.....  | 120        |
| clearfix.....  | 121        |
| <b>Clearfix (con il margine superiore di compressione dei float ancora presenti).....</b>  | <b>121</b> |
| <b>Clearfix impedisce anche il collasso dei margini superiori dei float contenuti.....</b> | <b>121</b> |
| <b>Clearfix con supporto di browser obsoleti IE6 e IE7.....</b>                            | <b>122</b> |
| DIV in linea usando float.....   | 122        |
| Uso della proprietà di overflow per cancellare i float.....                                | 124        |
| <b>Capitolo 26: Griglia.....</b>   | <b>125</b> |

|   |            |
|---|------------|
| introduzione.....   | 125        |
| Osservazioni.....   | 125        |
| Examples.....   | 125        |
| Esempio di base.....  | 125        |
| <b>Capitolo 27: Hack di Internet Explorer.....</b>                            | <b>127</b> |
| Osservazioni.....   | 127        |
| Examples.....   | 127        |
| Modalità Contrasto elevato in Internet Explorer 10 e versioni successive..... | 127        |
| Esempi.....   | 127        |
| Maggiori informazioni:.....   | 127        |
| Solo Internet Explorer 6 e Internet Explorer 7.....                           | 128        |
| Solo Internet Explorer 8.....   | 128        |
| Aggiunta del supporto Inline Block a IE6 e IE7.....                           | 128        |
| <b>Capitolo 28: I selettori.....</b>  | <b>129</b> |
| introduzione.....   | 129        |
| Sintassi.....   | 129        |
| Osservazioni.....   | 129        |
| Examples.....   | 129        |
| Selettori di attributi.....   | 129        |
| <b>Panoramica.....</b>  | <b>129</b> |
| <b>Dettagli.....</b>  | <b>130</b> |
| [attribute].....  | 130        |
| [attribute="value"].....  | 131        |
| [attribute*="value"].....   | 131        |
| [attribute~="value"].....   | 131        |
| [attribute^="value"].....   | 132        |
| [attribute\$="value"].....  | 132        |
| [attribute "value"].....  | 132        |
| [attribute="value" i].....  | 132        |
| <b>Specificità dei selettori di attributi.....</b>                            | <b>133</b> |
| 0-1-0.....  | 133        |

|  |            |
|--|------------|
| combinatori.....   | 133        |
| <b>Panoramica.....</b>   | <b>133</b> |
| <b>Combinatore discendente: selector selector.....</b>                               | <b>134</b> |
| <b>Combinatore di bambini: selector &gt; selector.....</b>                           | <b>134</b> |
| <b>Combinatore Sibling adiacente: selector + selector.....</b>                       | <b>134</b> |
| <b>General Sibling Combinator: selector ~ selector.....</b>                          | <b>135</b> |
| Selettori del nome di classe.....  | 135        |
| Selettori ID.....  | 136        |
| Pseudo-classi.....   | 137        |
| Sintassi.....  | 137        |
| Elenco di pseudo-classi:.....  | 137        |
| Selettori di base.....   | 140        |
| Come plasmare un input Range.....  | 141        |
| Global boolean con checkbox: checked e ~ (combinatore di fratellanza generale).....  | 141        |
| <b>Aggiungi booleano come casella di controllo.....</b>                              | <b>141</b> |
| <b>Cambia il valore del booleano.....</b>  | <b>142</b> |
| <b>Accesso al valore booleano con CSS.....</b>                                       | <b>142</b> |
| <b>In azione.....</b>  | <b>142</b> |
| CSS3: esempio di selettore all'interno del campo.....                                | 142        |
| Pseudo classe bambino.....   | 143        |
| Seleziona l'elemento usando il suo ID senza l'alta specificità del selettore ID..... | 143        |
| A. L'esempio: non pseudo-classe e B.: focus-all'interno della pseudo-classe CSS..... | 144        |
| L'esempio del selettore pseudo-classe: only-child.....                               | 146        |
| Il: ultimo tipo di selettore.....  | 146        |
| <b>Capitolo 29: Il modello di scatola.....</b>                                       | <b>148</b> |
| Sintassi.....  | 148        |
| Parametri.....   | 148        |
| Osservazioni.....  | 148        |
| <b>A proposito di padding-box.....</b>   | <b>148</b> |
| Examples.....  | 148        |
| Qual è il modello di scatola?.....   | 148        |

|   |            |
|---|------------|
| <b>I bordi</b> .....  | <b>148</b> |
| <b>Esempio</b> .....  | <b>149</b> |
| box-sizing.....   | 151        |
| <b>Capitolo 30: Imbottitura</b> .....                                   | <b>153</b> |
| Sintassi.....   | 153        |
| Osservazioni.....   | 153        |
| Examples.....   | 153        |
| Imbottitura su un lato.....   | 153        |
| Stenografia di imbottitura.....   | 154        |
| <b>Capitolo 31: La frammentazione</b> .....                             | <b>156</b> |
| Sintassi.....   | 156        |
| Parametri.....  | 156        |
| Osservazioni.....   | 156        |
| Examples.....   | 156        |
| Media stampa page-break.....  | 156        |
| <b>Capitolo 32: Layout di blocco in linea</b> .....                     | <b>158</b> |
| Examples.....   | 158        |
| Barra di navigazione giustificata.....                                  | 158        |
| <b>HTML</b> .....   | <b>158</b> |
| <b>CSS</b> .....  | <b>158</b> |
| <b>Gli appunti</b> .....  | <b>158</b> |
| <b>Capitolo 33: Layout flessibile della scatola (Flexbox)</b> .....     | <b>160</b> |
| introduzione.....   | 160        |
| Sintassi.....   | 160        |
| Osservazioni.....   | 160        |
| <b>Prefissi di Vender</b> .....   | <b>160</b> |
| <b>risorse</b> .....  | <b>160</b> |
| Examples.....   | 161        |
| Piè di pagina appiccicoso a altezza variabile.....                      | 161        |
| Layout di Holy Grail usando Flexbox.....                                | 162        |
| Pulsanti perfettamente allineati all'interno di schede con flexbox..... | 163        |

|   |            |
|---|------------|
| Centrata verticale e orizzontale dinamica (allineare elementi, giustificare-contenuto)..... | 165        |
| <b>Esempio semplice (centrata di un singolo elemento).....</b>                              | <b>165</b> |
| HTML.....   | 165        |
| CSS.....  | 165        |
| <b>Ragionamento.....</b>  | <b>165</b> |
| <b>Esempi di proprietà individuali.....</b>   | <b>166</b> |
| Esempio: justify-content: center su un flexbox orizzontale.....                             | 166        |
| Esempio: justify-content: center su un flexbox verticale.....                               | 167        |
| Esempio: align-content: center su una flexbox orizzontale.....                              | 168        |
| Esempio: align-content: center su una flexbox verticale.....                                | 169        |
| Esempio: combinazione per il centraggio sia sulla flexbox orizzontale.....                  | 170        |
| Esempio: combinazione per il centraggio sia sulla flexbox verticale.....                    | 171        |
| Stessa altezza sui contenitori nidificati.....  | 172        |
| Adatta in modo ottimale gli elementi al loro contenitore.....                               | 173        |
| <b>Capitolo 34: Lineamenti.....</b>   | <b>175</b> |
| Sintassi.....   | 175        |
| Parametri.....  | 175        |
| Osservazioni.....   | 175        |
| Examples.....   | 176        |
| Panoramica.....   | 176        |
| outline-style.....  | 176        |
| <b>Capitolo 35: margini.....</b>  | <b>178</b> |
| Sintassi.....   | 178        |
| Parametri.....  | 178        |
| Osservazioni.....   | 178        |
| Examples.....   | 178        |
| Applica margine su un dato lato.....  | 178        |
| <b>Proprietà specifiche della direzione.....</b>  | <b>178</b> |
| <b>Specifica della direzione utilizzando la proprietà di stenografia.....</b>               | <b>179</b> |
| Margine che crolla.....   | 180        |
| Centra orizzontalmente gli elementi su una pagina usando il margine.....                    | 182        |

|  |            |
|--|------------|
| Semplificazione della proprietà del margine.....   | 182        |
| Margini negativi.....  | 183        |
| Esempio 1:.....  | 183        |
| <b>Capitolo 36: Media Queries.....</b>   | <b>185</b> |
| Sintassi.....  | 185        |
| Parametri.....   | 185        |
| Osservazioni.....  | 186        |
| Examples.....  | 187        |
| Esempio di base.....   | 187        |
| Utilizzare sul tag link.....   | 187        |
| tipo di supporto.....  | 188        |
| Utilizzo di query multimediali per scegliere come target dimensioni schermo diverse.....               | 189        |
| Larghezza vs Viewport.....   | 189        |
| Media Queries per Retina e Non Retina Screens.....   | 190        |
| Terminologia e struttura.....  | 190        |
| <b>Struttura generale di una query multimediale.....</b>   | <b>191</b> |
| <b>Una query multimediale contenente un tipo di media.....</b>   | <b>191</b> |
| <b>Una query multimediale contenente un tipo di supporto e una funzione multimediale.....</b>          | <b>191</b> |
| <b>Una query multimediale contenente una funzione multimediale (e un tipo di media implicito).....</b> | <b>191</b> |
| Media query e IE8.....   | 191        |
| Una soluzione alternativa basata su Javascript.....  | 192        |
| L'alternativa.....   | 192        |
| <b>Capitolo 37: Modelli di progettazione CSS.....</b>  | <b>193</b> |
| introduzione.....  | 193        |
| Osservazioni.....  | 193        |
| Examples.....  | 193        |
| BEM.....   | 193        |
| <b>Esempio di codice.....</b>  | <b>194</b> |
| <b>Capitolo 38: Normalizzazione degli stili del browser.....</b>                                       | <b>195</b> |
| introduzione.....  | 195        |
| Osservazioni.....  | 195        |

|  |            |
|--|------------|
| Examples.....  | 195        |
| normalize.css.....   | 195        |
| Che cosa fa.....   | 195        |
| Differenza di reset.css.....   | 196        |
| Approcci ed esempi.....  | 196        |
| <b>Capitolo 39: Opacità.....</b>   | <b>198</b> |
| Sintassi.....  | 198        |
| Osservazioni.....  | 198        |
| Examples.....  | 198        |
| Proprietà Opacità.....   | 198        |
| Compatibilità IE per `opacity`.....                                      | 198        |
| <b>Capitolo 40: Posizionamento.....</b>                                  | <b>200</b> |
| Sintassi.....  | 200        |
| Parametri.....   | 200        |
| Osservazioni.....  | 200        |
| Examples.....  | 200        |
| Posizione fissa.....   | 200        |
| Elementi sovrapposti con z-index.....                                    | 201        |
| <b>Esempio.....</b>  | <b>201</b> |
| HTML.....  | 201        |
| CSS.....   | 201        |
| <b>Sintassi.....</b>   | <b>202</b> |
| <b>Osservazioni.....</b>   | <b>202</b> |
| Posizione relativa.....  | 203        |
| Posizione assoluta.....  | 203        |
| Posizionamento statico.....  | 204        |
| <b>Capitolo 41: Prestazione.....</b>                                     | <b>205</b> |
| Examples.....  | 205        |
| Usa la trasformazione e l'opacità per evitare il layout del trigger..... | 205        |
| <b>NON.....</b>  | <b>205</b> |
| <b>FARE.....</b>   | <b>206</b> |

|  |            |
|--|------------|
| <b>Capitolo 42: Proprietà filtro</b>                     | <b>207</b> |
| Sintassi   | 207        |
| Parametri  | 207        |
| Osservazioni   | 208        |
| Examples   | 208        |
| Ombra esterna (usa invece box-shadow se possibile)       | 208        |
| Più valori di filtro                                     | 208        |
| Tonalità Ruota   | 209        |
| Inverti colore   | 210        |
| blur   | 210        |
| <b>Capitolo 43: Proprietà personalizzate (variabili)</b> | <b>212</b> |
| introduzione   | 212        |
| Sintassi   | 212        |
| Osservazioni   | 212        |
| SUPPORTO AL BROWSER / COMPATIBILITÀ                      | 212        |
| Examples   | 213        |
| Colore variabile   | 213        |
| Dimensioni variabili                                     | 213        |
| Cascading variabile                                      | 213        |
| Validi / invalidi  | 214        |
| Con le domande dei media                                 | 215        |
| <b>Capitolo 44: Pseudo-Elementi</b>                      | <b>218</b> |
| introduzione   | 218        |
| Sintassi   | 218        |
| Parametri  | 218        |
| Osservazioni   | 219        |
| Examples   | 219        |
| Pseudo-Elementi  | 219        |
| Pseudo-elementi nelle liste                              | 219        |
| <b>Capitolo 45: Query di caratteristiche</b>             | <b>221</b> |
| Sintassi   | 221        |
| Parametri  | 221        |

|   |            |
|---|------------|
| Osservazioni.....   | 221        |
| Examples.....   | 221        |
| L'utilizzo di base @supports.....                                       | 221        |
| Rilevazione delle feature di concatenazione.....                        | 221        |
| <b>Capitolo 46: Ritaglio e mascheratura.....</b>                        | <b>223</b> |
| Sintassi.....   | 223        |
| Parametri.....  | 223        |
| Osservazioni.....   | 224        |
| <b>Maschere:.....</b>   | <b>224</b> |
| <b>Clip-path:.....</b>  | <b>225</b> |
| Examples.....   | 225        |
| Ritaglio (poligono).....  | 225        |
| <b>CSS:.....</b>  | <b>225</b> |
| <b>HTML:.....</b>   | <b>225</b> |
| Ritaglio (Cerchio).....   | 226        |
| <b>CSS:.....</b>  | <b>226</b> |
| <b>HTML.....</b>  | <b>226</b> |
| Ritaglio e mascheratura: panoramica e differenza.....                   | 227        |
| Ritaglio.....   | 227        |
| mascheramento.....  | 227        |
| Maschera semplice che dissolve un'immagine da solida a trasparente..... | 228        |
| <b>CSS.....</b>   | <b>228</b> |
| <b>HTML.....</b>  | <b>228</b> |
| Utilizzo di maschere per tagliare un buco nel mezzo di un'immagine..... | 229        |
| <b>CSS.....</b>   | <b>229</b> |
| <b>HTML.....</b>  | <b>229</b> |
| Utilizzo di maschere per creare immagini con forme irregolari.....      | 230        |
| <b>CSS.....</b>   | <b>230</b> |
| <b>HTML.....</b>  | <b>230</b> |
| <b>Capitolo 47: sfondi.....</b>   | <b>232</b> |

|   |            |
|---|------------|
| introduzione.....                                   | 232        |
| Sintassi.....                                       | 232        |
| Osservazioni.....                                   | 232        |
| Examples.....                                       | 232        |
| Colore di sfondo.....                               | 232        |
| Nomi di colori.....                                 | 233        |
| Codici colore esadecimali.....                      | 233        |
| RGB / RGBa.....                                     | 233        |
| HSL / HSLa.....                                     | 234        |
| Interazione con l'immagine di sfondo.....           | 234        |
| Immagine di sfondo.....                             | 235        |
| Gradienti di sfondo.....                            | 236        |
| <b>lineare-gradiente ().....</b>                    | <b>236</b> |
| <b>radiale gradiente ().....</b>                    | <b>237</b> |
| <b>Ripetendo le sfumature.....</b>                  | <b>237</b> |
| Stenografia di base.....                            | 238        |
| Sintassi.....                                       | 239        |
| Esempi.....   | 239        |
| Posizione di sfondo.....                            | 239        |
| <b>Proprietà posizione sfondo a mano lunga.....</b> | <b>240</b> |
| Allegato sfondo.....                                | 240        |
| <b>Esempi.....</b>                                  | <b>241</b> |
| background-attachment: scroll.....                  | 241        |
| background-attachment: fixed.....                   | 241        |
| background-attachment: local.....                   | 241        |
| Ripeti lo sfondo.....                               | 241        |
| Colore di sfondo con opacità.....                   | 242        |
| Immagine di sfondo multipla.....                    | 243        |
| La proprietà background-origin.....                 | 243        |
| Clip di sfondo.....                                 | 245        |
| Dimensione dello sfondo.....                        | 246        |

|   |            |
|---|------------|
| <b>Panoramica generale</b> .....                                      | <b>246</b> |
| <b>Mantenere le proporzioni</b> .....                                 | <b>248</b> |
| Eggsplanation per contain e cover.....                                | 248        |
| contain.....  | 249        |
| cover.....  | 249        |
| Dimostrazione con codice reale.....                                   | 250        |
| Proprietà background-blend-mode.....                                  | 251        |
| <b>Capitolo 48: straripamento</b> .....                               | <b>252</b> |
| Sintassi.....   | 252        |
| Parametri.....  | 252        |
| Osservazioni.....   | 252        |
| Examples.....   | 252        |
| overflow: scorrere.....   | 253        |
| trabocco-wrap.....  | 253        |
| overflow: visibile.....   | 254        |
| Contesto di formattazione dei blocchi creato con overflow.....        | 255        |
| overflow-x e overflow-y.....  | 256        |
| <b>Capitolo 49: Struttura e formattazione di una regola CSS</b> ..... | <b>258</b> |
| Osservazioni.....   | 258        |
| <b>Buono</b> .....  | <b>258</b> |
| <b>Male</b> .....   | <b>258</b> |
| <b>One-Liner</b> .....  | <b>258</b> |
| Examples.....   | 258        |
| Regole, selettori e blocchi di dichiarazione.....                     | 258        |
| Elenchi di proprietà.....   | 258        |
| Selettori multipli.....   | 259        |
| <b>Capitolo 50: Supporto browser e prefissi</b> .....                 | <b>260</b> |
| Parametri.....  | 260        |
| Osservazioni.....   | 260        |
| Examples.....   | 260        |
| transizioni.....  | 261        |

|                                       |            |
|---------------------------------------|------------|
| Trasformare.....                      | 261        |
| <b>Capitolo 51: tabelle.....</b>      | <b>262</b> |
| Sintassi.....                         | 262        |
| Osservazioni.....                     | 262        |
| Examples.....                         | 262        |
| table-layout.....                     | 262        |
| border-collapse.....                  | 263        |
| border-spacing.....                   | 263        |
| empty-cells.....                      | 264        |
| caption-side.....                     | 264        |
| <b>Capitolo 52: Tipografia.....</b>   | <b>266</b> |
| Sintassi.....                         | 266        |
| Parametri.....                        | 266        |
| Osservazioni.....                     | 267        |
| Examples.....                         | 267        |
| Dimensione del font.....              | 267        |
| La stenografia del font.....          | 267        |
| Stack di font.....                    | 268        |
| Spaziatura del carattere.....         | 269        |
| Trasformazione del testo.....         | 269        |
| Indentatura del testo.....            | 269        |
| Decorazione del testo.....            | 270        |
| Overflow del testo.....               | 270        |
| Word Spacing.....                     | 271        |
| Direzione del testo.....              | 272        |
| Carattere Variante.....               | 272        |
| Citazioni.....                        | 273        |
| Ombra di testo.....                   | 273        |
| Ombra senza raggio di sfocatura.....  | 273        |
| Ombra con il raggio di sfocatura..... | 273        |
| Ombre multiple.....                   | 273        |
| <b>Capitolo 53: transizioni.....</b>  | <b>275</b> |

|  |            |
|--|------------|
| Sintassi.....  | 275        |
| Parametri.....   | 275        |
| Osservazioni.....  | 275        |
| Examples.....  | 275        |
| Stenografia di transizione.....  | 276        |
| Transizione (longhand).....  | 276        |
| <b>CSS.....</b>  | <b>276</b> |
| <b>HTML.....</b>   | <b>276</b> |
| cubico-Bezier.....   | 277        |
| <b>Capitolo 54: Trasformazioni 2D.....</b>                               | <b>279</b> |
| Sintassi.....  | 279        |
| Parametri.....   | 279        |
| Osservazioni.....  | 280        |
| Sistema 2D Coordinante.....  | 280        |
| Supporto per browser e prefissi.....                                     | 281        |
| Esempio di trasformazione prefissata:.....                               | 281        |
| Examples.....  | 281        |
| Ruotare.....   | 281        |
| Scala.....   | 282        |
| Tradurre.....  | 282        |
| Storto.....  | 283        |
| Trasformazioni multiple.....   | 283        |
| Trasforma l'origine.....   | 285        |
| <b>Capitolo 55: Trasformazioni 3D.....</b>                               | <b>287</b> |
| Osservazioni.....  | 287        |
| Sistema di coordinate.....   | 287        |
| Examples.....  | 287        |
| Cubo 3D.....   | 288        |
| controfaccia visibilità.....   | 289        |
| Puntatore della bussola o forma dell'ago mediante trasformazioni 3D..... | 290        |
| <b>CSS.....</b>  | <b>290</b> |

|   |            |
|---|------------|
| <b>HTML</b> .....                                       | <b>290</b> |
| Effetto testo 3D con ombra.....                         | 291        |
| <b>Capitolo 56: Unità di lunghezza</b> .....            | <b>294</b> |
| introduzione.....                                       | 294        |
| Sintassi.....   | 294        |
| Parametri.....  | 294        |
| Osservazioni.....                                       | 295        |
| Examples.....   | 295        |
| Dimensione del carattere con rem.....                   | 295        |
| Creazione di elementi scalabili usando rems ed ems..... | 296        |
| vh e vw.....  | 297        |
| vmin e vmax.....  | 297        |
| usando percentuale%.....                                | 297        |
| <b>Titoli di coda</b> .....                             | <b>299</b> |

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [css](#)

It is an unofficial and free CSS ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official CSS.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Capitolo 1: Iniziare con i CSS

## Osservazioni

Gli stili possono essere creati in diversi modi, consentendo diversi gradi di riutilizzo e ambito quando sono specificati in un documento HTML sorgente. I fogli di stile *esterni* possono essere riutilizzati tra i documenti HTML. I fogli di stile *incorporati* si applicano all'intero documento in cui sono specificati. Gli stili in *linea* si applicano solo al singolo elemento HTML su cui sono specificati.

## Versioni

| Versione | Data di rilascio |
|----------|------------------|
| 1        | 1996/12/17       |
| 2        | 1998/05/12       |
| 3        | 2015/10/13       |

## Examples

### Foglio di stile esterno

Un foglio di stile CSS esterno può essere applicato a qualsiasi numero di documenti HTML posizionando un elemento `<link>` in ogni documento HTML.

L'attributo `rel` del `rel <link>` deve essere impostato su `"stylesheet"` e l'attributo `href` al percorso relativo o assoluto del foglio di stile. Sebbene l'utilizzo di percorsi URL relativi sia generalmente considerato una buona pratica, è possibile utilizzare anche percorsi assoluti. In HTML5 l'attributo `type` [può essere omissso](#).

Si consiglia di posizionare il tag `<link>` tag `<head>` del file HTML in modo che gli stili vengano caricati prima degli elementi da loro stilizzati. In caso contrario, gli [utenti vedranno un lampo di contenuti senza stile](#).

---

## Esempio

### ciao-world.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
```

```
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>Hello world!</h1>
  <p>I ♥ CSS</p>
</body>
</html>
```

## style.css

```
h1 {
  color: green;
  text-decoration: underline;
}
p {
  font-size: 25px;
  font-family: 'Trebuchet MS', sans-serif;
}
```

Assicurati di includere il percorso corretto per il tuo file CSS nell'href. Se il file CSS si trova nella stessa cartella del file HTML, non è richiesto alcun percorso (come nell'esempio precedente), ma se è salvato in una cartella, specificarlo come questo `href="foldername/style.css"`.

```
<link rel="stylesheet" type="text/css" href="foldername/style.css">
```

I fogli di stile esterni sono considerati il modo migliore per gestire il tuo CSS. C'è una ragione molto semplice per questo: quando gestisci un sito di, ad esempio, 100 pagine, tutte controllate da un singolo foglio di stile e desideri cambiare i colori dei collegamenti da blu a verde, è molto più semplice apportare la modifica nel tuo file CSS e lascia che le modifiche "si sovrappongano" su tutte le 100 pagine piuttosto che andare in 100 pagine separate e fare la stessa modifica 100 volte. Anche in questo caso, se si desidera modificare completamente l'aspetto del proprio sito Web, è necessario aggiornare solo questo file.

È possibile caricare tutti i file CSS nella pagina HTML in base alle esigenze.

```
<link rel="stylesheet" type="text/css" href="main.css">
<link rel="stylesheet" type="text/css" href="override.css">
```

Le regole CSS sono applicate con alcune regole di base e l'ordine non importa. Ad esempio, se hai un file `main.css` con del codice in esso:

```
p.green { color: #00FF00; }
```

Tutti i tuoi paragrafi con la classe 'verde' saranno scritti in verde chiaro, ma puoi sovrascriverli con un altro file `.css` semplicemente includendolo *dopo* `main.css`. Puoi avere `override.css` con il seguente codice seguire `main.css`, ad esempio:

```
p.green { color: #006600; }
```

Ora tutti i tuoi paragrafi con la classe "verde" saranno scritti in verde scuro anziché verde chiaro.

Si applicano altri principi, come la regola "! Importante", la specificità e l'ereditarietà.

Quando qualcuno visita per la prima volta il tuo sito Web, il suo browser scarica l'HTML della pagina corrente più il file CSS collegato. Quindi, quando navigano verso un'altra pagina, il loro browser deve solo scaricare l'HTML di quella pagina; il file CSS è memorizzato nella cache, quindi non è necessario scaricarlo nuovamente. Poiché i browser memorizzano nella cache il foglio di stile esterno, le tue pagine vengono caricate più velocemente.

## Stili interni

I CSS racchiusi tra i `<style></style>` all'interno di un documento HTML funzionano come un foglio di stile esterno, tranne per il fatto che risiede nel documento HTML che disegna invece che in un file separato e pertanto può essere applicato solo al documento in cui vive. Si noti che questo elemento *deve* trovarsi all'interno dell'elemento `<head>` per la convalida HTML (sebbene funzioni in tutti i browser correnti se inserito nel `body` ).

```
<head>
  <style>
    h1 {
      color: green;
      text-decoration: underline;
    }
    p {
      font-size: 25px;
      font-family: 'Trebuchet MS', sans-serif;
    }
  </style>
</head>
<body>
  <h1>Hello world!</h1>
  <p>I ♥ CSS</p>
</body>
```

## Stili in linea

Usa gli stili in linea per applicare lo stile a un elemento specifico. Si noti che questo **non** è ottimale. L'inserimento di regole di stile in un tag `<style>` o in un file CSS esterno è incoraggiato al fine di mantenere una distinzione tra contenuto e presentazione.

Gli stili in linea sostituiscono qualsiasi CSS in un tag `<style>` o in un foglio di stile esterno. Sebbene ciò possa essere utile in alcune circostanze, questo fatto riduce il più delle volte la manutenibilità di un progetto.

Gli stili nel seguente esempio si applicano direttamente agli elementi a cui sono collegati.

```
<h1 style="color: green; text-decoration: underline;">Hello world!</h1>
<p style="font-size: 25px; font-family: 'Trebuchet MS';">I ♥ CSS</p>
```

Gli stili in linea sono generalmente il modo più sicuro per garantire la compatibilità di rendering tra vari client, programmi e dispositivi di posta elettronica, ma può richiedere molto tempo per essere scritto e un po' difficile da gestire.

## Regola @import CSS (una delle regole at-CSS)

La regola @import @import viene utilizzata per importare regole di stile da altri fogli di stile. Queste regole devono precedere tutti gli altri tipi di regole, eccetto le regole di @charset; poiché non è un'istruzione nidificata, @import non può essere utilizzato all'interno di regole condizionali del gruppo. @import .

## Come usare @import

Puoi utilizzare la regola @import nei seguenti modi:

### A. Con tag di stile interno

```
<style>
  @import url('/css/styles.css');
</style>
```

### B. Con foglio di stile esterno

La seguente riga importa un file CSS denominato `additional-styles.css` nella directory root nel file CSS in cui appare:

```
@import '/additional-styles.css';
```

È anche possibile importare CSS esterno. Un caso d'uso comune sono i file dei font.

```
@import 'https://fonts.googleapis.com/css?family=Lato';
```

Un secondo argomento facoltativo per la regola @import è un elenco di query multimediali:

```
@import '/print-styles.css' print;
@import url('landscape.css') screen and (orientation:landscape);
```

## Cambiare CSS con JavaScript

### JavaScript puro

È possibile aggiungere, rimuovere o modificare i valori delle proprietà CSS con JavaScript attraverso la proprietà di `style` un elemento.

```
var el = document.getElementById("element");
el.style.opacity = 0.5;
el.style.fontFamily = 'sans-serif';
```

Si noti che le proprietà di stile sono denominate in uno stile di cassa del cammello inferiore. Nell'esempio si vede che la `font-family` proprietà css diventa `fontFamily` in javascript.

In alternativa al lavoro diretto sugli elementi, puoi creare un elemento `<style>` o `<link>` in JavaScript e aggiungerlo al `<body>` o `<head>` del documento HTML.

## jQuery

La modifica delle proprietà CSS con jQuery è ancora più semplice.

```
$('#element').css('margin', '5px');
```

Se è necessario modificare più di una regola di stile:

```
$('#element').css({
  margin: "5px",
  padding: "10px",
  color: "black"
});
```

jQuery include due modi per cambiare le regole CSS che contengono trattini (ad es. `font-size`). Puoi metterli tra virgolette o cammello-caso il nome della regola di stile.

```
$('.example-class').css({
  "background-color": "blue",
  fontSize: "10px"
});
```

---

## Guarda anche

- [Documentazione JavaScript - Leggere e modificare lo stile CSS](#) .
- [Documentazione jQuery - Manipolazione CSS](#)

## Elenchi di stili con CSS

Esistono tre diverse proprietà per lo stile di voci di `list-style-type`: `list-style-type`, `list-style-image` e `list-style-position`, che dovrebbero essere dichiarate in tale ordine. I valori predefiniti sono `disc`, `outside` e `none`, rispettivamente. Ogni proprietà può essere dichiarata separatamente o utilizzando la proprietà abbreviata in `list-style`.

`list-style-type` definisce la forma o il tipo di punto elenco utilizzato per ogni elemento dell'elenco.

Alcuni dei valori accettabili per il tipo di `list-style-type`:

- disco
- cerchio
- piazza
- decimale
- inferiore-romana
- superiore romano

- nessuna

(Per un elenco completo, consultare il [wiki delle specifiche](#) del W3C )

Ad esempio, per utilizzare punti elenco quadrati per ciascun elemento di elenco, si utilizzerà la seguente coppia valore-valore:

```
li {  
  list-style-type: square;  
}
```

---

La proprietà `list-style-image` determina se l'icona elenco-elemento è impostata con un'immagine e accetta un valore pari a `none` o un URL che punta a un'immagine.

```
li {  
  list-style-image: url(images/bullet.png);  
}
```

---

La proprietà `list-style-position` definisce dove posizionare il marker item-list e accetta uno dei due valori: "inside" o "outside".

```
li {  
  list-style-position: inside;  
}
```

Leggi Iniziare con i CSS online: <https://riptutorial.com/it/css/topic/293/iniziare-con-i-css>

# Capitolo 2: Adattamento e posizionamento dell'oggetto

## Osservazioni

Le proprietà `object-fit` e `object-position` non sono supportate da Internet Explorer.

## Examples

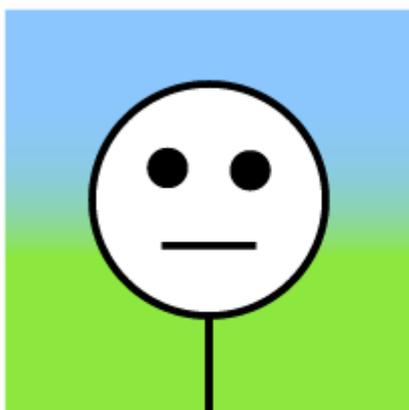
### oggetto-fit

La proprietà **object-fit** definirà il modo in cui un elemento si adatterà a una scatola con un'altezza e una larghezza stabilite. Generalmente applicato a un'immagine o a un video, Object-fit accetta i seguenti cinque valori:

#### RIEMPIRE

```
object-fit:fill;
```

original image



object-fit: fill;

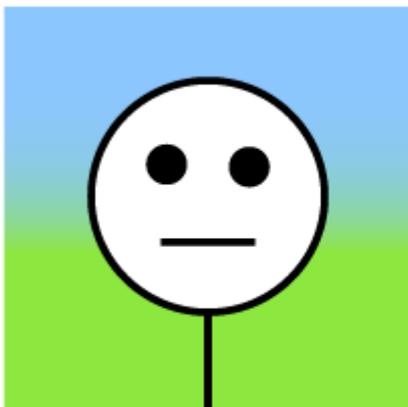


Riempimento allunga l'immagine per adattarla alla casella del contenuto, indipendentemente dalle proporzioni originali dell'immagine.

#### CONTENERE

```
object-fit:contain;
```

original image



object-fit: contain;

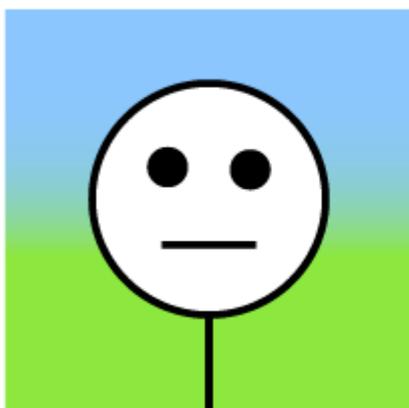


Contiene adatta l'immagine nell'altezza o larghezza della scatola mantenendo le proporzioni dell'immagine.

## COPERTINA

```
object-fit: cover;
```

original image



object-fit: cover;

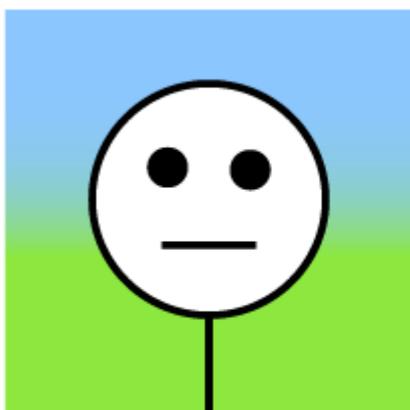


La copertina riempie l'intera scatola con l'immagine. Le proporzioni dell'immagine vengono mantenute, ma l'immagine viene ritagliata alle dimensioni della casella.

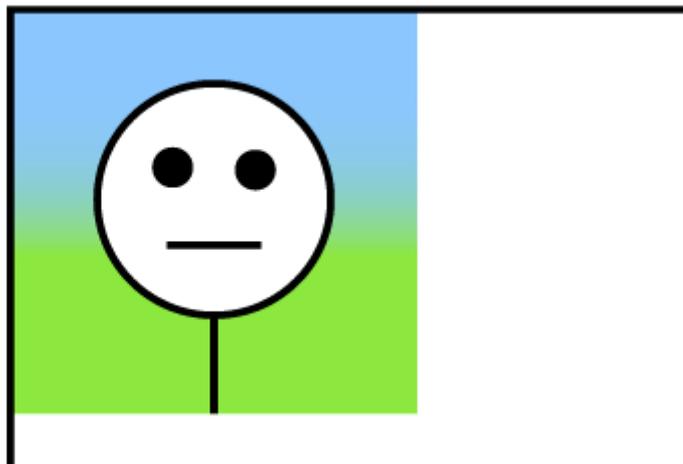
## NESSUNA

```
object-fit: none;
```

original image



object-fit: none;



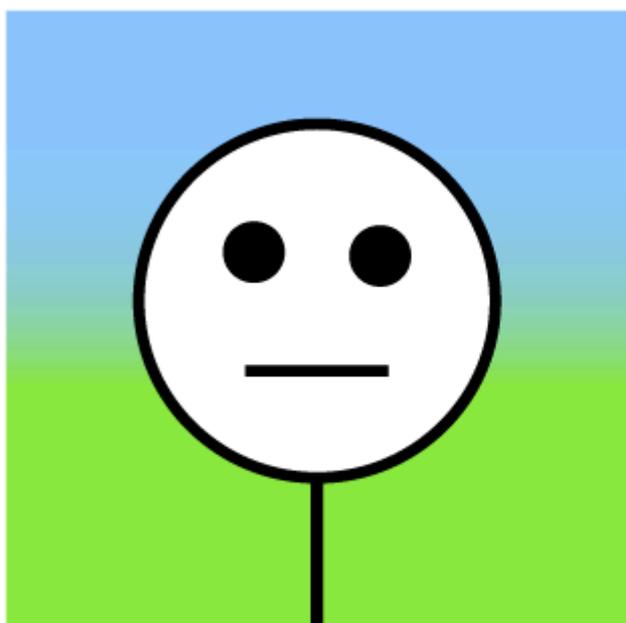
Nessuno ignora le dimensioni della casella e non viene ridimensionato.

## RIMPICCIOLIRE

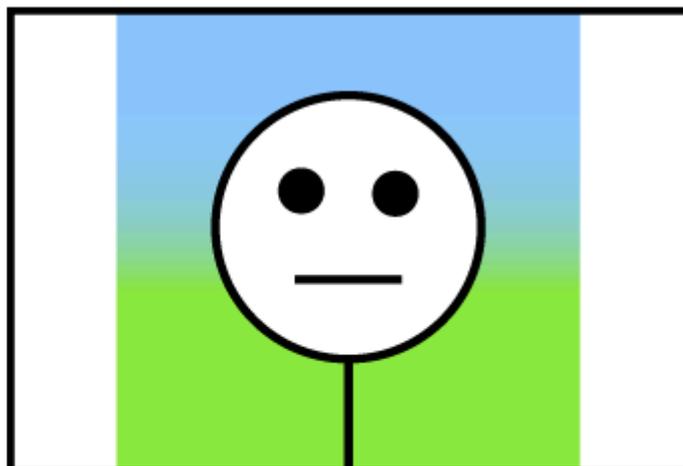
```
object-fit: scale-down;
```

Ridimensiona o ridimensiona l'oggetto come `none` o come `contain`. Mostra qualsiasi opzione si traduce in una dimensione dell'immagine più piccola.

original image



object-fit: scale-down;



Leggi [Adattamento e posizionamento dell'oggetto online](https://riptutorial.com/it/css/topic/5520/adattamento-e-posizionamento-dell-oggetto):

<https://riptutorial.com/it/css/topic/5520/adattamento-e-posizionamento-dell-oggetto>

# Capitolo 3: animazioni

## Sintassi

- `transition: <property> <duration> <timing-function> <delay>;`
- `@keyframes <identifier>`
- `[ [ from | to | <percentage> ] [, from | to | <percentage> ]* block ]*`

## Parametri

| Transizione                 |   |
|-----------------------------|---|
| Parametro                   | Dettagli  |
| proprietà                   | O la proprietà CSS per la transizione, o <code>all</code> , che specifica tutte le proprietà di transizione.  |
| durata                      | Tempo di transizione, in secondi o in millisecondi.   |
| temporizzazione<br>funzione | Specifica una funzione per definire come vengono calcolati i valori intermedi per le proprietà. I valori comuni sono <code>ease</code> , <code>linear</code> e <code>step-end</code> . Controlla il <a href="#">cheat-sheet della funzione easing</a> per saperne di più. |
| ritardo                     | Quantità di tempo, in secondi o millisecondi, da attendere prima di riprodurre l'animazione.  |
|                             |   |
| <b>@keyframes</b>           |   |
| [da   a  <br><percentage> ] | È possibile specificare un tempo impostato con un valore percentuale o due valori percentuali, ad esempio <code>10%</code> , <code>20%</code> , per un periodo di tempo in cui sono impostati gli attributi dell'insieme di fotogrammi chiave.                            |
| block                       | Qualsiasi quantità di attributi CSS per il fotogramma chiave.   |

## Examples

### Animazioni con la proprietà di transizione

Utile per animazioni semplici, la proprietà di `transition` CSS consente alle proprietà CSS basate su numero di animare tra stati.

## Esempio

```
.Example{
  height: 100px;
  background: #fff;
}

.Example:hover{
  height: 120px;
  background: #ff0000;
}
```

### Visualizza risultato

Per impostazione predefinita, passando con il mouse su un elemento con la classe `.Example`, l'altezza dell'elemento salta `100px` a `120px` e il colore di sfondo `100px` rosso (`#ff0000`).

Aggiungendo la proprietà di `transition`, è possibile che queste modifiche si verifichino nel tempo:

```
.Example{
  ...
  transition: all 400ms ease;
}
```

### Visualizza risultato

Il valore `all` applica la transizione a tutte le proprietà compatibili (basate sui numeri). Qualsiasi nome struttura compatibile (ad esempio `height` o `top`) può essere sostituito per questa parola chiave.

`400ms` specifica il tempo richiesto dalla transizione. In questo caso, il cambio di altezza dell'elemento richiederà 400 millisecondi.

Infine, la `ease` valore è la funzione di animazione, che determina come viene riprodotta l'animazione. `ease` significa iniziare lentamente, accelerare, quindi terminare lentamente. Altri valori sono `linear`, `ease-out` e `ease-in`.

---

## Compatibilità tra browser

La proprietà di `transition` è generalmente ben supportata su tutti i principali browser, ad eccezione di IE 9. Per le versioni precedenti dei browser basati su Firefox e Webkit, utilizzare i prefissi dei fornitori in questo modo:

```
.Example{
  transition:          all 400ms ease;
  -moz-transition:    all 400ms ease;
  -webkit-transition: all 400ms ease;
}
```

---

**Nota:** la proprietà di `transition` può animare le modifiche tra due valori numerici, indipendentemente dall'unità. Può anche transitare tra unità, come da `100px` a `50vh`. Tuttavia, non

può transitare tra un numero e un valore predefinito o automatico, come la transizione dell'altezza di un elemento da `100px` a `auto`.

## Aumentare le prestazioni dell'animazione usando l'attributo `will-change`

Quando crei animazioni e altre azioni pesanti in GPU, è importante comprendere l'attributo `will-change`.

Entrambi i fotogrammi chiave CSS e la proprietà di `transition` utilizzano l'accelerazione GPU. Le prestazioni aumentano scaricando i calcoli sulla GPU del dispositivo. Questo viene fatto creando dei layer di vernice (parti della pagina che vengono singolarmente renderizzati) che vengono scaricati nella GPU per essere calcolati. La proprietà `will-change` dice al browser cosa si animerà, consentendo al browser di creare aree di disegno più piccole, aumentando così le prestazioni.

La proprietà `will-change` accetta un elenco di proprietà separate da virgola da animare. Ad esempio, se hai intenzione di trasformare un oggetto e modificarne l'opacità, devi specificare:

```
.Example{
  ...
  will-change: transform, opacity;
}
```

**Nota:** utilizzare `will-change` con parsimonia. L'impostazione `will-change` per ogni elemento di una pagina può causare problemi di prestazioni, poiché il browser potrebbe tentare di creare livelli di pittura per ogni elemento, aumentando notevolmente la quantità di elaborazione eseguita dalla GPU.

## Animazioni con fotogrammi chiave

Per le animazioni CSS multistadio, è possibile creare CSS `@keyframes`. I fotogrammi chiave consentono di definire più punti di animazione, chiamati fotogrammi chiave, per definire animazioni più complesse.

---

## Esempio di base

In questo esempio, creeremo un'animazione di base di base che cicla tra tutti i colori.

```
@keyframes rainbow-background {
  0%      { background-color: #ff0000; }
  8.333%  { background-color: #ff8000; }
  16.667% { background-color: #ffff00; }
  25.000% { background-color: #80ff00; }
  33.333% { background-color: #00ff00; }
  41.667% { background-color: #00ff80; }
  50.000% { background-color: #00ffff; }
  58.333% { background-color: #0080ff; }
  66.667% { background-color: #0000ff; }
  75.000% { background-color: #8000ff; }
  83.333% { background-color: #ff00ff; }
```

```
    91.667%    { background-color: #ff0080; }
    100.00%   { background-color: #ff0000; }
}

.RainbowBackground {
  animation: rainbow-background 5s infinite;
}
```

## Visualizza risultato

Ci sono alcune cose diverse da notare qui. Innanzitutto, la sintassi `@keyframes` effettiva.

```
@keyframes rainbow-background{
```

Questo imposta il nome dell'animazione `rainbow-background`.

```
0%    { background-color: #ff0000; }
```

Questa è la definizione di un fotogramma chiave all'interno dell'animazione. La prima parte, lo `0%` nel caso, definisce dove si trova il fotogramma chiave durante l'animazione. Lo `0%` indica che è `0%` del tempo di animazione totale dall'inizio.

L'animazione passerà automaticamente tra i fotogrammi chiave. Quindi, impostando il colore di sfondo successivo al `8.333%`, l'animazione impiegherà senza intoppi l'`8,333%` delle volte per passare da un fotogramma chiave all'altro.

```
.RainbowBackground {
  animation: rainbow-background 5s infinite;
}
```

Questo codice associa la nostra animazione a tutti gli elementi che hanno la classe

```
.RainbowBackground.
```

La proprietà di animazione effettiva accetta i seguenti argomenti.

- **nome-animazione** : il nome della nostra animazione. In questo caso, `rainbow-background`
- **durata animazione** : la **durata** dell'animazione, in questo caso 5 secondi.
- **animation-iteration-count (Opzionale)** : il numero di volte in cui l'animazione verrà ripetuta. In questo caso, l'animazione continuerà all'infinito. Per impostazione predefinita, l'animazione verrà riprodotta una volta.
- **ritardo di animazione (facoltativo)** : specifica il tempo di attesa prima dell'avvio dell'animazione. Il valore predefinito è 0 secondi e può assumere valori negativi. Ad esempio, `-2s` l'animazione per 2 secondi nel suo ciclo.
- **animation-timing-function (Opzionale)** : specifica la curva di velocità dell'animazione. Il valore predefinito per `ease`, in cui l'animazione inizia lenta, diventa più veloce e finisce lento.

In questo particolare esempio, entrambi i fotogrammi chiave `0%` e `100%` specificano `{ background-color: #ff0000; }`. Ovunque due o più fotogrammi chiave condividano uno stato, è possibile specificarli in una singola istruzione. In questo caso, le due linee `0%` e `100%` potrebbero essere

sostituire con questa singola riga:

```
0%, 100% { background-color: #ff0000; }
```

## Compatibilità cross-browser

Per i browser più vecchi basati su WebKit, è necessario utilizzare il prefisso del venditore sia sulla dichiarazione `@keyframes` sia sulla proprietà di `animation`, in questo modo:

```
@-webkit-keyframes{  
  
-webkit-animation: ...
```

## Esempi di sintassi

Il nostro primo esempio di sintassi mostra l'animazione della proprietà abbreviata utilizzando tutte le proprietà / parametri disponibili:

```
animation: 3s ease-in 1s 2 reverse both  
paused slidein;  
/* duration | timing-function | delay | iteration-count | direction | fill-mode |  
play-state | name */
```

Il nostro secondo esempio è un po' più semplice e mostra che alcune proprietà possono essere omesse:

```
animation: 3s linear 1s slidein;  
/* duration | timing-function | delay | name */
```

Il nostro terzo esempio mostra la dichiarazione più minimale. Nota che il nome dell'animazione e la durata dell'animazione devono essere dichiarati:

```
animation: 3s slidein;  
/* duration | name */
```

Vale anche la pena ricordare che quando si utilizza l'animazione stenografia l'ordine delle proprietà fa la differenza. Ovviamente il browser potrebbe confondere la durata con il ritardo.

Se la brevità non è la tua passione, puoi anche saltare la proprietà stenografia e scrivere ciascuna proprietà individualmente:

```
animation-duration: 3s;  
animation-timing-function: ease-in;  
animation-delay: 1s;  
animation-iteration-count: 2;  
animation-direction: reverse;  
animation-fill-mode: both;
```

```
animation-play-state: paused;  
animation-name: slidein;
```

Leggi animazioni online: <https://riptutorial.com/it/css/topic/590/animazioni>

---

# Capitolo 4: Blocca i contesti di formattazione

## Osservazioni

[Un contesto di formattazione dei blocchi fa parte di un rendering CSS visuale di una pagina Web. È la regione in cui si verifica il layout delle caselle di blocco e in cui i float interagiscono tra loro.] [1]

[1]: [https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Block\\_formatting\\_context](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Block_formatting_context) MDN

## Examples

### Utilizzando la proprietà di overflow con un valore diverso da visibile

```
img{
  float:left;
  width:100px;
  margin:0 10px;
}
.div1{
  background:#f1f1f1;
  /* does not create block formatting context */
}
.div2{
  background:#f1f1f1;
  overflow:hidden;
  /* creates block formatting context */
}
```

```

1 
2 <div class=div1>
3 <p>Lorem ipsum dolor sit amet, cum no paulo mollis pertinacia. Eam in velit graecis, sea mucus insolens ne. Amet doming at has, omnis errem an cum. Eu vim appareat persecuti, ea putant definitionem has, vis ea legendos expetenda. No eros graeci minimum nam, justo augue instructor usu ne. At ludus suscipit disputationi vel.</p>
4
5 <p>Ad case omnis nam, mutat deseruisse persequeris eos ad, in tollit debitis sea. Cu eos munere virtute vituperata. Exerci bonorum sed id, id nec tantas praesent complectitur. Vel cu legendos mediocritatem. Enim liberavisse ei sea.</p>
6 </div>
7
8 
9 <div class=div2>
10 <p>Lorem ipsum dolor sit amet, cum no paulo mollis pertinacia. Eam in velit graecis, sea mucus insolens ne. Amet doming at has, omnis errem an cum. Eu vim appareat persecuti, ea putant definitionem has, vis ea legendos expetenda. No eros graeci minimum nam, justo augue instructor usu ne. At ludus suscipit disputationi vel.</p>
11
12 <p>Ad case omnis nam, mutat deseruisse persequeris eos ad, in tollit debitis sea. Cu eos munere virtute vituperata. Exerci bonorum sed id, id nec tantas praesent complectitur. Vel cu legendos mediocritatem. Enim liberavisse ei sea.</p>
13 </div>

```

```

1 img{
2   float:left;
3   width:100px;
4   margin:0 10px;
5 }
6 .div1{
7   background:#f1f1f1;
8 }
9 .div2{
10  background:#f1f1f1;
11  overflow:hidden;
12  /* creates block formatting c
13 }

```



Ad case omnis nam, mutat deseruisse persequeris eos ad, in tollit debitis sea. Cu eos munere virtute vituperata. Exerci bonorum sed id, id nec tantas praesent complectitur.



Ad case omnis nam, mutat deseruisse persequeris eos ad, in tollit debitis sea. Cu eos munere virtute vituperata. Exerci bonorum sed id, id nec tantas praesent complectitur.

<https://jsfiddle.net/MadalinaTn/qkwwmu6m/2/>

L'utilizzo della proprietà di overflow con un valore diverso da visible (predefinito) creerà un nuovo contesto di formattazione dei blocchi. Ciò è tecnicamente necessario: se un oggetto mobile si interseca con l'elemento di scorrimento, riavvolge forzatamente il contenuto.

Questo esempio che mostra come un certo numero di paragrafi interagirà con un'immagine fluttuante è simile a [questo esempio](#) , su css-tricks.com.

2 : <https://developer.mozilla.org/en-US/docs/Web/CSS/overflow> MDN

Leggi [Blocca i contesti di formattazione online](#): <https://riptutorial.com/it/css/topic/5069/blocca-i-contesti-di-formattazione>

# Capitolo 5: box-ombra

## Sintassi

- box-shadow: none | h-shadow v-shadow blur diffonde colore | inset | initial | inherit;

## Parametri

| parametri        | Dettagli   |
|------------------|--|
| inserire         | per impostazione predefinita, l'ombra viene considerata come ombra esterna. la parola chiave inserita disegna l'ombra all'interno della cornice / del bordo. |
| Offset-x         | la distanza orizzontale  |
| offset y         | la distanza verticale  |
| sfocatura raggio | 0 per impostazione predefinita. il valore non può essere negativo. più grande è il valore, più grande e leggero diventa l'ombra.                             |
| spread-radius    | 0 per impostazione predefinita. i valori positivi causeranno l'espansione dell'ombra. valori negativi provocheranno il restringimento dell'ombra.            |
| colore           | può essere di varie notazioni: una parola chiave color, esadecimale, <code>rgb()</code> , <code>rgba()</code> , <code>hsl()</code> , <code>hsla()</code>     |

## Osservazioni

Supporto per browser:

- Chrome 10.0
- IE 9.0
- Firefox 4.0 3.5 -moz
- Safari 5.1 3.1 -webkit-
- Opera 10.5

## Examples

### drop shadow

JSFiddle: <https://jsfiddle.net/UnsungHero97/80qod7aL/>

### HTML

```
<div class="box_shadow"></div>
```

## CSS

```
.box_shadow {  
  -webkit-box-shadow: 0px 0px 10px -1px #444444;  
  -moz-box-shadow: 0px 0px 10px -1px #444444;  
  box-shadow: 0px 0px 10px -1px #444444;  
}
```

## ombra esterna

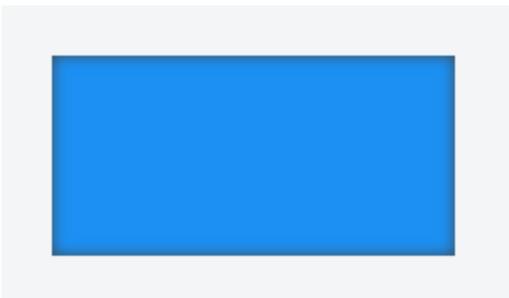
### HTML

```
<div class="box_shadow"></div>
```

## CSS

```
.box_shadow {  
  background-color: #1C90F3;  
  width: 200px;  
  height: 100px;  
  margin: 50px;  
  -webkit-box-shadow: inset 0px 0px 10px 0px #444444;  
  -moz-box-shadow: inset 0px 0px 10px 0px #444444;  
  box-shadow: inset 0px 0px 10px 0px #444444;  
}
```

### Risultato:



JSFiddle: <https://jsfiddle.net/UnsungHero97/80qod7aL/1/>

## ombreggiatura di solo livello inferiore utilizzando uno pseudo-elemento

JSFiddle: <https://jsfiddle.net/UnsungHero97/80qod7aL/2/>

### HTML

```
<div class="box_shadow"></div>
```

## CSS

```
.box_shadow {
  background-color: #1C90F3;
  width: 200px;
  height: 100px;
  margin: 50px;
}

.box_shadow:after {
  content: "";
  width: 190px;
  height: 1px;
  margin-top: 98px;
  margin-left: 5px;
  display: block;
  position: absolute;
  z-index: -1;
  -webkit-box-shadow: 0px 0px 8px 2px #444444;
  -moz-box-shadow: 0px 0px 8px 2px #444444;
  box-shadow: 0px 0px 8px 2px #444444;
}
```



## più ombre

JSFiddle: <https://jsfiddle.net/UnsungHero97/80qpd7aL/5/>

## HTML

```
<div class="box_shadow"></div>
```

## CSS

```
.box_shadow {
  width: 100px;
  height: 100px;
```

```
margin: 100px;
box-shadow:
  -52px -52px 0px 0px #f65314,
  52px -52px 0px 0px #7cbb00,
  -52px 52px 0px 0px #00a1f1,
  52px 52px 0px 0px #ffbb00;
}
```



Leggi box-ombra online: <https://riptutorial.com/it/css/topic/1746/box-ombra>

---

# Capitolo 6: Cascading e Specificità

## Osservazioni

La specificità dei CSS intende promuovere la concisione del codice consentendo all'autore di definire alcune regole generali di formattazione per un ampio insieme di elementi e quindi di sovrascriverli per un determinato sottoinsieme.

## Examples

### Cascading

Cascading e specificità sono usati insieme per determinare il valore finale di una proprietà di stile CSS. Definiscono inoltre i meccanismi per risolvere i conflitti nelle serie di regole CSS.

---

## Ordine di caricamento CSS

Gli stili sono letti dalle seguenti fonti, in questo ordine:

1. Foglio di stile User Agent (gli stili forniti dal fornitore del browser)
2. Foglio di stile dell'utente (Lo stile aggiuntivo che un utente ha impostato sul suo browser)
3. Stili di autore (qui autore indica il creatore della pagina web / sito Web)
  - Forse uno o più file `.css`
  - Nell'elemento `<style>` del documento HTML
4. Stili in linea (nell'attributo `style` su un elemento HTML)

Il browser cercherà gli stili corrispondenti durante il rendering di un elemento.

---

## Come vengono risolti i conflitti?

Quando solo una serie di regole CSS sta tentando di impostare uno stile per un elemento, non vi è alcun conflitto e tale serie di regole viene utilizzata.

Quando vengono trovati più set di regole con impostazioni in conflitto, prima le regole di specificità e quindi le regole a cascata vengono utilizzate per determinare quale stile utilizzare.

## Esempio 1 - Regole di specificità

```
.mystyle { color: blue; } /* specificity: 0, 0, 1, 0 */
div { color: red; }      /* specificity: 0, 0, 0, 1 */
```

```
<div class="mystyle">Hello World</div>
```

Di che colore sarà il testo? (passa il mouse per vedere la risposta)

blu

Prima vengono applicate le regole di specificità e quella con la specificità più alta "vince".

## Esempio 2 - Regole in cascata con selettori identici

*File css esterno*

```
.class {  
  background: #FFF;  
}
```

*Css interno (nel file HTML)*

```
<style>  
.class {  
  background: #000;  
}  
</style>
```

In questo caso, in cui si hanno selettori identici, la cascata si attiva e determina che l'ultimo caricato "vince".

## Esempio 3: regole a cascata dopo le regole di specificità

```
body > .mystyle { background-color: blue; } /* specificity: 0, 0, 1, 1 */  
.otherstyle > div { background-color: red; } /* specificity: 0, 0, 1, 1 */
```

```
<body class="otherstyle">  
  <div class="mystyle">Hello World</div>  
</body>
```

Di che colore sarà lo sfondo?

rosso

Dopo aver applicato le regole di specificità, c'è ancora un conflitto tra blu e rosso, quindi le regole a cascata vengono applicate in aggiunta alle regole di specificità. Il collegamento in cascata esamina l'ordine di caricamento delle regole, sia all'interno dello stesso file `.css` che nella raccolta di fonti di stile. L'ultimo caricato sovrascrive quelli precedenti. In questo caso, la regola del `.otherstyle > div` "vince".

### Un'ultima nota

- La specificità del selettore ha sempre la precedenza.
- Legami di rottura dell'ordine di fogli di stile.
- Gli stili in linea vincono tutto.

## La! Dichiarazione importante

La dichiarazione `!important` viene utilizzata per ignorare la solita specificità in un foglio di stile dando una priorità più alta a una regola. Il suo utilizzo è: `property : value !important;`

```
#mydiv {
  font-weight: bold !important;      /* This property won't be overridden
                                     by the rule below */
}

#outerdiv #mydiv {
  font-weight: normal;               /* #mydiv font-weight won't be set to normal
                                     even if it has a higher specificity because
                                     of the !important declaration above */
}
```

Evitare l'uso di `!important` è fortemente raccomandato (a meno che non sia assolutamente necessario), perché disturberà il flusso naturale delle regole CSS che può portare incertezza nel tuo foglio di stile. Inoltre è importante notare che quando più dichiarazioni `!important` sono applicate alla stessa regola su un determinato elemento, quella con la specificità più alta sarà l'ona applicata.

Ecco alcuni esempi in cui l'uso di una dichiarazione `!important` può essere giustificato:

- Se le tue regole non devono essere sovrascritte da alcuno stile in linea dell'elemento che è scritto all'interno dell'attributo `style` dell'elemento html.
- Per dare all'utente un maggiore controllo sull'accessibilità del web, come aumentare o diminuire la dimensione della dimensione del font, ignorando lo stile dell'autore usando `!important`.
- Per testare e fare il debug usando l'elemento inspect.

Guarda anche:

- [W3C - 6 Assegnazione di valori di proprietà, Cascading ed Ereditarietà - 6.4.2! Regole importanti](#)

## Calcolo della specificità del selettore

Ogni singolo selettore CSS ha il proprio valore di specificità. Ogni selettore in una sequenza aumenta la specificità complessiva della sequenza. I selectori rientrano in uno dei tre diversi gruppi di specificità: *A*, *B* e *C*. Quando più sequenze di selectori selezionano un determinato elemento, il browser utilizza gli stili applicati dalla sequenza con la massima specificità complessiva.

| Gruppo    | Composto da  | Esempi   |
|-----------|--|--|
| <i>UN</i> | selectori di identificazione                                   | <code>#foo</code>  |
| <i>B</i>  | selectori di classe<br>selectori di attributi<br>pseudo-classi | <code>.bar</code><br><code>[title]</code> , <code>[colspan="2"]</code><br><code>:hover</code> <code>:nth-child(2)</code> |

| Gruppo   | Composto da                             | Esempi  |
|----------|---|---|
| <b>C</b> | selezionare i selettori pseudo-elementi | <code>div, li</code><br><code>::before, ::first-letter</code> |

Il gruppo *A* è il più specifico, seguito dal gruppo *B*, infine il gruppo *C*.

Il selettore universale (`*`) e i combinatori (come `>` e `~`) non hanno specificità.

### Esempio 1: specificità delle varie sequenze di selettori

```
#foo #baz {}      /* a=2, b=0, c=0 */
#foo.bar {}      /* a=1, b=1, c=0 */
#foo {}          /* a=1, b=0, c=0 */
.bar:hover {}    /* a=0, b=2, c=0 */
div.bar {}       /* a=0, b=1, c=1 */
:hover {}        /* a=0, b=1, c=0 */
[title] {}       /* a=0, b=1, c=0 */
.bar {}          /* a=0, b=1, c=0 */
div ul + li {}   /* a=0, b=0, c=3 */
p::after {}      /* a=0, b=0, c=2 */
*::before {}     /* a=0, b=0, c=1 */
::before {}      /* a=0, b=0, c=1 */
div {}           /* a=0, b=0, c=1 */
* {}             /* a=0, b=0, c=0 */
```

### Esempio 2: come viene utilizzata la specificità dal browser

Immagina la seguente implementazione CSS:

```
#foo {
  color: blue;
}

.bar {
  color: red;
  background: black;
}
```

Qui abbiamo un selettore ID che dichiara il `color` come *blu* e un selettore di classe che dichiara il `color` come *rosso* e lo `background` come *nero*.

Un elemento con un ID di `#foo` e una classe di `.bar` saranno selezionati da entrambe le dichiarazioni. I selettori ID hanno una specificità di Gruppo A e i selettori di classe hanno una specificità di Gruppo B. Un selettore ID supera ogni numero di selettori di classe. Per questo `color:blue;` , `color:blue;` dal selettore `#foo` e dallo `background:black;` dal selettore `.bar` verrà applicato all'elemento. La maggiore specificità del selettore ID farà sì che il browser ignori la `.bar` di `color` del selettore `.bar` .

Ora immagina una diversa implementazione CSS:

```
.bar {
  color: red;
  background: black;
}

.baz {
  background: white;
}
```

Qui abbiamo due selettori di classe; uno dei quali dichiara il `color` come *rosso* e lo `background` come *nero* e l'altro dichiara lo `background` come *bianco* .

Un elemento con entrambe le classi `.bar` e `.baz` sarà influenzato da entrambe queste dichiarazioni, tuttavia il problema che abbiamo ora è che sia `.bar` che `.baz` hanno una specificità identica di Gruppo B. La natura a cascata di CSS risolve questo per noi: come `.baz` è definito *dopo* `.bar` , il nostro elemento finisce con il `color` *rosso* da `.bar` ma lo `background` *bianco* da `.baz` .

### Esempio 3: come manipolare la specificità

L'ultimo frammento dell'esempio 2 può essere manipolato per garantire che la nostra `.bar` di `color` del selettore di classe `.bar` venga utilizzata al posto di quella del selettore di classe `.baz` .

```
.bar {}          /* a=0, b=1, c=0 */
.baz {}          /* a=0, b=1, c=0 */
```

Il modo più comune per ottenere questo risultato è scoprire quali altri selettori possono essere applicati alla sequenza del selettore `.bar` . Ad esempio, se la `.bar` classe è stata sempre e solo applicato a `span` elementi, potremmo modificare il `.bar` selettore `span.bar` . Ciò gli darebbe una nuova specificità di Gruppo C , che `.baz` la mancanza del selettore `.baz` :

```
span.bar {}     /* a=0, b=1, c=1 */
.baz {}         /* a=0, b=1, c=0 */
```

Tuttavia, potrebbe non essere sempre possibile trovare un altro selettore comune condiviso tra qualsiasi elemento che utilizza la classe `.bar` . Per questo motivo, il CSS ci consente di duplicare i selettori per aumentare la specificità. Invece di solo `.bar` , possiamo usare invece `.bar.bar` (vedi [La grammatica dei selettori, Raccomandazione W3C](#) ). Questo seleziona ancora qualsiasi elemento con una classe di `.bar` , ma ora ha il doppio della specificità di Gruppo B :

```
.bar.bar {}     /* a=0, b=2, c=0 */
```

```
.baz {} /* a=0, b=1, c=0 */
```

## `!important` Dichiarazioni di stile `!important` e in linea

Il flag `!important` su una dichiarazione di stile e gli stili dichiarati dall'attributo di `style` HTML sono considerati con una maggiore specificità rispetto a qualsiasi selettore. Se questi esistono, la dichiarazione di stile che hanno influenza annullerà altre dichiarazioni indipendentemente dalla loro specificità. Cioè, a meno che tu non abbia più di una dichiarazione che contiene un flag `!important` per la stessa proprietà che si applica allo stesso elemento. Quindi, le normali regole di specificità si applicano a quelle proprietà in riferimento l'una all'altra.

Dal momento che superano completamente la specificità, l'uso di `!important` è disapprovato nella maggior parte dei casi d'uso. Uno dovrebbe usarlo il meno possibile. Per mantenere il codice CSS efficiente e mantenibile a lungo termine, è quasi sempre preferibile aumentare la specificità del selettore circostante che usare `!important`.

Una di quelle rare eccezioni in cui `!important` non è disapprovato, è quando si implementano classi helper generiche come una classe `.hidden` o `.background-yellow` che devono sempre sovrascrivere una o più proprietà ovunque si trovino. E anche allora, devi sapere cosa stai facendo. L'ultima cosa che vuoi, quando scrivi CSS gestibile, è avere `!important` flag in tutto il tuo CSS.

## Un'ultima nota

Un malinteso comune sulla specificità dei CSS è che i valori dei gruppi *A*, *B* e *c* dovrebbero essere combinati tra loro ( $a=1, b=5, c=1 \Rightarrow 151$ ). Questo **non** è il caso. In tal caso, disporre di 20 di un selettore di gruppo *B* o *c* sarebbe sufficiente per ignorare rispettivamente un singolo selettore di gruppo *A* o *B*. I tre gruppi dovrebbero essere considerati come livelli individuali di specificità. La specificità non può essere rappresentata da un singolo valore.

Quando crei il tuo foglio di stile CSS, dovresti mantenere la specificità più bassa possibile. Se è necessario aumentare leggermente la specificità per sovrascrivere un altro metodo, renderlo più alto ma più basso possibile per renderlo più alto. Non dovresti avere bisogno di un selettore come questo:

```
body.page header.container nav div#main-nav li a {}
```

Questo rende le modifiche future più difficili e inquina quella pagina css.

---

Puoi calcolare la specificità del tuo selettore [qui](#)

## Esempio di specificità più complesso

```
div {
  font-size: 7px;
  border: 3px dotted pink;
  background-color: yellow;
  color: purple;
```

```

}

body.mystyle > div.myotherstyle {
  font-size: 11px;
  background-color: green;
}

#elmnt1 {
  font-size: 24px;
  border-color: red;
}

.mystyle .myotherstyle {
  font-size: 16px;
  background-color: black;
  color: red;
}

```

```

<body class="mystyle">
  <div id="elmnt1" class="myotherstyle">
    Hello, world!
  </div>
</body>

```

Quali sono i bordi, i colori e le dimensioni dei caratteri del testo?

dimensione del font:

`font-size: 24;` , poiché la `#elmnt1` regola `#elmnt1` ha la specificità più alta per il `<div>` in questione, ogni proprietà qui è impostata.

confine:

`border: 3px dotted red;` . Il colore del bordo `red` è preso dal set di regole `#elmnt1` , poiché ha la massima specificità. Le altre proprietà del bordo, dello spessore del bordo e dello stile del bordo provengono dalla serie di regole `div` .

colore di sfondo:

`background-color: green;` . Il `background-color` è impostato nelle serie di `body.mystyle > div.myotherstyle div` , `body.mystyle > div.myotherstyle` e `.mystyle .myotherstyle` . Le specificità sono (0, 0, 1) contro (0, 2, 2) vs. (0, 2, 0), quindi il medio "vince".

colore:

`color: red;` . Il colore è impostato in entrambi i `.mystyle .myotherstyle` regole `div` e `.mystyle .myotherstyle` . Quest'ultimo ha la specificità più alta di (0, 2, 0) e "vince".

Leggi Cascading e Specificità online: <https://riptutorial.com/it/css/topic/450/cascading-e-specificita>

---

# Capitolo 7: Centraggio

## Examples

### Usando la trasformazione CSS

Le [trasformazioni CSS](#) sono basate sulla dimensione degli elementi, quindi se non sai quanto è alto o largo il tuo elemento, puoi posizionarlo assolutamente al 50% dalla parte superiore e a sinistra di un contenitore relativo e tradurlo del 50% a sinistra e verso l'alto per centrarlo verticalmente e orizzontalmente.

Tieni presente che con questa tecnica, l'elemento potrebbe terminare con il rendering a un limite di pixel non intero, rendendolo sfocato. Vedere [questa risposta in SO](#) per una soluzione alternativa.

### HTML

```
<div class="container">
  <div class="element"></div>
</div>
```

### CSS

```
.container {
  position: relative;
}

.element {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}
```

[Visualizza l'esempio in JSFiddle](#)

## COMPATIBILITÀ DEL BROWSER CROSS

La proprietà di trasformazione ha bisogno dei prefissi per essere supportati dai browser più vecchi. I prefissi sono necessari per Chrome <= 35, Safari <= 8, Opera <= 22, Browser Android <= 4.4.4 e IE9. Le trasformazioni CSS non sono supportate da IE8 e versioni precedenti.

Ecco una dichiarazione di trasformazione comune per l'esempio precedente:

```
-webkit-transform: translate(-50%, -50%); /* Chrome, Safari, Opera, Android */
-ms-transform: translate(-50%, -50%); /* IE 9 */
transform: translate(-50%, -50%);
```

Per maggiori informazioni vedi [canluse](#) .

## MAGGIORI INFORMAZIONI

- L'elemento viene posizionato in base al primo genitore non statico ( `position: relative` , `absolute` o `fixed` ). Esplora di più in questo [violino](#) e questo [argomento di documentazione](#) .
- Per il centraggio solo orizzontale, usa `left: 50%` e `transform: translateX(-50%)` . Lo stesso vale per il centraggio solo verticale: centro con `top: 50%` e `transform: translateY(-50%)` .
- L'uso di elementi di larghezza / altezza non statici con questo metodo di centraggio può far sì che l'elemento centrato appaia schiacciato. Questo accade principalmente con elementi contenenti testo e può essere corretto aggiungendo: `margin-right: -50%`; e `margin-bottom: -50%`; . Guarda questo [violino](#) per maggiori informazioni.

## Utilizzando Flexbox

### HTML:

```
<div class="container">
  
</div>
```

### CSS:

```
html, body, .container {
  height: 100%;
}
.container {
  display: flex;
  justify-content: center; /* horizontal center */
}
img {
  align-self: center; /* vertical center */
}
```

### [Visualizza risultato](#)

---

### HTML:

```

```

### CSS:

```
html, body {
  height: 100%;
}
body {
  display: flex;
  justify-content: center; /* horizontal center */
}
```

```
align-items: center;    /* vertical center */
}
```

## Visualizza risultato

Vedi [Dynamic Vertical e Horizontal Centering](#) nella documentazione di [Flexbox](#) per maggiori dettagli su flexbox e cosa significano gli stili.

## Supporto del browser

Flexbox è supportato da tutti i principali browser, [tranne le versioni di IE prima di 10](#) .

Alcune versioni recenti del browser, come Safari 8 e IE10, richiedono i [prefissi del fornitore](#) .

Per un modo rapido per generare prefissi c'è [Autoprefixer](#) , uno strumento di terze parti.

Per i browser più vecchi (come IE 8 e 9) è [disponibile](#) un [Polyfill](#) .

Per uno sguardo più dettagliato al supporto del browser Flexbox, vedere [questa risposta](#) .

## Usando la posizione: assoluta

*Lavorare con i vecchi browser (IE >= 8)*

I margini automatici, accoppiati con valori pari a zero per gli offset `left` e `right` o `top` e `bottom` , centrano gli elementi posizionati in modo assoluto all'interno del relativo elemento principale.

## Visualizza risultato

### HTML

```
<div class="parent">
  
</div>
```

### CSS

```
.parent {
  position: relative;
  height: 500px;
}

.center {
  position: absolute;
  margin: auto;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
}
```

Gli elementi che non hanno la loro larghezza e altezza implicite come fanno le immagini, avranno

bisogno di quei valori definiti.

Altre risorse: [centrata assoluta nei CSS](#)

## Tecnica dell'elemento fantasma (hack di Michał Czernow)

Questa tecnica funziona anche quando le dimensioni del contenitore sono sconosciute.

Imposta un elemento "fantasma" all'interno del contenitore che deve essere centrato con altezza al 100%, quindi usa `vertical-align: middle` su entrambi e l'elemento da centrare.

### CSS

```
/* This parent can be any width and height */
.block {
  text-align: center;

  /* May want to do this if there is risk the container may be narrower than the element
  inside */
  white-space: nowrap;
}

/* The ghost element */
.block:before {
  content: '';
  display: inline-block;
  height: 100%;
  vertical-align: middle;

  /* There is a gap between ghost element and .centered,
  caused by space character rendered. Could be eliminated by
  nudging .centered (nudge distance depends on font family),
  or by zeroing font-size in .parent and resetting it back
  (probably to 1rem) in .centered. */
  margin-right: -0.25em;
}

/* The element to be centered, can also be of any width and height */
.centered {
  display: inline-block;
  vertical-align: middle;
  width: 300px;
  white-space: normal; /* Resetting inherited nowrap behavior */
}
```

### HTML

```
<div class="block">
  <div class="centered"></div>
</div>
```

## Utilizzando testo allineato

Il tipo di centratura più comune e più semplice è quello delle righe di testo in un elemento. Il CSS ha la regola `text-align: center` per questo scopo:

## HTML

```
<p>Lorem ipsum</p>
```

## CSS

```
p {
  text-align: center;
}
```

*Questo non funziona per il centraggio di interi elementi di blocco* . allineamento del `text-align` controlla solo l'allineamento del contenuto in linea come il testo nel suo elemento blocco genitore.

Ulteriori informazioni `text-align` nella sezione [Tipografia](#) .

## Centraggio in relazione ad un altro oggetto

Vedremo come centrare il contenuto in base all'altezza di un elemento vicino.

Compatibilità: IE8 +, tutti gli altri browser moderni.

## HTML

```
<div class="content">
  <div class="position-container">
    <div class="thumb">
      
    </div>
    <div class="details">
      <p class="banner-title">text 1</p>
      <p class="banner-text">content content content content content content content content
content content content content content content content content</p>
      <button class="btn">button</button>
    </div>
  </div>
</div>
```

## CSS

```
.content * {
  box-sizing: border-box;
}
.content .position-container {
  display: table;
}
.content .details {
  display: table-cell;
  vertical-align: middle;
  width: 33.333333%;
  padding: 30px;
  font-size: 17px;
  text-align: center;
}
.content .thumb {
```

```
width: 100%;
}
.content .thumb img {
width: 100%;
}
```

Collegamento a [JSFiddle](#)

I punti principali sono i `.thumb` `3` `.thumb` , `.details` e `.position-container` container:

- Il `.position-container` deve avere `display: table` .
- I `.details` devono avere la larghezza di larghezza reale `width: ...` e `display: table-cell` , `vertical-align: middle` .
- Il `.thumb` deve avere `width: 100%` se vuoi che `.thumb` tutto lo spazio rimanente e sarà influenzato dalla larghezza `.details` .
- L'immagine (se hai un'immagine) all'interno `.thumb` dovrebbe avere `width: 100%` , ma non è necessario se hai le proporzioni corrette.

## Allineare verticalmente qualsiasi cosa con 3 linee di codice

Supportato da IE11 +

[Visualizza risultato](#)

Usa queste 3 linee per allineare verticalmente praticamente tutto. Assicurati che l'immagine / div che applichi il codice abbia un genitore con un'altezza.

### CSS

```
div.vertical {
position: relative;
top: 50%;
transform: translateY(-50%);
}
```

### HTML

```
<div class="vertical">Vertical aligned text!</div>
```

## Allinea verticalmente un'immagine all'interno div

### HTML

```
<div class="wrap">
  
</div>
```

### CSS

```

.wrap {
  height: 50px; /* max image height */
  width: 100px;
  border: 1px solid blue;
  text-align: center;
}
.wrap:before {
  content: "";
  display: inline-block;
  height: 100%;
  vertical-align: middle;
  width: 1px;
}
img {
  vertical-align: middle;
}

```

## Centrata orizzontale e verticale usando la disposizione della tabella

Si potrebbe facilmente centrare un elemento figlio usando la proprietà di visualizzazione `table`.

### HTML

```

<div class="wrapper">
  <div class="parent">
    <div class="child"></div>
  </div>
</div>

```

### CSS

```

.wrapper {
  display: table;
  vertical-align: center;
  width: 200px;
  height: 200px;
  background-color: #9e9e9e;
}
.parent {
  display: table-cell;
  vertical-align: middle;
  text-align: center;
}
.child {
  display: inline-block;
  vertical-align: middle;
  text-align: center;
  width: 100px;
  height: 100px;
  background-color: teal;
}

```

## Utilizzo di `calc ()`

La funzione `calc ()` è la parte di una nuova sintassi in CSS3 in cui puoi calcolare

(matematicamente) quale dimensione / posizione occupa il tuo elemento usando una varietà di valori come pixel, percentuali, ecc. Nota: - Ogni volta che usi questa funzione , `calc(100% - 80px)` sempre attenzione allo spazio tra due valori `calc(100% - 80px)` .

## CSS

```
.center {
  position: absolute;
  height: 50px;
  width: 50px;
  background: red;
  top: calc(50% - 50px / 2); /* height divided by 2*/
  left: calc(50% - 50px / 2); /* width divided by 2*/
}
```

## HTML

```
<div class="center"></div>
```

## Allinea verticalmente gli elementi di altezza dinamica

L'applicazione intuitiva di css non produce i risultati desiderati perché

- `vertical-align:middle` **non è applicabile agli elementi a livello di blocco**
- `margin-top:auto` e `margin-bottom:auto` **valori** `margin-bottom:auto` **utilizzati dovrebbero essere calcolati come zero**
- `margin-top:-50%` **valori di margine basati sulla percentuale vengono calcolati in relazione alla larghezza del blocco contenitore**

Per il supporto del browser più ampio, una soluzione alternativa con gli elementi helper:

## HTML

```
<div class="vcenter--container">
  <div class="vcenter--helper">
    <div class="vcenter--content">
      <!--stuff-->
    </div>
  </div>
</div>
```

## CSS

```
.vcenter--container {
  display: table;
  height: 100%;
  position: absolute;
  overflow: hidden;
  width: 100%;
}
.vcenter--helper {
  display: table-cell;
```

```
vertical-align: middle;
}
.vcenter--content {
margin: 0 auto;
width: 200px;
}
```

[jsfiddle](#) dalla [domanda originale](#) . Questo approccio

- funziona con elementi di altezza dinamici
- rispetta il flusso di contenuti
- è supportato dai browser legacy

## Utilizzando l'altezza della linea

Puoi anche usare l' `line-height` per centrare verticalmente una singola riga di testo all'interno di un contenitore:

### CSS

```
div {
height: 200px;
line-height: 200px;
}
```

È piuttosto brutto, ma può essere utile all'interno di un elemento `<input />` . La proprietà `line-height` funziona solo quando il testo da centrare si estende su una singola riga. Se il testo viene disposto su più righe, l'output risultante non sarà centrato.

## Centrata verticale e orizzontale senza preoccuparsi di altezza o larghezza

La seguente tecnica ti consente di aggiungere il tuo contenuto a un elemento HTML e centrarlo sia orizzontalmente che verticalmente **senza preoccuparti della sua altezza o larghezza** .

## Il contenitore esterno

- dovrebbe avere `display: table;`

## Il contenitore interno

- dovrebbe avere `display: table-cell;`
- dovrebbe avere `vertical-align: middle;`
- dovrebbe avere `text-align: center;`

## La scatola dei contenuti

- dovrebbe avere `display: inline-block;`
- dovrebbe riaggiustare l'allineamento orizzontale del testo ad es. `text-align: left;` o `text-`

`align: right;` , a meno che tu non voglia che il testo sia centrato

## dimostrazione

### HTML

```
<div class="outer-container">
  <div class="inner-container">
    <div class="centered-content">
      You can put anything here!
    </div>
  </div>
</div>
```

### CSS

```
body {
  margin : 0;
}

.outer-container {
  position : absolute;
  display: table;
  width: 100%; /* This could be ANY width */
  height: 100%; /* This could be ANY height */
  background: #ccc;
}

.inner-container {
  display: table-cell;
  vertical-align: middle;
  text-align: center;
}

.centered-content {
  display: inline-block;
  text-align: left;
  background: #fff;
  padding: 20px;
  border: 1px solid #000;
}
```

Vedi anche [questo Fiddle](#) !

## Centrata con dimensioni fisse

Se la dimensione dei tuoi contenuti è fissa, puoi utilizzare il posizionamento assoluto al 50% con `margin` che riduce la metà della larghezza e dell'altezza del tuo contenuto:

### HTML

```
<div class="center">
  Center vertically and horizontally
</div>
```

## CSS

```
.center {
  position: absolute;
  background: #ccc;

  left: 50%;
  width: 150px;
  margin-left: -75px; /* width * -0.5 */

  top: 50%;
  height: 200px;
  margin-top: -100px; /* height * -0.5 */
}
```

---

## Centrata orizzontale con larghezza fissa

Puoi centrare l'elemento orizzontalmente anche se non conosci l'altezza del contenuto:

## HTML

```
<div class="center">
  Center only horizontally
</div>
```

## CSS

```
.center {
  position: absolute;
  background: #ccc;

  left: 50%;
  width: 150px;
  margin-left: -75px; /* width * -0.5 */
}
```

---

## Centrata verticale con altezza fissa

Puoi centrare l'elemento verticalmente se conosci l'altezza dell'elemento:

## HTML

```
<div class="center">
  Center only vertically
</div>
```

## CSS

```
.center {
  position: absolute;
  background: #ccc;
```

```
top: 50%;
height: 200px;
margin-top: -100px; /* width * -0.5 */
}
```

## Utilizzo del margine: 0 auto;

Gli oggetti possono essere centrati usando il `margin: 0 auto;` se sono elementi di blocco e hanno una larghezza definita.

## HTML

```
<div class="containerDiv">
  <div id="centeredDiv"></div>
</div>

<div class="containerDiv">
  <p id="centeredParagraph">This is a centered paragraph.</p>
</div>

<div class="containerDiv">
  
</div>
```

## CSS

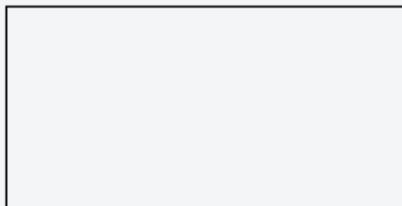
```
.containerDiv {
  width: 100%;
  height: 100px;
  padding-bottom: 40px;
}

#centeredDiv {
  margin: 0 auto;
  width: 200px;
  height: 100px;
  border: 1px solid #000;
}

#centeredParagraph {
  width: 200px;
  margin: 0 auto;
}

#centeredImage {
  display: block;
  width: 200px;
  margin: 0 auto;
}
```

Risultato:



This is a centered paragraph.



Esempio JSFiddle: [Centrare gli oggetti con margine: 0 auto;](#)

Leggi Centraggio online: <https://riptutorial.com/it/css/topic/299/centraggio>

---

# Capitolo 8: Centratatura verticale

## Osservazioni

Questo viene utilizzato quando le dimensioni dell'elemento ( `width` e `height` ) non sono note o dinamiche.

Preferisci utilizzare **Flexbox** su tutte le altre opzioni poiché è ottimizzato per la progettazione dell'interfaccia utente.

## Examples

### Centratatura con `display: tabella`

HTML:

```
<div class="wrapper">
  <div class="outer">
    <div class="inner">
      centered
    </div>
  </div>
</div>
```

CSS:

```
.wrapper {
  height: 600px;
  text-align: center;
}
.outer {
  display: table;
  height: 100%;
  width: 100%;
}
.outer .inner {
  display: table-cell;
  text-align: center;
  vertical-align: middle;
}
```

### Centratatura con Trasforma

HTML:

```
<div class="wrapper">
  <div class="centered">
    centered
  </div>
</div>
```

## CSS:

```
.wrapper {
  position: relative;
  height: 600px;
}
.centered {
  position: absolute;
  z-index: 999;
  transform: translate(-50%, -50%);
  top: 50%;
  left: 50%;
}
```

## Centraggio con Flexbox

### HTML:

```
<div class="container">
  <div class="child"></div>
</div>
```

### CSS:

```
.container {
  height: 500px;
  width: 500px;
  display: flex;           // Use Flexbox
  align-items: center;     // This centers children vertically in the parent.
  justify-content: center; // This centers children horizontally.
  background: white;
}

.child {
  width: 100px;
  height: 100px;
  background: blue;
}
```

## Centrata del testo con l'altezza della linea

### HTML:

```
<div class="container">
  <span>vertically centered</span>
</div>
```

### CSS:

```
.container{
  height: 50px;           /* set height */
  line-height: 50px;     /* set line-height equal to the height */
  vertical-align: middle; /* works without this rule, but it is good having it explicitly set */
}
```

```
}
```

**Nota:** questo metodo centrerà solo verticalmente una *singola riga di testo* . Non centrerà correttamente gli elementi di blocco e se il testo si interrompe su una nuova riga, avrai due righe di testo molto alte.

## Centrata con posizione: assoluta

HTML:

```
<div class="wrapper">
  
</div>
```

CSS:

```
.wrapper{
  position:relative;
  height: 600px;
}
.wrapper img {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  margin: auto;
}
```

Se vuoi centrare altre immagini, allora devi dare altezza e larghezza a quell'elemento.

HTML:

```
<div class="wrapper">
  <div class="child">
    make me center
  </div>
</div>
```

CSS:

```
.wrapper{
  position:relative;
  height: 600px;
}
.wrapper .child {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  margin: auto;
  width: 200px;
}
```

```
height: 30px;
border: 1px solid #f00;
}
```

## Centrata con pseudo elemento

### HTML:

```
<div class="wrapper">
  <div class="content"></div>
</div>
```

### CSS:

```
.wrapper{
  min-height: 600px;
}

.wrapper:before{
  content: "";
  display: inline-block;
  height: 100%;
  vertical-align: middle;
}

.content {
  display: inline-block;
  height: 80px;
  vertical-align: middle;
}
```

Questo metodo è utilizzato al meglio nei casi in cui si ha un'altezza `.content` centrato all'interno del `.wrapper` ; e si desidera `.wrapper` 'altezza s per espandere quando `.content` ' altezza s superi `.wrapper` `min-height` 's.

Leggi Centrata verticale online: <https://riptutorial.com/it/css/topic/5070/centrata-verticale>

---

# Capitolo 9: colonne

## Sintassi

- numero di colonne: auto | numero | inherit | initial | unset;
- larghezza della colonna: auto | lunghezza;
- column: [column-width] | [column-count];
- column-span: none | all | inherit | initial | unset;
- column-gap: normal | length | inherit | initial | unset;
- column-fill: auto | balance | inherit | initial | unset;
- column-rule-color: color | inherit | initial | unset;
- column-rule-style: none | hidden | punteggiato | dashed | solid | double | groove | ridge | inset | outset | inherit | initial | unset;
- column-rule-width: thin | medium | thick | length | inherit | initial | unset;
- column-rule: [column-rule-width] | [column-rule-style] | [column-rule-color];
- break-after: auto | always | left | right | recto | verso | page | column | region | avoid | avoid-page | avoid-column | avoid-region;
- break-before: auto | always | left | right | recto | verso | page | column | region | avoid | avoid-page | avoid-column | avoid-region;
- break-inside: auto | avoid | avoid-page | avoid-column | avoid-region;

## Examples

### Esempio semplice (numero di colonne)

Il layout multi-colonna CSS semplifica la creazione di più colonne di testo.

### Codice

```
<div id="multi-columns">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum</div>
```

```
.multi-columns {  
  -moz-column-count: 2;  
  -webkit-column-count: 2;  
  column-count: 2;  
}
```

### Risultato

Stack Overflow is a privately held website, the flagship site of the Stack Exchange Network,[4][5][6] created in 2008 by Jeff Atwood and Joel Spolsky. [7][8] It was created to be a more open alternative to earlier Q&A sites such as Experts-Exchange. The name for the website was chosen by voting in April 2008 by readers of Coding Horror, Atwood's popular programming blog.

The website serves as a platform for users to ask and answer questions, and, through membership and active participation, to vote questions and

answers and answers wiki or Di Overflow "badges" awarded receiving given to a badges fo [14] which gamificat or forum. licensed Attribute-

## Larghezza della colonna

La proprietà `column-width` imposta la larghezza minima della colonna. Se il `column-count` non è definito, il browser eseguirà tutte le colonne che rientrano nella larghezza disponibile.

### Codice:

```
<div id="multi-columns">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum
</div>
```

```
.multi-columns {
  -moz-column-width: 100px;
  -webkit-column-width: 100px;
  column-width: 100px;
}
```

## Risultato

|  |  |  |  |  |
|--|--|--|--|--|
| Stack Overflow is a privately held website, the flagship site of the Stack Exchange Network,[4][5][6] created in 2008 by Jeff Atwood and Joel Spolsky. [7][8] It was created to be a more open alternative to earlier Q&A sites such as Experts- | Exchange. The name for the website was chosen by voting in April 2008 by readers of Coding Horror, Atwood's popular programming blog. The website serves as a platform for users to ask and answer | questions, and, through membership and active participation, to vote questions and answers up or down and edit questions and answers in a fashion similar to a wiki or Digg.[13] Users of Stack Overflow can earn reputation | points and "badges"; for example, a person is awarded 10 reputation points for receiving an "up" vote on an answer given to a question, and can receive badges for their valued contributions, [14] which represents a kind of | gamification of the traditional Q&A site or forum. All user-generated content is licensed under a Creative Commons Attribute-ShareAlike license. |
|--|--|--|--|--|

Leggi colonne online: <https://riptutorial.com/it/css/topic/3042/colonne>

---

# Capitolo 10: Colonne multiple

## introduzione

Il CSS consente di definire che i contenuti degli elementi si avvolgono in più colonne con lacune e regole tra di loro.

## Osservazioni

[Modulo di layout multi-colonna CSS di livello 1](#) è, a partire dal 12 aprile 2011, una raccomandazione del Candidato W3C. Da allora, [sono state apportate](#) alcune [modifiche minori](#) . È considerato in [fase stabile](#) .

A partire dal 3 luglio 2017, i browser Internet Explorer 10 e 11 di Microsoft e Edge supportano solo una versione precedente delle specifiche utilizzando un prefisso del fornitore.

## Examples

### Esempio di base

Considera il seguente markup HTML:

```
<section>
  <p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor
  invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et
  justo duo dolores et ea rebum.</p>
  <p> Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem
  ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut
  labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo
  dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor
  sit amet.</p>
  <p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor
  invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et
  justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
  ipsum dolor sit amet.</p>
</section>
```

Con il seguente CSS applicato, il contenuto è diviso in tre colonne separate da una colonna grigia di due pixel.

```
section {
  columns: 3;
  column-gap: 40px;
  column-rule: 2px solid gray;
}
```

Guarda un [esempio dal vivo di questo su JSFiddle](#) .

## Crea più colonne

```
<div class="content">
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh
eismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim
ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl
ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in
hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu
feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui
blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla
facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil
imperdiet doming id quod mazim placerat facer possim assum.
</div>
```

## Css

```
.content {
-webkit-column-count: 3; /* Chrome, Safari, Opera */
-moz-column-count: 3; /* Firefox */
column-count: 3;
}
```

Leggi Colonne multiple online: <https://riptutorial.com/it/css/topic/10688/colonne-multiple>

# Capitolo 11: Colori

## Sintassi

- colore: #rgb
- colore: #rrggbb
- color: rgb [a] (<rosso>, <verde>, <blu> [, <alpha>])
- colore: hsl [a] (<tonalità>, <saturazione%>, <luminosità%> [, <alpha>])
- colore: **colorkeyword** / \* verde, blu, giallo, arancione, rosso, ..etc \* /

## Examples

### Parole chiave di colore

La maggior parte dei browser supporta l'utilizzo di parole chiave a colori per specificare un colore. Ad esempio, per impostare il `color` di un elemento su blu, utilizzare la parola chiave `blue` :

```
.some-class {  
  color: blue;  
}
```

Le parole chiave CSS non sono case sensitive- `blue` , `Blue` e `BLUE` saranno tutti risultato `#0000FF` .

## Parole chiave di colore

| Nome del colore | Valore esadecimale | Valori RGB        | Colore  |
|-----------------|--------------------|-------------------|---|
| AliceBlue       | # F0F8FF           | rgb (240.248.255) |  |
| AntiqueWhite    | # FAEBD7           | rgb (250.235.215) |  |
| acqua           | # 00FFFF           | rgb (0.255.255)   |  |
| Acquamarina     | # 7FFFD4           | rgb (127.255.212) |  |
| Azzurro         | # F0FFFF           | rgb (240.255.255) |  |
| Beige           | # F5F5DC           | rgb (245.245.220) |  |
| biscotto        | # FFE4C4           | rgb (255.228.196) |  |
| Nero            | # 000000           | rgb (0,0,0)       |  |

| Nome del colore | Valore esadecimale | Valori RGB        | Colore  |
|-----------------|--------------------|-------------------|---|
| BlanchedAlmond  | #FFEBCD            | rgb (255.235.205) |    |
| Blu             | # 0000FF           | rgb (0,0,255)     |    |
| Blu viola       | # 8A2BE2           | rgb (138,43,226)  |    |
| Marrone         | # A52A2A           | rgb (165,42,42)   |    |
| burlywood       | # DEB887           | rgb (222.184.135) |    |
| CadetBlue       | # 5F9EA0           | rgb (95.158.160)  |    |
| certosa         | # 7FFF00           | rgb (127,255,0)   |    |
| Cioccolato      | # D2691E           | rgb (210,105,30)  |    |
| Corallo         | # FF7F50           | rgb (255,127,80)  |    |
| cornflowerblue  | # 6495ED           | rgb (100.149.237) |   |
| Cornsilk        | # FFF8DC           | rgb (255.248.220) |  |
| Cremisi         | # DC143C           | rgb (220,20,60)   |  |
| Ciano           | # 00FFFF           | rgb (0.255.255)   |  |
| Blu scuro       | # 00008B           | rgb (0,0,139)     |  |
| DarkCyan        | # 008B8B           | rgb (0.139.139)   |  |
| DarkGoldenRod   | # B8860B           | rgb (184,134,11)  |  |
| Grigio scuro    | # A9A9A9           | rgb (169.169.169) |  |
| Grigio scuro    | # A9A9A9           | rgb (169.169.169) |  |
| Verde scuro     | # 006.400          | rgb (0,100,0)     |  |
| DarkKhaki       | # BDB76B           | rgb (189.183.107) |  |
| DarkMagenta     | # 8B008B           | rgb (139,0,139)   |  |

| Nome del colore | Valore esadecimale | Valori RGB        | Colore  |
|-----------------|--------------------|-------------------|---|
| DarkOliveGreen  | # 556B2F           | rgb (85,107,47)   |    |
| Arancione scuro | # FF8C00           | rgb (255,140,0)   |    |
| DarkOrchid      | # 9932CC           | rgb (153,50,204)  |    |
| Rosso scuro     | # 8B0000           | rgb (139,0,0)     |    |
| DarkSalmon      | # E9967A           | rgb (233.150.122) |    |
| DarkSeaGreen    | # 8FBC8F           | rgb (143.188.143) |    |
| darkslateblue   | # 483D8B           | rgb (72,61,139)   |    |
| Darkslategray   | # 2F4F4F           | rgb (47,79,79)    |    |
| DarkSlateGrey   | # 2F4F4F           | rgb (47,79,79)    |   |
| DarkTurquoise   | # 00CED1           | rgb (0.206.209)   |  |
| Darkviolet      | # 9400D3           | rgb (148,0,211)   |  |
| Rosa scuro      | # FF1493           | rgb (255,20,147)  |  |
| DeepSkyBlue     | # 00BFFF           | rgb (0.191.255)   |  |
| dimgray         | # 696969           | rgb (105.105.105) |  |
| DimGrey         | # 696969           | rgb (105.105.105) |  |
| Blu Dodger      | # 1E90FF           | rgb (30.144.255)  |  |
| firebrick       | # B22222           | rgb (178,34,34)   |  |
| FloralWhite     | # FFFAF0           | rgb (255.250.240) |  |
| Verde foresta   | # 228B22           | rgb (34,139,34)   |  |
| Fucsia          | # FF00FF           | rgb (255,0,255)   |  |
| Gainsboro       | #DCDCDC            | rgb (220.220.220) |  |

| Nome del colore      | Valore esadecimale | Valori RGB        | Colore  |
|----------------------|--------------------|-------------------|---|
| GhostWhite           | # F8F8FF           | rgb (248.248.255) |    |
| Oro                  | # FFD700           | rgb (255,215,0)   |    |
| Verga d'oro          | # DAA520           | rgb (218,165,32)  |    |
| Grigio               | # 808080           | rgb (128,128,128) |    |
| Grigio               | # 808080           | rgb (128,128,128) |    |
| verde                | # 008000           | rgb (0,128,0)     |    |
| Verde giallo         | # ADFF2F           | rgb (173,255,47)  |    |
| Melata               | # F0FFF0           | rgb (240.255.240) |    |
| Rosa caldo           | # FF69B4           | rgb (255.105.180) |   |
| IndianRed            | # CD5C5C           | rgb (205,92,92)   |  |
| Indaco               | # 4B0082           | rgb (75,0,130)    |  |
| Avorio               | # FFFFF0           | rgb (255.255.240) |  |
| Cachi                | # F0E68C           | rgb (240.230.140) |  |
| Lavanda              | # E6E6FA           | rgb (230.230.250) |  |
| LavenderBlush        | # FFF0F5           | rgb (255.240.245) |  |
| Lawngreen            | # 7CFC00           | rgb (124,252,0)   |  |
| limone crema         | #FFFACD            | rgb (255.250.205) |  |
| Azzurro              | # ADD8E6           | rgb (173.216.230) |  |
| LightCoral           | # F08080           | rgb (240.128.128) |  |
| lightcyan            | # E0FFFF           | rgb (224.255.255) |  |
| LightGoldenRodYellow | # FAFAD2           | rgb (250.250.210) |  |

| Nome del colore  | Valore esadecimale | Valori RGB        | Colore  |
|------------------|--------------------|-------------------|---|
| Grigio chiaro    | # D3D3D3           | rgb (211.211.211) |    |
| Grigio chiaro    | # D3D3D3           | rgb (211.211.211) |    |
| Verde chiaro     | # 90EE90           | rgb (144.238.144) |    |
| Rosa chiaro      | # FFB6C1           | rgb (255.182.193) |    |
| lightsalmon      | # FFA07A           | rgb (255.160.122) |    |
| lightseagreen    | # 20B2AA           | rgb (32.178.170)  |    |
| Lightskyblue     | # 87CEFA           | rgb (135.206.250) |    |
| LightSlateGray   | # 778.899          | rgb (119.136.153) |    |
| LightSlateGrey   | # 778.899          | rgb (119.136.153) |    |
| LightSteelBlue   | # B0C4DE           | rgb (176.196.222) |   |
| Giallo chiaro    | # FFFFE0           | rgb (255.255.224) |  |
| Lime             | # 00FF00           | rgb (0,255,0)     |  |
| Verde lime       | # 32CD32           | rgb (50,205,50)   |  |
| Biancheria       | # FAF0E6           | rgb (250.240.230) |  |
| Magenta          | # FF00FF           | rgb (255,0,255)   |  |
| Marrone          | # 800000           | rgb (128,0,0)     |  |
| MediumAquaMarine | # 66CDAA           | rgb (102.205.170) |  |
| mediumblue       | # 0000CD           | rgb (0,0,205)     |  |
| MediumOrchid     | # BA55D3           | rgb (186,85,211)  |  |
| MediumPurple     | # 9370DB           | rgb (147.112.219) |  |
| MediumSeaGreen   | # 3CB371           | rgb (60.179.113)  |  |

| Nome del colore   | Valore esadecimale | Valori RGB        | Colore  |
|-------------------|--------------------|-------------------|---|
| MediumSlateBlue   | # 7B68EE           | rgb (123.104.238) |    |
| MediumSpringGreen | # 00FA9A           | rgb (0.250.154)   |    |
| MediumTurquoise   | # 48D1CC           | rgb (72.209.204)  |    |
| MediumVioletRed   | # C71585           | rgb (199,21,133)  |    |
| Blu notte         | # 191970           | rgb (25,25,112)   |    |
| MintCream         | # F5FFFA           | rgb (245.255.250) |    |
| Mistyrose         | # FFE4E1           | rgb (255.228.225) |    |
| Mocassino         | # FFE4B5           | rgb (255.228.181) |    |
| navajowhite       | #FFDEAD            | rgb (255.222.173) |   |
| Marina Militare   | # 000080           | rgb (0,0,128)     |  |
| OldLace           | # FDF5E6           | rgb (253.245.230) |  |
| Oliva             | # 808.000          | rgb (128,128,0)   |  |
| Olivedrab         | # 6B8E23           | rgb (107,142,35)  |  |
| arancia           | # FFA500           | rgb (255,165,0)   |  |
| Rosso-arancio     | # FF4500           | rgb (255,69,0)    |  |
| Orchidea          | # DA70D6           | rgb (218.112.214) |  |
| PaleGoldenRod     | # EEE8AA           | rgb (238.232.170) |  |
| Verde pallido     | # 98FB98           | rgb (152.251.152) |  |
| PaleTurquoise     | #AFEEEE            | rgb (175.238.238) |  |
| PaleVioletRed     | # DB7093           | rgb (219.112.147) |  |
| PapayaWhip        | # FFEFD5           | rgb (255.239.213) |  |

| Nome del colore | Valore esadecimale | Valori RGB        | Colore  |
|-----------------|--------------------|-------------------|---|
| peachpuff       | # FFDAB9           | rgb (255.218.185) |    |
| Perù            | # CD853F           | rgb (205,133,63)  |    |
| Rosa            | # FFC0CB           | rgb (255.192.203) |    |
| prugna          | # DDA0DD           | rgb (221.160.221) |    |
| Powderblue      | # B0E0E6           | rgb (176.224.230) |    |
| Viola           | # 800080           | rgb (128,0,128)   |    |
| RebeccaPurple   | # 663399           | rgb (102,51,153)  |    |
| Rosso           | # FF0000           | rgb (255,0,0)     |    |
| RosyBrown       | # BC8F8F           | rgb (188.143.143) |    |
| Blu reale       | # 4169E1           | rgb (65.105.225)  |   |
| SaddleBrown     | # 8B4513           | rgb (139,69,19)   |  |
| salmone         | # FA8072           | rgb (250.128.114) |  |
| Sandybrown      | # F4A460           | rgb (244,164,96)  |  |
| Mare verde      | # 2E8B57           | rgb (46,139,87)   |  |
| Conchiglia      | # FFF5EE           | rgb (255.245.238) |  |
| terra di siena  | # A0522D           | rgb (160,82,45)   |  |
| Argento         | # C0C0C0           | rgb (192,192,192) |  |
| Cielo blu       | # 87CEEB           | rgb (135.206.235) |  |
| Blu ardesia     | # 6A5ACD           | rgb (106,90,205)  |  |
| Slategray       | # 708.090          | rgb (112.128.144) |  |
| Grigio ardesia  | # 708.090          | rgb (112.128.144) |  |

| Nome del colore | Valore esadecimale | Valori RGB        | Colore  |
|-----------------|--------------------|-------------------|---|
| La neve         | #FFFAFA            | rgb (255.250.250) |    |
| Verde primavera | # 00FF7F           | rgb (0.255.127)   |    |
| Blu acciaio     | # 4682B4           | rgb (70.130.180)  |    |
| abbronzatura    | # D2B48C           | rgb (210.180.140) |    |
| Teal            | # 008080           | rgb (0.128.128)   |    |
| Cardo           | # D8BFD8           | rgb (216.191.216) |    |
| Pomodoro        | # FF6347           | rgb (255,99,71)   |    |
| Turchese        | # 40E0D0           | rgb (64.224.208)  |    |
| Viola           | # EE82EE           | rgb (238.130.238) |   |
| Grano           | # F5DEB3           | rgb (245.222.179) |  |
| bianca          | #FFFFFF            | rgb (255,255,255) |  |
| Fumo bianco     | # F5F5F5           | rgb (245.245.245) |  |
| Giallo          | # FFFF00           | rgb (255,255,0)   |  |
| Giallo verde    | # 9ACD32           | rgb (154,205,50)  |  |

Oltre ai colori con nome, c'è anche la parola chiave `transparent`, che rappresenta un nero completamente trasparente: `rgba(0,0,0,0)`

## Valore esadecimale

### sfondo

I colori CSS possono anche essere rappresentati come una tripletta esadecimale, in cui i membri rappresentano i componenti rosso, verde e blu di un colore. Ciascuno di questi valori rappresenta un numero compreso tra `00` e `FF` o tra `0` e `255` nella notazione decimale. Possono essere usati valori esadecimali maiuscoli e / o minuscoli (es. `#3fc` = `#3FC` = `#33ffcc`). Il browser interpreta `#369` come `#336699`. Se questo non è ciò che intendi, ma preferisci `#306090`, devi specificarlo

esplicitamente.

Il numero totale di colori che possono essere rappresentati con la notazione esadecimale è  $256^3$  o 16.777.216.

## Sintassi

```
color: #rrggbb;  
color: #rgb
```

| Valore | Descrizione                      |
|--------|----------------------------------|
| rr     | 00 - FF per la quantità di rosso |
| gg     | 00 - FF per la quantità di verde |
| bb     | 00 - FF per la quantità di blu   |

```
.some-class {  
  /* This is equivalent to using the color keyword 'blue' */  
  color: #0000FF;  
}  
  
.also-blue {  
  /* If you want to specify each range value with a single number, you can!  
  This is equivalent to '#0000FF' (and 'blue') */  
  color: #00F;  
}
```

La [notazione esadecimale](#) viene utilizzata per specificare i valori dei colori nel formato colore RGB, in base ai ["valori numerici dei colori"](#) del [W3C](#) .

Ci sono molti strumenti disponibili su Internet per cercare valori di colore esadecimale (o semplicemente esadecimale).

Cerca **"tavolozza colori esadecimale"** o **"selettore colori esadecimale"** con il tuo browser web preferito per trovare un sacco di opzioni!

I valori esadecimale iniziano sempre con un cancelletto (#), hanno un massimo di sei "cifre" e non fanno distinzione tra maiuscole e minuscole: cioè, non si preoccupano delle lettere maiuscole.

#FFC125 e #ffc125 sono dello stesso colore.

### rgb () Notazione

RGB è un modello di colore additivo che rappresenta i colori come miscele di luce rossa, verde e blu. In sostanza, la rappresentazione RGB è l'equivalente decimale della notazione esadecimale. In esadecimale, ogni numero varia da 00-FF che equivale a 0-255 in decimale e 0% -100% in percentuale.

```

.some-class {
  /* Scalar RGB, equivalent to 'blue'*/
  color: rgb(0, 0, 255);
}

.also-blue {
  /* Percentile RGB values*/
  color: rgb(0%, 0%, 100%);
}

```

## Sintassi

```
rgb(<red>, <green>, <blue>)
```

| Valore  | Descrizione   |
|---------|---|
| <red>   | un numero intero compreso tra 0 e 255 o percentuale compreso tra 0 e 100% |
| <green> | un numero intero compreso tra 0 e 255 o percentuale compreso tra 0 e 100% |
| <blue>  | un numero intero compreso tra 0 e 255 o percentuale compreso tra 0 e 100% |

## hsl () Notazione

HSL sta per **tonalità** ("che colore"), **saturazione** ("quanto colore") e **leggerezza** ("quanto bianco").

La tonalità è rappresentata da un angolo compreso tra 0 ° e 360 ° (senza unità), mentre la saturazione e la luminosità sono rappresentate come percentuali.

```

p {
  color: hsl(240, 100%, 50%); /* Blue */
}

```

## Sintassi

```
color: hsl(<hue>, <saturation>%, <lightness>%);
```

| Valore       | Descrizione   |
|--------------|---|
| <hue>        | specificato in gradi attorno alla ruota dei colori (senza unità), dove 0 ° è rosso, 60 ° è giallo, 120 ° è verde, 180 ° è ciano, 240 ° è blu, 300 ° è magenta e 360 ° è rosso |
| <saturation> | specificato in percentuale dove lo 0% è completamente desaturato (scala di grigi) e il 100% è completamente saturo (colori vividi)  |

| Valore      | Descrizione   |
|-------------|---|
| <lightness> | specificato in percentuale dove 0% è completamente nero e 100% è completamente bianco |

## Gli appunti

- Una saturazione dello 0% produce sempre un colore in scala di grigi; cambiare la tonalità non ha alcun effetto.
- Una luminosità dello 0% produce sempre il nero e il 100% produce sempre il bianco; cambiare la tonalità o la saturazione non ha alcun effetto.

### currentColor

`currentColor` restituisce il valore del colore calcolato dell'elemento corrente.

## Utilizzare nello stesso elemento

Qui `currentColor` viene valutato in rosso poiché la proprietà `color` è impostata su `red` :

```
div {
  color: red;
  border: 5px solid currentColor;
  box-shadow: 0 0 5px currentColor;
}
```

In questo caso, specificare `currentColor` per il bordo è molto probabilmente ridondante perché ometterlo dovrebbe produrre risultati identici. Usa `currentColor` solo all'interno della proprietà `border` all'interno dello stesso elemento se verrebbe sovrascritto altrimenti a causa di un selettore [più specifico](#) .

Poiché è il colore calcolato, il bordo sarà verde nell'esempio seguente a causa della seconda regola che ignora il primo:

```
div {
  color: blue;
  border: 3px solid currentColor;
  color: green;
}
```

## Ereditato dall'elemento padre

Il colore del genitore è ereditato, qui `currentColor` viene valutato in 'blu', rendendo blu il colore del bordo dell'elemento figlio.

```
.parent-class {
```

```

    color: blue;
}

.parent-class .child-class {
    border-color: currentColor;
}

```

currentColor può anche essere utilizzato da altre regole che normalmente non ereditano dalla proprietà color, come ad esempio il colore di sfondo. L'esempio seguente mostra i bambini che usano il set di colori nel genitore come sfondo:

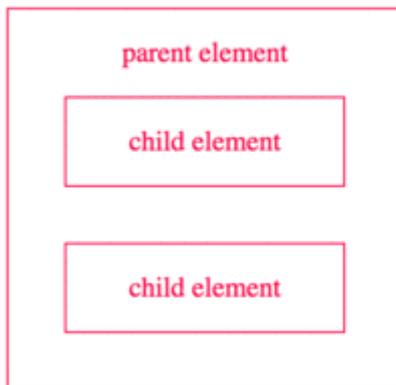
```

.parent-class {
    color: blue;
}

.parent-class .child-class {
    background-color: currentColor;
}

```

### Possibile risultato:



### rgba () Notazione

Simile alla [notazione rgb \(\)](#) , ma con un valore alfa (opacità) aggiuntivo.

```

.red {
    /* Opaque red */
    color: rgba(255, 0, 0, 1);
}

.red-50p {
    /* Half-translucent red. */
    color: rgba(255, 0, 0, .5);
}

```

# Sintassi

```
rgba(<red>, <green>, <blue>, <alpha>);
```

| Valore  | Descrizione  |
|---------|--|
| <red>   | un numero intero compreso tra 0 e 255 o percentuale compreso tra 0 e 100%                      |
| <green> | un numero intero compreso tra 0 e 255 o percentuale compreso tra 0 e 100%                      |
| <blue>  | un numero intero compreso tra 0 e 255 o percentuale compreso tra 0 e 100%                      |
| <alpha> | un numero compreso tra 0 e 1, dove 0.0 è completamente trasparente e 1.0 è completamente opaco |

## hsla () Notazione

Simile alla [notazione hsl \(\)](#) , ma con un valore alfa (opacità) aggiunto.

```
hsla(240, 100%, 50%, 0)    /* transparent */  
hsla(240, 100%, 50%, 0.5) /* half-translucent blue */  
hsla(240, 100%, 50%, 1)   /* fully opaque blue */
```

# Sintassi

```
hsla(<hue>, <saturation>%, <lightness>%, <alpha>);
```

| Valore       | Descrizione   |
|--------------|---|
| <hue>        | specificato in gradi attorno alla ruota dei colori (senza unità), dove 0 ° è rosso, 60 ° è giallo, 120 ° è verde, 180 ° è ciano, 240 ° è blu, 300 ° è magenta e 360 ° è rosso |
| <saturation> | percentuale in cui lo 0% è completamente desaturato (scala di grigi) e il 100% è completamente saturo (colori vividi)   |
| <lightness>  | percentuale dove 0% è completamente nero e 100% è completamente bianco  |
| <alpha>      | un numero da 0 a 1 dove 0 è completamente trasparente e 1 è completamente opaco   |

Leggi Colori online: <https://riptutorial.com/it/css/topic/644/colori>

---

# Capitolo 12: Commenti

## Sintassi

- `/* Commento */`

## Osservazioni

- I commenti nei CSS iniziano sempre con `/*` e terminano con `*/`
- I commenti non possono essere nidificati

## Examples

### Linea singola

```
/* This is a CSS comment */
div {
  color: red; /* This is a CSS comment */
}
```

### Linea multipla

```
/*
  This
  is
  a
  CSS
  comment
*/
div {
  color: red;
}
```

Leggi Commenti online: <https://riptutorial.com/it/css/topic/1625/commenti>

---

# Capitolo 13: Confine

## Sintassi

- **confine**
- border: border-width border-style border-color | iniziale | ereditare;
- border-top: border-width border-style border-color | iniziale | ereditare;
- border-bottom: border-width border-style border-color | iniziale | ereditare;
- border-left: border-width border-style border-color | iniziale | ereditare;
- border-right: border-width border-style border-color | iniziale | ereditare;
- **stile del bordo**
- border-style: 1-4 nessuno | nascosto | punteggiato | tratteggiata | solido | doppio | groove | cresta | inserto | inizio | iniziale | ereditare;
- **border-radius**
- border-radius: 1-4 lunghezza | % / 1-4 lunghezza | % | iniziale | ereditare;
- border-top-left-radius: length | % [lunghezza | %] | iniziale | ereditare;
- border-top-right-radius: length | % [lunghezza | %] | iniziale | ereditare;
- border-bottom-left-radius: length | % [lunghezza | %] | iniziale | ereditare;
- border-bottom-right-radius: length | % [lunghezza | %] | iniziale | ereditare;
- **border-immagine**
- border-image: border-image-source border-image-slice [border-image-width [border-image-outset]] border-image-repeat
- border-image-source: nessuno | Immagine;
- border-image-slice: numero 1-4 | percentuale [riempimento]
- border-image-repeat: 1-2 stretch | ripetere | rotondo | spazio
- **border-collapse**
- border-collapse: separato | collasso | iniziale | ereditare

## Osservazioni

## Proprietà correlate:

- confine
- border-bottom
- border-bottom-color
- border-bottom-left-radius
- border-bottom-right-radius
- -Bottom border-style
- border-bottom-width
- colore del bordo
- border-immagine
- border-image-fin
- border-image-repeat
- border-image-slice
- border-image-source
- border-image-width
- border-left
- border-left-color
- border-left-style
- border-left-width
- border-radius
- border-right
- border-right-color
- border-right-style
- border-right-width
- stile del bordo
- border-top
- border-top-color

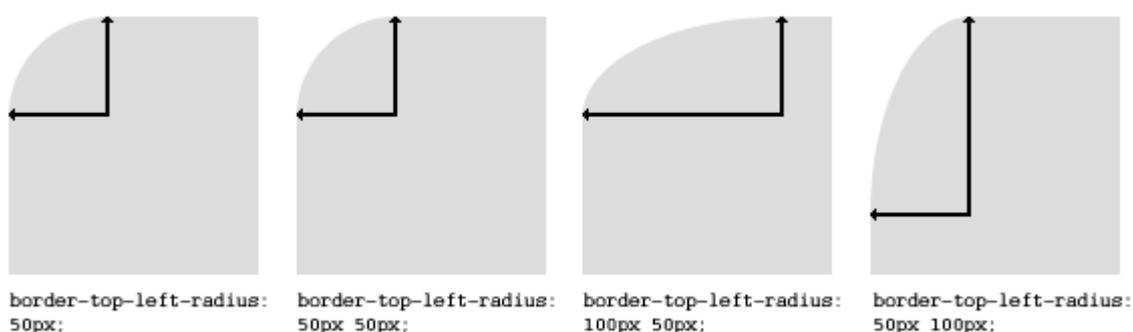
- border-top-left-radius
- border-top-right-radius
- -Top border-style
- border-top-width
- larghezza del bordo

## Examples

### border-radius

La proprietà border-radius consente di modificare la forma del modello di box base.

Ogni angolo di un elemento può avere fino a due valori, per il raggio verticale e orizzontale di quell'angolo (per un massimo di 8 valori).



Il primo insieme di valori definisce il raggio orizzontale. Il secondo set di valori opzionale, preceduto da un '/', definisce il raggio verticale. Se viene fornito un solo set di valori, viene utilizzato sia per il raggio verticale che orizzontale.

```
border-radius: 10px 5% / 20px 25em 30px 35em;
```

Il 10px è il raggio orizzontale in alto a sinistra e in basso a destra. E il 5% è il raggio orizzontale in alto a destra e in basso a sinistra. Gli altri quattro valori dopo "/" sono i raggi verticali per top-left, top-right, bottom-right e bottom-left.

Come per molte proprietà CSS, le abbreviazioni possono essere utilizzate per qualsiasi o tutti i valori possibili. È quindi possibile specificare qualsiasi valore da uno a otto valori. La seguente stenografia consente di impostare il raggio orizzontale e verticale di ogni angolo sullo stesso valore:

HTML:

```
<div class='box'></div>
```

## CSS:

```
.box {
  width: 250px;
  height: 250px;
  background-color: black;
  border-radius: 10px;
}
```

Border-radius è più comunemente usato per convertire elementi di riquadri in cerchi. Impostando il raggio del bordo a metà della lunghezza di un elemento quadrato, viene creato un elemento circolare:

```
.circle {
  width: 200px;
  height: 200px;
  border-radius: 100px;
}
```

Poiché il limite del raggio di confine accetta percentuali, è comune utilizzare il 50% per evitare di calcolare manualmente il valore del raggio del bordo:

```
.circle {
  width: 150px;
  height: 150px;
  border-radius: 50%;
}
```

Se le proprietà width e height non sono uguali, la forma risultante sarà un ovale piuttosto che un cerchio.

Esempio di raggio di confine specifico del browser:

```
-webkit-border-top-right-radius: 4px;
-webkit-border-bottom-right-radius: 4px;
-webkit-border-bottom-left-radius: 0;
-webkit-border-top-left-radius: 0;
-moz-border-radius-topright: 4px;
-moz-border-radius-bottomright: 4px;
-moz-border-radius-bottomleft: 0;
-moz-border-radius-topleft: 0;
border-top-right-radius: 4px;
border-bottom-right-radius: 4px;
border-bottom-left-radius: 0;
border-top-left-radius: 0;
```

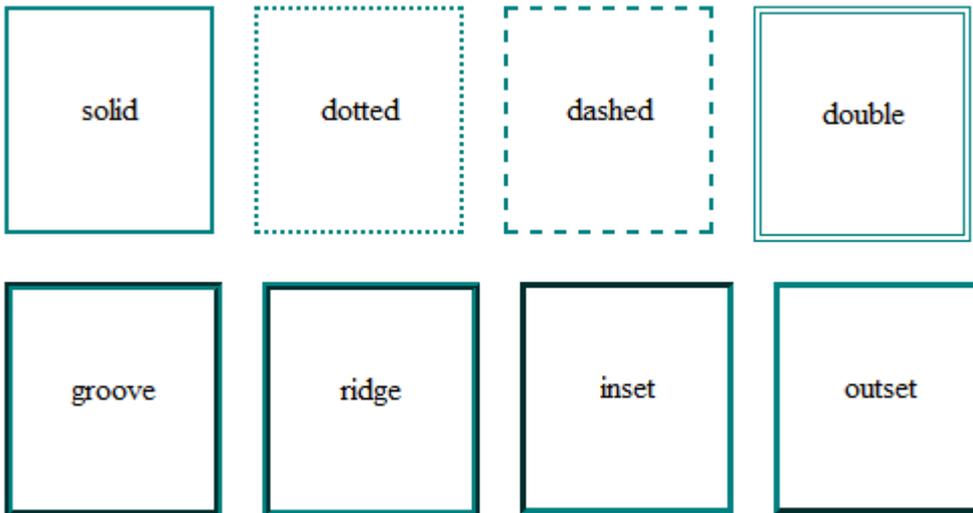
## stile del bordo

La proprietà `border-style` imposta lo stile del bordo di un elemento. Questa proprietà può avere da uno a quattro valori (per ogni lato dell'elemento un valore).

Esempi:

```
border-style: dotted;
```

```
border-style: dotted solid double dashed;
```



`border-style` può anche avere i valori `none` e `hidden` . Hanno lo stesso effetto, ad eccezione delle opere `hidden` per la risoluzione dei conflitti di confine per gli elementi `<table>` . In una `<table>` con più bordi, `none` ha la priorità più bassa (nel senso di un conflitto, il bordo dovrebbe mostrare), e `hidden` ha la priorità più alta (nel senso di un conflitto, il bordo non mostrerebbe).

## confine (stenografia)

Nella maggior parte dei casi, si desidera definire diverse proprietà del bordo ( `border-width border-style border-color` ) per tutti i lati di un elemento.

Invece di scrivere:

```
border-width: 1px;  
border-style: solid;  
border-color: #000;
```

Puoi semplicemente scrivere:

```
border: 1px solid #000;
```

Queste stenografie sono disponibili anche per ogni lato di un elemento: `border-top` , `border-left` , `border-right` e `border-bottom` . Quindi puoi fare:

```
border-top: 2px double #aaaaaa;
```

## border-immagine

Con la proprietà `border-image` hai la possibilità di impostare un'immagine da usare al posto dei normali stili di bordo.

Un'immagine di `border-image` consiste essenzialmente di a

- `border-image-source` : il percorso dell'immagine da utilizzare
- `border-image-slice` : specifica l'offset utilizzato per dividere l'immagine in **nove regioni** (quattro **angoli** , quattro **bordi** e una parte **centrale** )
- `border-image-repeat` : specifica come vengono ridimensionate le immagini per i lati e il centro dell'immagine del bordo

Considera il seguente esempio dove `border.png` è un'immagine di 90x90 pixel:

```
border-image: url("border.png") 30 stretch;
```

L'immagine sarà divisa in nove regioni con 30x30 pixel. I bordi saranno usati come angoli del bordo mentre il lato sarà usato in mezzo. Se l'elemento è più alto / più largo di 30px questa parte dell'immagine sarà **allungata** . La parte centrale dell'immagine è di default trasparente.

## `border-[left | right | top | bottom]`

La proprietà `border-[left|right|top|bottom]` viene utilizzata per aggiungere un bordo a un lato specifico di un elemento.

Ad esempio, se si desidera aggiungere un bordo sul lato sinistro di un elemento, è possibile:

```
#element {
  border-left: 1px solid black;
}
```

## `border-collapse`

La proprietà `border-collapse` applica solo alla `table s` (e agli elementi visualizzati come `display: table` o `display: table inline-table` ) e stabilisce se i bordi della tabella vengono compressi in un singolo bordo o separati come in HTML standard.

```
table {
  border-collapse: separate; /* default */
  border-spacing: 2px; /* Only works if border-collapse is separate */
}
```

Vedi anche [Tabelle](#) - voce della documentazione di [compressione del bordo](#)

## Bordi multipli

Utilizzo del contorno:

```
.div1{
  border: 3px solid black;
  outline: 6px solid blue;
  width: 100px;
  height: 100px;
  margin: 20px;
```

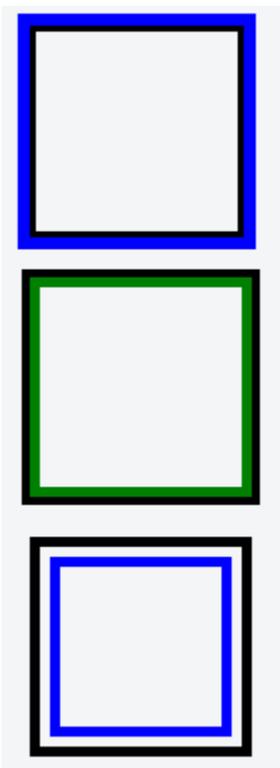
```
}
```

## Usando box-shadow:

```
.div2{  
  border: 5px solid green;  
  box-shadow: 0px 0px 0px 4px #000;  
  width: 100px;  
  height: 100px;  
  margin: 20px;  
}
```

## Usando uno pseudo elemento:

```
.div3 {  
  position: relative;  
  border: 5px solid #000;  
  width: 100px;  
  height: 100px;  
  margin: 20px;  
}  
.div3:before {  
  content: " ";  
  position: absolute;  
  border: 5px solid blue;  
  z-index: -1;  
  top: 5px;  
  left: 5px;  
  right: 5px;  
  bottom: 5px;  
}
```



<http://jsfiddle.net/MadalinaTn/bvqpcohm/2/>

## Creare un bordo multicolore usando l'immagine del bordo

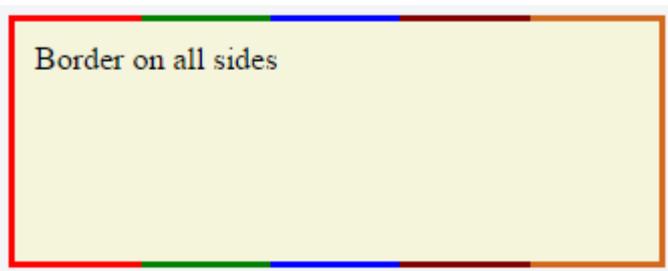
### CSS

```
.bordered {  
  border-image: linear-gradient(to right, red 20%, green 20%, green 40%, blue 40%, blue 60%,  
  maroon 60%, maroon 80%, chocolate 80%); /* gradient with required colors */  
  border-image-slice: 1;  
}
```

### HTML

```
<div class='bordered'>Border on all sides</div>
```

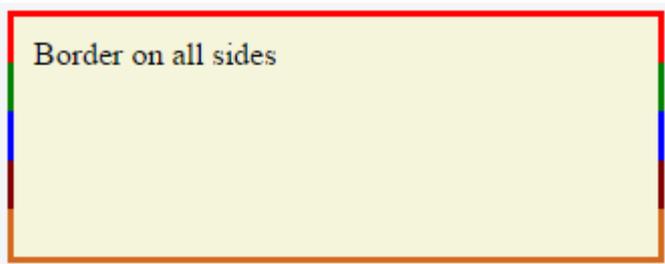
L'esempio precedente produrrebbe un bordo che comprende 5 colori diversi. I colori sono definiti attraverso un `linear-gradient` (puoi trovare maggiori informazioni sui gradienti nei [documenti](#)). Puoi trovare maggiori informazioni sulla proprietà `border-image-slice` nell'esempio [border-image](#) nella stessa pagina.



( Nota: ulteriori proprietà sono state aggiunte all'elemento per scopi di presentazione ) .

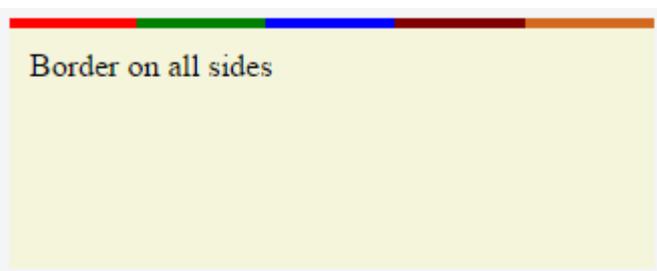
Avresti notato che il bordo sinistro ha un solo colore (il colore iniziale del gradiente) mentre il bordo destro ha anche un solo colore (il colore finale del gradiente). Questo è dovuto al modo in cui funziona la proprietà dell'immagine del bordo. È come se il gradiente fosse applicato all'intera casella e quindi i colori venissero mascherati dalle aree di riempimento e di contenuto, facendolo apparire come se solo il bordo avesse il gradiente.

Quale / i bordo / i hanno un singolo colore dipende dalla definizione del gradiente. Se il gradiente è un gradiente da `to right`, il bordo sinistro sarà il colore iniziale del gradiente e il bordo destro sarà il colore finale. Se era un gradiente in `to bottom`, il bordo superiore sarebbe il colore iniziale del gradiente e il bordo inferiore sarebbe il colore finale. Sotto è l'uscita di un `to bottom 5` sfumatura di colore.



Se il bordo è richiesto solo su lati specifici dell'elemento, allora la proprietà `border-width` può essere utilizzata come qualsiasi altro bordo normale. Ad esempio, l'aggiunta del codice seguente produrrebbe un bordo solo nella parte superiore dell'elemento.

```
border-width: 5px 0px 0px 0px;
```



Si noti che, qualsiasi elemento che ha proprietà `border-image` **non rispetterà il** `border-radius` (cioè il bordo non curva). Questo si basa sulla seguente affermazione nelle specifiche:

Gli sfondi di una scatola, ma non la sua immagine di bordo, vengono ritagliati sulla curva appropriata (come determinato da 'background-clip').

Leggi Confine online: <https://riptutorial.com/it/css/topic/2160/confine>

# Capitolo 14: contatori

## Sintassi

- contatore: [`<counter-name> <numero intero>?`] + | nessuna
- counter-reset: [`<counter-name> <numero intero>?`] + | nessuna
- counter-increment: [`<counter-name> <integer>?`] + | nessuna
- counter (`<counter-name> [, <counter-style>]?`)
- contatori (`<counter-name>, <connector-string> [, <counter-style>]?`)

## Parametri

| Parametro          | Dettagli   |
|--------------------|--|
| contro-nome        | Questo è il nome del contatore che deve essere creato o incrementato o stampato. Può essere un nome personalizzato come lo desiderano gli sviluppatori.  |
| numero intero      | Questo intero è un valore facoltativo che, se fornito accanto al nome del contatore, rappresenterà il valore iniziale del contatore (in <code>counter-set</code> , proprietà di <code>counter-reset</code> ) o il valore con cui il contatore deve essere incrementato (in <code>counter-increment</code> ). |
| nessuna            | Questo è il valore iniziale per tutte e 3 le proprietà <code>counter-*</code> . Quando questo valore viene utilizzato per il <code>counter-increment</code> , il valore di nessuno dei contatori è interessato. Quando viene utilizzato per gli altri due, non viene creato alcun contatore.                 |
| contro-style       | Specifica lo stile in cui deve essere visualizzato il valore del contatore. Supporta tutti i valori supportati dalla proprietà <code>list-style-type</code> . Se <code>none</code> si utilizza <code>none</code> il valore del contatore non viene stampato affatto.   |
| Connettore-stringa | Rappresenta la stringa che deve essere posizionata tra i valori di due diversi livelli di contatore (come "." In "2.1.1").   |

## Osservazioni

I contatori non sono un nuovo argomento nei CSS. Era una parte delle Specifiche CSS Livello 2 (Revisione 1 per la precisione) e quindi ha un supporto browser molto alto.

Tutti i browser tranne IE6 e IE7 supportano i contatori CSS.

# Examples

## Applicando lo stile dei numeri romani all'uscita contatore

### CSS

```
body {
  counter-reset: item-counter;
}

.item {
  counter-increment: item-counter;
}

.item:before {
  content: counter(item-counter, upper-roman) ". "; /* by specifying the upper-roman as style
the output would be in roman numbers */
}
```

### HTML

```
<div class='item'>Item No: 1</div>
<div class='item'>Item No: 2</div>
<div class='item'>Item No: 3</div>
```

Nell'esempio sopra, l'uscita del contatore verrà visualizzata come I, II, III (numeri romani) invece del solito 1, 2, 3 poiché lo sviluppatore ha esplicitamente specificato lo stile del contatore.

## Numera ogni oggetto usando il contatore CSS

### CSS

```
body {
  counter-reset: item-counter; /* create the counter */
}

.item {
  counter-increment: item-counter; /* increment the counter every time an element with class
"item" is encountered */
}

.item-header:before {
  content: counter(item-counter) ". "; /* print the value of the counter before the header and
append a "." to it */
}

/* just for demo */

.item {
  border: 1px solid;
```

```
height: 100px;
margin-bottom: 10px;
}
.item-header {
border-bottom: 1px solid;
height: 40px;
line-height: 40px;
padding: 5px;
}
.item-content {
padding: 8px;
}
```

---

## HTML

```
<div class='item'>
  <div class='item-header'>Item 1 Header</div>
  <div class='item-content'>Lorem Ipsum Dolor Sit Amet...</div>
</div>
<div class='item'>
  <div class='item-header'>Item 2 Header</div>
  <div class='item-content'>Lorem Ipsum Dolor Sit Amet...</div>
</div>
<div class='item'>
  <div class='item-header'>Item 3 Header</div>
  <div class='item-content'>Lorem Ipsum Dolor Sit Amet...</div>
</div>
```

L'esempio numererà sopra ogni "elemento" nella pagina e aggiunge il numero del articolo prima sua intestazione (con `content` di proprietà di `.item-header` elemento di `:before` pseudo). Una demo live di questo codice è disponibile [qui](#).

## Implementazione della numerazione multilivello utilizzando i contatori CSS

---

## CSS

```
ul {
list-style: none;
counter-reset: list-item-number; /* self nesting counter as name is same for all levels */
}
li {
counter-increment: list-item-number;
}
li:before {
content: counters(list-item-number, ".") " "; /* usage of counters() function means value of
counters at all higher levels are combined before printing */
}
```

---

## HTML

```

<ul>
  <li>Level 1
    <ul>
      <li>Level 1.1
        <ul>
          <li>Level 1.1.1</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>Level 2
    <ul>
      <li>Level 2.1
        <ul>
          <li>Level 2.1.1</li>
          <li>Level 2.1.2</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>Level 3</li>
</ul>

```

Quanto sopra è un esempio di numerazione multi-livello che utilizza i contatori CSS. Fa uso del concetto di contatori **auto-nidificante**. L'annidamento automatico è un concetto in cui se un elemento ha già un contatore con il nome dato ma deve crearne un altro, lo crea come figlio del contatore esistente. Qui, il secondo livello `ul` eredita già il contatore `list-item-number` dal suo genitore, ma deve quindi creare il proprio numero `list-item-number` (per i suoi figli `li`) e quindi crea il numero `list-item-number[1]` (contatore per secondo livello) e lo annida sotto il numero `list-item-number[0]` (contatore per il primo livello). In tal modo raggiunge la numerazione multilivello.

L'output viene stampato utilizzando la funzione `counters()` anziché la funzione `counter()` poiché la funzione `counters()` è progettata per prefissare il valore di tutti i contatori di livello superiore (parent) durante la stampa dell'output.

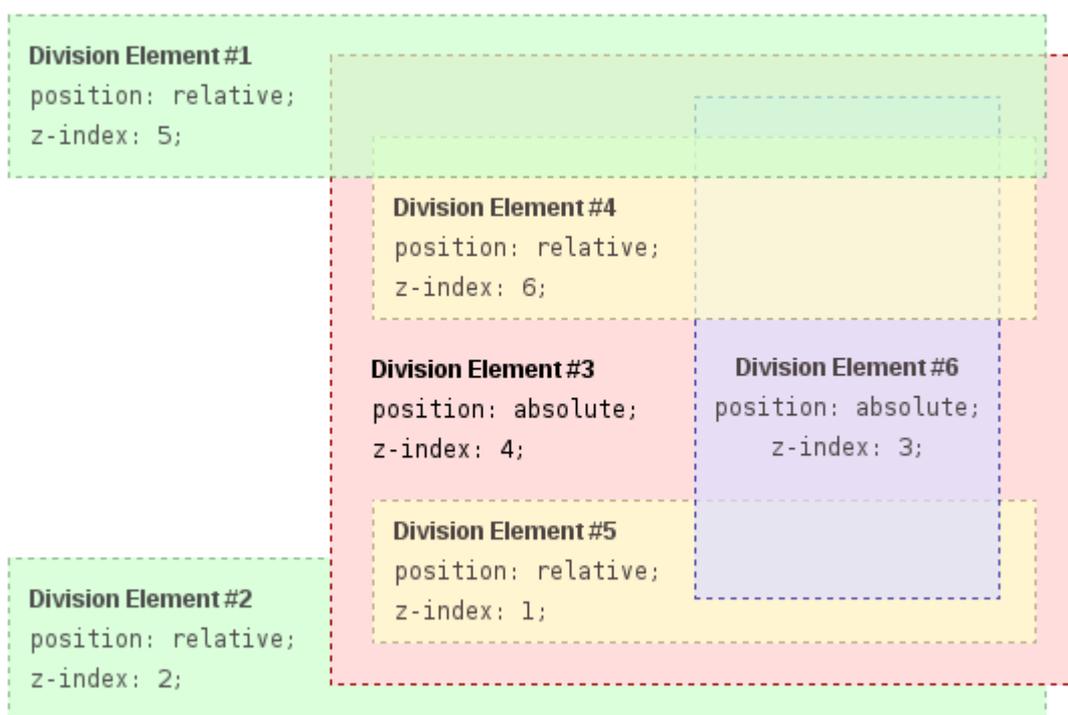
Leggi contatori online: <https://riptutorial.com/it/css/topic/2575/contatori>

# Capitolo 15: Contesto impilabile

## Examples

### Contesto impilabile

In questo esempio, ogni elemento posizionato crea il proprio contesto di impilamento, a causa del loro posizionamento e dei valori di z-index. La gerarchia dei contesti di impilamento è organizzata come segue:



- Radice
  - DIV # 1
  - DIV # 2
  - DIV # 3
  - DIV # 4
  - DIV # 5
  - DIV # 6

È importante notare che DIV # 4, DIV # 5 e DIV # 6 sono figli di DIV # 3, quindi l'impilamento di questi elementi è completamente risolto nel DIV # 3. Una volta completato il raggruppamento e il rendering in DIV # 3, l'intero elemento DIV # 3 viene passato per l'impilamento nell'elemento radice rispetto al DIV di pari livello.

### HTML:

```
<div id="div1">  
  <h1>Division Element #1</h1>
```

```

<code>position: relative;<br/>
z-index: 5;</code>
</div>
<div id="div2">
  <h1>Division Element #2</h1>
  <code>position: relative;<br/>
  z-index: 2;</code>
</div>
<div id="div3">
  <div id="div4">
    <h1>Division Element #4</h1>
    <code>position: relative;<br/>
    z-index: 6;</code>
  </div>
  <h1>Division Element #3</h1>
  <code>position: absolute;<br/>
  z-index: 4;</code>
  <div id="div5">
    <h1>Division Element #5</h1>
    <code>position: relative;<br/>
    z-index: 1;</code>
  </div>
  <div id="div6">
    <h1>Division Element #6</h1>
    <code>position: absolute;<br/>
    z-index: 3;</code>
  </div>
</div>

```

## CSS:

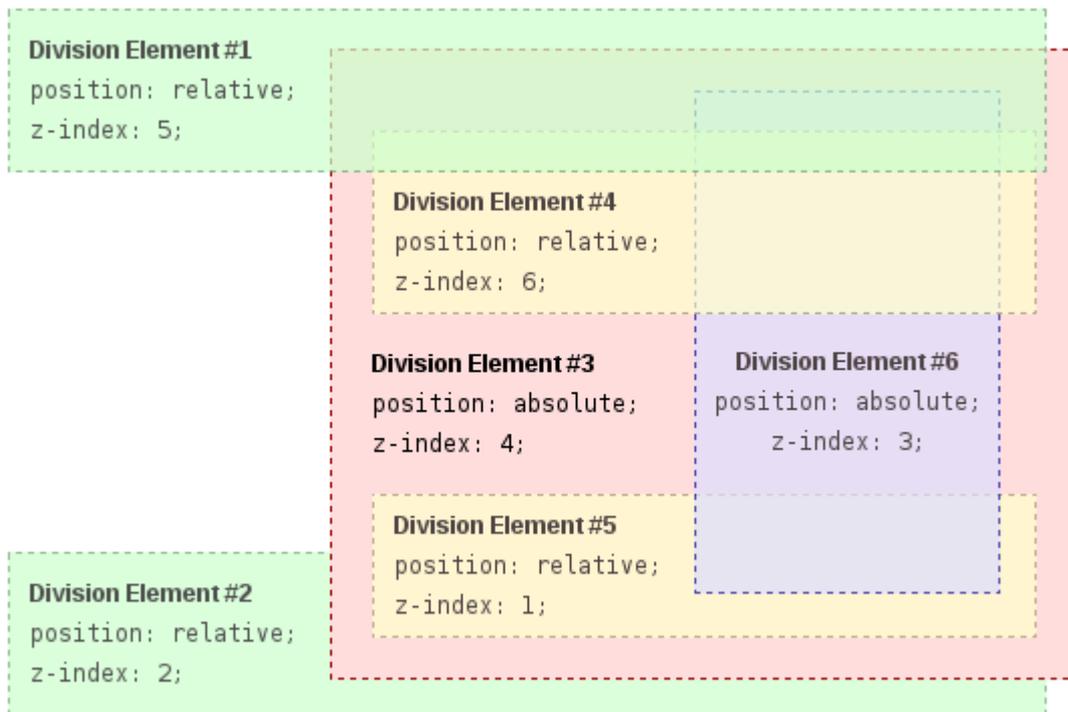
```

* {
  margin: 0;
}
html {
  padding: 20px;
  font: 12px/20px Arial, sans-serif;
}
div {
  opacity: 0.7;
  position: relative;
}
h1 {
  font: inherit;
  font-weight: bold;
}
#div1,
#div2 {
  border: 1px dashed #696;
  padding: 10px;
  background-color: #cfc;
}
#div1 {
  z-index: 5;
  margin-bottom: 190px;
}
#div2 {
  z-index: 2;
}
#div3 {

```

```
z-index: 4;
opacity: 1;
position: absolute;
top: 40px;
left: 180px;
width: 330px;
border: 1px dashed #900;
background-color: #fdd;
padding: 40px 20px 20px;
}
#div4,
#div5 {
border: 1px dashed #996;
background-color: #ffc;
}
#div4 {
z-index: 6;
margin-bottom: 15px;
padding: 25px 10px 5px;
}
#div5 {
z-index: 1;
margin-top: 15px;
padding: 5px 10px;
}
#div6 {
z-index: 3;
position: absolute;
top: 20px;
left: 180px;
width: 150px;
height: 125px;
border: 1px dashed #009;
padding-top: 125px;
background-color: #ddf;
text-align: center;
}
```

## Risultato:



Fonte: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Positioning/Understanding\\_z\\_index/The\\_stacking\\_context](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context) .

Leggi Contesto impilabile online: <https://riptutorial.com/it/css/topic/5037/contesto-impilabile>

# Capitolo 16: Controllo del layout

## Sintassi

- display: nessuno | inline | blocco | lista-elemento | elemento della lista inline | blocco in linea | tavolo in linea | tabella | cella-tabella | tabella-colonna | gruppo-colonna-tabella | tabella-footer-group | tabella-intestazione-gruppo | table-row | gruppo di righe della tabella | flex | inline-flex | griglia | griglia in linea | run-in | rubino | rubino-base | rubino-testo | contenitore-rubino-base | contenitore-testo-rubino | contenuto;

## Parametri

| Valore       | Effetto  |
|--------------|--|
| none         | Nascondi l'elemento e impediscilo di occupare spazio.  |
| block        | Elemento di blocco, occupa il 100% della larghezza disponibile, rompi dopo elemento.   |
| inline       | Elemento in linea, non occupa larghezza, nessuna interruzione dopo elemento.   |
| inline-block | Assunzione di proprietà speciali da elementi sia in linea che a blocchi, nessuna interruzione, ma può avere larghezza.                                 |
| inline-flex  | Visualizza un elemento come un contenitore flessibile a livello di linea.  |
| inline-table | L'elemento viene visualizzato come una tabella a livello in linea.   |
| grid         | Si comporta come un elemento di blocco e ne espone il contenuto in base al modello di griglia.   |
| flex         | Si comporta come un elemento di blocco e presenta il suo contenuto secondo il modello di flexbox.  |
| inherit      | Eredita il valore dall'elemento genitore.  |
| initial      | Reimposta il valore sul valore predefinito preso dai comportamenti descritti nelle specifiche HTML o dal foglio di stile predefinito browser / utente. |
| table        | Si comporta come l'elemento <code>table</code> HTML.   |
| table-cell   | Lascia che l'elemento si comporti come un elemento <code>&lt;td&gt;</code>   |
| table-column | Lascia che l'elemento si comporti come un elemento <code>&lt;col&gt;</code>  |

| Valore                 | Effetto  |
|------------------------|--|
| <code>table-row</code> | Lascia che l'elemento si comporti come un elemento <code>&lt;tr&gt;</code>   |
| <code>list-item</code> | Lascia che l'elemento si comporti come un elemento <code>&lt;li&gt;</code> . |

## Examples

### La proprietà di visualizzazione

La proprietà CSS di `display` è fondamentale per controllare il layout e il flusso di un documento HTML. La maggior parte degli elementi ha un valore di `display` predefinito di entrambi i `block` o `inline` (sebbene alcuni elementi abbiano altri valori predefiniti).

## In linea

Un elemento in `inline` occupa solo la larghezza necessaria. Si impila orizzontalmente con altri elementi dello stesso tipo e non può contenere altri elementi non in linea.

```
<span>This is some <b>bolded</b> text!</span>
```

This is some **bolded** text!

Come dimostrato sopra, due elementi `inline` , `<span>` e `<b>` , sono in linea (da cui il nome) e non interrompono il flusso del testo.

## Bloccare

Un elemento di `block` occupa la larghezza massima disponibile del suo elemento padre. Inizia con una nuova riga e, a differenza degli elementi in `inline` , non limita il tipo di elementi che può contenere.

```
<div>Hello world!</div><div>This is an example!</div>
```

Hello world!  
This is an example!

L'elemento `div` è a livello di blocco per impostazione predefinita e, come mostrato sopra, i due elementi di `block` sono impilati verticalmente e, a differenza degli elementi in `inline` , il flusso del testo si interrompe.

## Blocco in linea

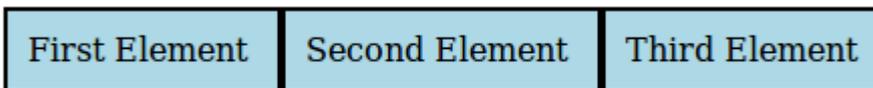
Il valore `inline-block` ci offre il meglio di entrambi i mondi: unisce l'elemento al flusso del testo e ci consente di utilizzare `padding`, `margin`, `height` e proprietà simili che non hanno alcun effetto visibile sugli elementi in `inline`.

Gli elementi con questo valore di visualizzazione si comportano come se fossero testo normale e, di conseguenza, sono influenzati dalle regole che controllano il flusso del testo, ad esempio `text-align`. Di default sono anche ridotte alla dimensione più piccola possibile per ospitare il loro contenuto.

```
<!--Inline: unordered list-->
<style>
li {
  display : inline;
  background : lightblue;
  padding:10px;

  border-width:2px;
  border-color:black;
  border-style:solid;
}
</style>

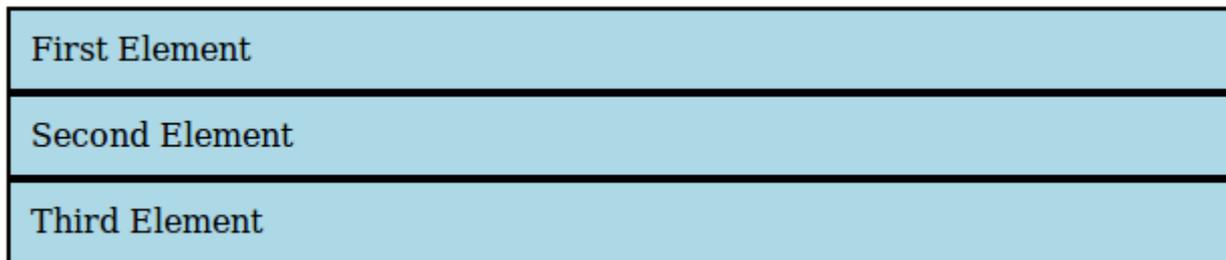
<ul>
<li>First Element </li>
<li>Second Element </li>
<li>Third Element </li>
</ul>
```



```
<!--block: unordered list-->
<style>
li {
  display : block;
  background : lightblue;
  padding:10px;

  border-width:2px;
  border-color:black;
  border-style:solid;
}
</style>

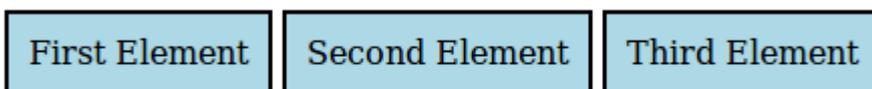
<ul>
<li>First Element </li>
<li>Second Element </li>
<li>Third Element </li>
</ul>
```



```
<!--Inline-block: unordered list-->
<style>
li {
  display : inline-block;
  background : lightblue;
  padding:10px;

  border-width:2px;
  border-color:black;
  border-style:solid;
  }
</style>

<ul>
<li>First Element </li>
<li>Second Element </li>
<li>Third Element </li>
</ul>
```



---

## nessuna

Un elemento a cui non viene assegnato il valore none alla proprietà display non verrà visualizzato affatto.

Ad esempio creiamo un elemento div che ha un id di `myDiv` :

```
<div id="myDiv"></div>
```

Questo può ora essere contrassegnato come non visualizzato dalla seguente regola CSS:

```
#myDiv {
  display: none;
}
```

Quando un elemento è stato impostato per essere `display:none`; il browser ignora ogni altra proprietà di layout per quell'elemento specifico (sia `position` che `float` ). Nessuna casella sarà resa per quell'elemento e la sua esistenza in html non influenzerà la posizione dei seguenti elementi.

Si noti che questo è diverso dall'impostazione della proprietà di `visibility` su `hidden`.

Impostazione della `visibility: hidden`; perché un elemento non visualizza l'elemento sulla pagina ma l'elemento continua a occupare lo spazio nel processo di rendering come se fosse visibile. Ciò influenzerà quindi il modo in cui i seguenti elementi vengono visualizzati sulla pagina.

Il valore `none` per la proprietà `display` è comunemente usato insieme a JavaScript per mostrare o nascondere elementi a volontà, eliminando la necessità di eliminarli e ricrearli effettivamente.

## Per ottenere la vecchia struttura della tabella utilizzando `div`

Questa è la normale struttura della tabella HTML

```
<style>
  table {
    width: 100%;
  }
</style>

<table>
  <tr>
    <td>
      I'm a table
    </td>
  </tr>
</table>
```

Puoi fare la stessa implementazione in questo modo

```
<style>
  .table-div {
    display: table;
  }
  .table-row-div {
    display: table-row;
  }
  .table-cell-div {
    display: table-cell;
  }
</style>

<div class="table-div">
  <div class="table-row-div">
    <div class="table-cell-div">
      I behave like a table now
    </div>
  </div>
</div>
```

Leggi Controllo del layout online: <https://riptutorial.com/it/css/topic/1473/controllo-del-layout>

---

# Capitolo 17: CSS Image Sprites

## Sintassi

- **// Utilizzo di background-position**  
background: url ("sprite-image.png");  
posizione di sfondo: -20px 50px;
- **// Stenografia della proprietà sullo sfondo**  
background: url ("sprite-image.png") -20px 50px;

## Osservazioni

Per alcuni casi di utilizzo, gli sprites stanno lentamente cadendo in disgrazia, venendo sostituiti da icone webfonts o [immagini SVG](#) .

## Examples

### Un'implementazione di base

#### Cos'è uno sprite di immagini?

Un immagine sprite è una singola risorsa situata all'interno di un foglio di immagine sprite. Un foglio immagine sprite è un file immagine che contiene più di un asset che può essere estratto da esso.

Per esempio:



L'immagine sopra è un foglio di immagine sprite, e ognuna di quelle stelle è uno sprite all'interno del foglio sprite. Questi fogli sprite sono utili perché migliorano le prestazioni riducendo il numero di richieste HTTP che un browser potrebbe dover effettuare.

Quindi come si implementa uno? Ecco alcuni esempi di codice.

## HTML

```
<div class="icon icon1"></div>
```

```
<div class="icon icon2"></div>
<div class="icon icon3"></div>
```

## CSS

```
.icon {
  background: url("icons-sprite.png");
  display: inline-block;
  height: 20px;
  width: 20px;
}
.icon1 {
  background-position: 0px 0px;
}
.icon2 {
  background-position: -20px 0px;
}
.icon3 {
  background-position: -40px 0px;
}
```

Usando l'impostazione della larghezza e dell'altezza dello sprite e usando la proprietà `background-position` in CSS (con un valore `key`) è possibile estrarre facilmente gli sprite da un foglio sprite usando CSS.

Leggi CSS Image Sprites online: <https://riptutorial.com/it/css/topic/3690/css-image-sprites>

# Capitolo 18: CSS Object Model (CSSOM)

## Osservazioni

Il CSS Object Model (CSSOM) è una specifica a parte.

La bozza attuale può essere trovata qui: <https://www.w3.org/TR/cssom-1/>

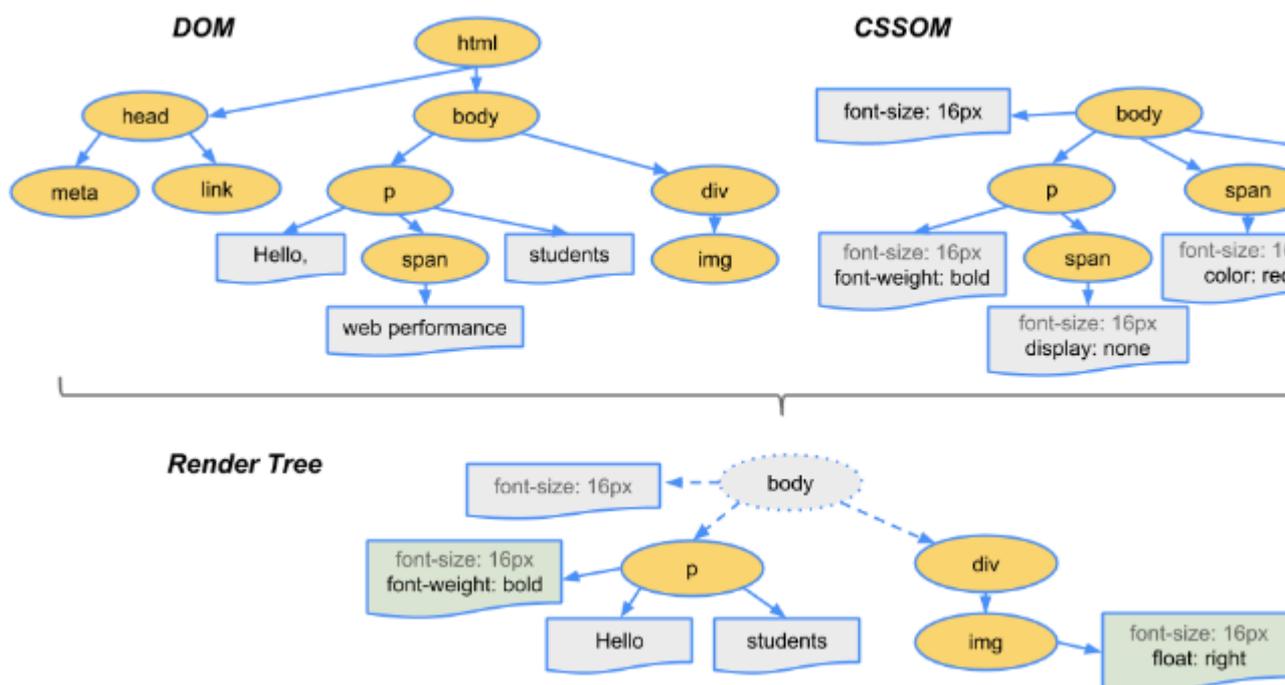
## Examples

### introduzione

Il browser identifica i token dal foglio di stile e li ricopre in nodi che sono collegati in una struttura ad albero. L'intera mappa di tutti i nodi con gli stili associati di una pagina sarebbe il modello a oggetti CSS.

Per visualizzare la pagina Web, un browser Web esegue i seguenti passaggi.

1. Il browser Web esamina il tuo HTML e crea il DOM (Document Object Model).
2. Il browser Web esamina il tuo CSS e crea il CSSOM (modello a oggetti CSS).
3. Il browser Web combina DOM e CSSOM per creare un albero di rendering. Il browser web mostra la tua pagina web.



## Aggiunta di una regola per l'immagine di sfondo tramite CSSOM

Per aggiungere una regola dell'immagine di sfondo tramite CSSOM, prima ottenere un riferimento

alle regole del primo foglio di stile:

```
var stylesheet = document.styleSheets[0].cssRules;
```

Quindi, ottieni un riferimento alla fine del foglio di stile:

```
var end = stylesheet.length - 1;
```

Infine, inserisci una regola background-image per l'elemento body alla fine del foglio di stile:

```
stylesheet.insertRule("body { background-image:  
url('http://cdn.sstatic.net/Sites/stackoverflow/img/favicon.ico'); }", end);
```

Leggi CSS Object Model (CSSOM) online: <https://riptutorial.com/it/css/topic/4961/css-object-model-cssom->

# Capitolo 19: Corsore Styling

## Sintassi

- cursore: auto | default | nessuno | menu contestuale | aiuto | puntatore | progresso | aspetta | cella | mirino | testo | testo verticale | alias | copia | mossa | no-drop | non consentito | e-ridimensiona | n-ridimensiona | ridimensionare | nw-resize | s-ridimensiona | se-ridimensiona | sw resize | w-ridimensiona | ew-resize | ns-resize | nesw-ridimensiona | nwse-resize | col-ridimensiona | ridimensionamento riga | tutto-scorrere | zoom in | | zoom indietro | afferrare | afferrare;

## Examples

### Cambiare il tipo di cursore

```
cursor: value;
```

|   |               |   |           |   |               |
|---|---------------|---|-----------|---|---------------|
|   | default       |   | n-resize  |   | not-allowed   |
|  | crosshair     |  | ne-resize |  | no-drop       |
|  | hand          |  | e-resize  |  | vertical-text |
|  | pointer       |  | se-resize |  | all-scroll    |
|  | Cross browser |  | s-resize  |  | col-resize    |
|  | move          |  | sw-resize |  | row-resize    |
|  | text          |  | w-resize  |   |               |
|  | wait          |  | nw-resize |   |               |
|  | help          |  | progress  |   |               |

### Esempi:

| Valore    | Descrizione   |
|-----------|---|
| nessuna   | Nessun cursore è reso per l'elemento                |
| auto      | Predefinito. Il browser imposta un cursore          |
| Aiuto     | Il cursore indica che l'aiuto è disponibile         |
| aspettare | Il cursore indica che il programma è occupato       |
| mossa     | Il cursore indica che qualcosa deve essere spostato |
| pointer   | Il cursore è un puntatore e indica un collegamento  |

## pointer-events

La proprietà `pointer-events` consente di controllare il modo in cui gli elementi HTML rispondono agli eventi mouse / touch.

```
.disabled {  
  pointer-events: none;  
}
```

In questo esempio,

'nessuno' impedisce tutte le opzioni di clic, stato e cursore sull'elemento HTML specificato [\[\[1\]\]](#)

Altri valori validi per gli elementi HTML sono:

- `auto`;
- `ereditare`.

1. <https://css-tricks.com/almanac/properties/p/pointer-events/>

Altre risorse:

- <https://developer.mozilla.org/en-US/docs/Web/CSS/pointer-events>
- <https://davidwalsh.name/pointer-events>

## caret-color

La proprietà CSS `color caret` specifica il colore del punto di inserimento, l'indicatore visibile del punto di inserimento in un elemento in cui il testo e altri contenuti vengono inseriti dalla digitazione o modifica dell'utente.

HTML

```
<input id="example" />
```

CSS

```
#example {  
  caret-color: red;  
}
```

risorse:

- <https://developer.mozilla.org/en-US/docs/Web/CSS/caret-color>

Leggi **Cursore Styling** online: <https://riptutorial.com/it/css/topic/1742/cursore-styling>

# Capitolo 20: Elenca stili

## Sintassi

- list-style: list-style-type | lista-stile-posizione | list-style-image | iniziale | ereditare;

## Parametri

| Valore              | Descrizione  |
|---------------------|--|
| list-style-type     | il tipo di marcatore elemento-lista.               |
| list-style-position | specifica dove posizionare il marcatore            |
| list-style-image    | specifica il tipo di marcatore elemento-elenco     |
| iniziale            | imposta questa proprietà sul valore predefinito    |
| ereditare           | eredita questa proprietà dal suo elemento genitore |

## Osservazioni

Sebbene il `list-style-type` sia in realtà una proprietà che si applica solo agli elementi di elenco (normalmente `<li>`), viene spesso specificato per il tag di elenco ( `<ol>` o `<ul>` ). In questo caso, gli elementi dell'elenco ereditano la proprietà.

## Examples

### Tipo di pallottola o numerazione

Specifico per i tag `<li>` all'interno di un elenco non ordinato ( `<ul>` ):

```
list-style: disc;           /* A filled circle (default) */
list-style: circle;        /* A hollow circle */
list-style: square;        /* A filled square */
list-style: '-';           /* any string */
```

Specifico per i tag `<li>` all'interno di un elenco ordinato ( `<ol>` ):

```
list-style: decimal;       /* Decimal numbers beginning with 1 (default) */
list-style: decimal-leading-zero; /* Decimal numbers padded by initial zeros (01, 02, 03, ... 10) */
list-style: lower-roman;   /* Lowercase roman numerals (i., ii., iii., iv., ...) */
list-style: upper-roman;   /* Uppercase roman numerals (I., II., III., IV., ...) */
list-style-type: lower-greek; /* Lowercase roman letters (α., β., γ., δ., ...) */
list-style-type: lower-alpha; /* Lowercase letters (a., b., c., d., ...) */
```

```
list-style-type: lower-latin; /* Lowercase letters (a., b., c., d., ...) */
list-style-type: upper-alpha; /* Uppercase letters (A., B., C., D., ...) */
list-style-type: upper-latin; /* Uppercase letters (A., B., C., D., ...) */
```

Non specifico:

```
list-style: none; /* No visible list marker */
list-style: inherit; /* Inherits from parent */
```

## Posizione di proiettile

Una lista è composta da elementi `<li>` all'interno di un elemento contenente ( `<ul>` o `<ol>` ). Sia gli elementi dell'elenco che il contenitore possono avere margini e paddings che influenzano la posizione esatta del contenuto dell'elemento dell'elenco nel documento. I valori predefiniti per il margine e il riempimento potrebbero essere diversi per ciascun browser. Per ottenere lo stesso cross-browser di layout, questi devono essere impostati in modo specifico.

Ogni elemento della lista riceve una "casella di marcatura", che contiene il segnalino punto. Questa casella può essere posizionata all'interno o all'esterno della casella di elenco.

```
list-style-position: inside;
```

posiziona il punto elenco all'interno dell'elemento `<li>` , spingendo il contenuto a destra secondo necessità.

```
list-style-position: outside;
```

posiziona il punto a sinistra dell'elemento `<li>` . Se non c'è abbastanza spazio nel padding dell'elemento contenitore, la casella del marker si estenderà a sinistra anche se cadrebbe dalla pagina.

Mostrando il risultato del posizionamento `inside` ed `outside` : [jsfiddle](#)

## Rimozione di elenchi puntati / numeri

A volte, una lista non deve semplicemente mostrare punti elenco o numeri. In tal caso, ricorda di specificare il margine e il riempimento.

```
<ul>
  <li>first item</li>
  <li>second item</li>
</ul>
```

## CSS

```
ul {
  list-style-type: none;
}
li {
```

```
margin: 0;  
padding: 0;  
}
```

Leggi Elenca stili online: <https://riptutorial.com/it/css/topic/4215/elenca-stili>

---

# Capitolo 21: Eredità

## Sintassi

- *proprietà*: eredita;

## Examples

### Ereditarietà automatica

Ereditarietà: un meccanismo fondamentale del CSS con cui i valori calcolati di alcune proprietà di un elemento vengono applicati ai suoi "figli". Ciò è particolarmente utile quando si desidera impostare uno stile globale per i propri elementi piuttosto che dover impostare dette proprietà su ogni singolo elemento del markup.

Le proprietà comuni che vengono automaticamente ereditate sono: `font` , `color` , `text-align` , `line-height` .

Assumi il seguente foglio di stile:

```
#myContainer {
  color: red;
  padding: 5px;
}
```

Questo applicherà il `color: red` non solo all'elemento `<div>` ma anche agli elementi `<h3>` e `<p>` . Tuttavia, a causa della natura del `padding` suo valore **non** sarà ereditato da tali elementi.

```
<div id="myContainer">
  <h3>Some header</h3>
  <p>Some paragraph</p>
</div>
```

### Ereditarietà forzata

Alcune proprietà non vengono automaticamente ereditate da un elemento fino ai suoi "figli". Questo perché in genere tali proprietà sono desiderate essere univoche per l'elemento (o la selezione di elementi) a cui viene applicata la proprietà. Proprietà comuni come `margin` , `padding` , `background` , `display` , **ecc.**

Tuttavia, a volte l'eredità è desiderata comunque. Per ottenere ciò, possiamo applicare il valore `inherit` alla proprietà che dovrebbe essere ereditata. Il valore `inherit` può essere appiati a *qualsiasi* proprietà CSS e *qualsiasi* elemento HTML.

Assumi il seguente foglio di stile:

```
#myContainer {
  color: red;
  padding: 5px;
}
#myContainer p {
  padding: inherit;
}
```

Questo applicherà il `color: red` ad entrambi gli elementi `<h3>` e `<p>` causa della natura ereditaria della proprietà `color`. Tuttavia, l'elemento `<p>` erediterà anche il valore `padding` dal suo 'parent' perché questo è stato specificato.

```
<div id="myContainer">
  <h3>Some header</h3>
  <p>Some paragraph</p>
</div>
```

Leggi Eredità online: <https://riptutorial.com/it/css/topic/3586/eredita>

---

# Capitolo 22: Forme ad elementi singoli

## Examples

### Piazza

Per creare un quadrato, definisci un elemento con larghezza e altezza. Nell'esempio seguente, abbiamo un elemento con una `width` e `height` di 100 pixel ciascuna.



```
<div class="square"></div>
```

```
.square {  
  width: 100px;  
  height: 100px;  
  background: rgb(246, 156, 85);  
}
```

### triangoli

Per creare un triangolo CSS, definisci un elemento con larghezza e altezza di 0 pixel. La forma del triangolo sarà formata usando le proprietà del bordo. Per un elemento con altezza e larghezza 0 i 4 bordi (in alto, a destra, in basso, a sinistra) formano ciascuno un triangolo. Ecco un elemento con 0 altezza / larghezza e 4 diversi bordi colorati.



Impostando alcuni bordi su trasparente e altri su un colore, possiamo creare vari triangoli. Ad esempio, nel triangolo Su, impostiamo il bordo inferiore sul colore desiderato, quindi impostiamo i bordi sinistro e destro su trasparente. Ecco un'immagine con i bordi sinistro e destro leggermente ombreggiati per mostrare come si forma il triangolo.



Le dimensioni del triangolo possono essere modificate modificando le diverse larghezze dei bordi: più alte, più corte, inclinate, ecc. Gli esempi seguenti mostrano un triangolo di 50x50 pixel.

### Triangolo - Puntamento verso l'alto



```
<div class="triangle-up"></div>
```

```
.triangle-up {  
  width: 0;  
  height: 0;  
  border-left: 25px solid transparent;  
  border-right: 25px solid transparent;  
  border-bottom: 50px solid rgb(246, 156, 85);  
}
```

### Triangolo - Puntamento verso il basso



```
<div class="triangle-down"></div>
```

```
.triangle-down {  
  width: 0;  
  height: 0;  
  border-left: 25px solid transparent;  
  border-right: 25px solid transparent;  
  border-top: 50px solid rgb(246, 156, 85);  
}
```

### Triangolo - Puntamento a destra



```
<div class="triangle-right"></div>
```

```
.triangle-right {  
  width: 0;  
  height: 0;  
  border-top: 25px solid transparent;  
  border-bottom: 25px solid transparent;  
  border-left: 50px solid rgb(246, 156, 85);  
}
```

## Triangolo - Puntamento a sinistra



```
<div class="triangle-left"></div>
```

```
.triangle-left {  
  width: 0;  
  height: 0;  
  border-top: 25px solid transparent;  
  border-bottom: 25px solid transparent;  
  border-right: 50px solid rgb(246, 156, 85);  
}
```

## Triangolo - Puntamento verso l'alto / verso destra



```
<div class="triangle-up-right"></div>
```

```
.triangle-up-right {  
  width: 0;  
  height: 0;  
  border-top: 50px solid rgb(246, 156, 85);  
  border-left: 50px solid transparent;  
}
```

## Triangolo - Puntamento verso l'alto / sinistra



```
<div class="triangle-up-left"></div>
```

```
.triangle-up-left {  
  width: 0;  
  height: 0;  
  border-top: 50px solid rgb(246, 156, 85);  
  border-right: 50px solid transparent;  
}
```

## Triangolo - Puntamento verso il basso / destra



```
<div class="triangle-down-right"></div>
```

```
.triangle-down-right {  
  width: 0;  
  height: 0;  
  border-bottom: 50px solid rgb(246, 156, 85);  
  border-left: 50px solid transparent;  
}
```

## Triangolo - Puntamento verso il basso / sinistra



```
<div class="triangle-down-left"></div>
```

```
.triangle-down-left {  
  width: 0;  
  height: 0;  
  border-bottom: 50px solid rgb(246, 156, 85);  
  border-right: 50px solid transparent;  
}
```

```
}
```

## Bursts

Una raffica è simile a una stella ma con i punti che si estendono meno dal corpo. Pensa a una forma a raffica come un quadrato con altri quadrati leggermente ruotati sovrapposti.

I quadrati aggiuntivi vengono creati usando gli elementi psuedo `::before` e `::after`.

### 8 punti Burst

Una raffica di 8 punti sono 2 quadrati sovrapposti. Il quadrato inferiore è l'elemento stesso, il quadrato aggiuntivo viene creato usando lo pseudo-elemento `:before`. Il fondo è ruotato di 20 °, il quadrato superiore è ruotato di 135 °.



```
<div class="burst-8"></div>

.burst-8 {
  background: rgb(246, 156, 85);
  width: 40px;
  height: 40px;
  position: relative;
  text-align: center;
  -ms-transform: rotate(20deg);
  transform: rotate(20deg);
}

.burst-8::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  height: 40px;
  width: 40px;
  background: rgb(246, 156, 85);
  -ms-transform: rotate(135deg);
  transform: rotate(135deg);
}
```

### 12 punti Burst

Una raffica di 12 punti sono 3 quadrati sovrapposti. Il quadrato inferiore è l'elemento stesso, i quadrati aggiuntivi vengono creati usando gli pseudo-elementi `:before` e `:after`. Il fondo viene ruotato di 0 °, il quadrato successivo viene ruotato di 30 ° e la parte superiore viene ruotata di 60 °.



```
<div class="burst-12"></div>

.burst-12 {
  width: 40px;
  height: 40px;
  position: relative;
  text-align: center;
  background: rgb(246, 156, 85);
}

.burst-12::before, .burst-12::after {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  height: 40px;
  width: 40px;
  background: rgb(246, 156, 85);
}

.burst-12::before {
  -ms-transform: rotate(30deg);
  transform: rotate(30deg);
}

.burst-12::after {
  -ms-transform: rotate(60deg);
  transform: rotate(60deg);
}
```

## Cerchi ed ellissi

### Cerchio

Per creare un **cerchio**, definire un elemento con `width` e `height` uguali (un *quadrato*) e quindi impostare la proprietà `border-radius` di questo elemento su `50%`.



#### HTML

```
<div class="circle"></div>
```

## CSS

```
.circle {  
  width: 50px;  
  height: 50px;  
  background: rgb(246, 156, 85);  
  border-radius: 50%;  
}
```

## Ellisse

Un'ellisse è simile a un cerchio, ma con valori diversi per `width` e `height`.



## HTML

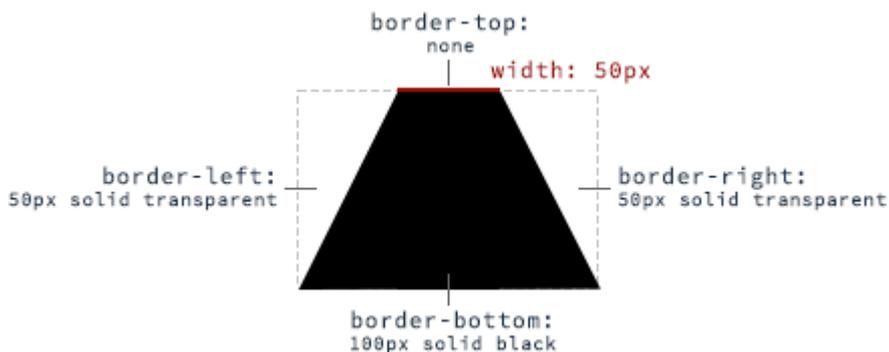
```
<div class="oval"></div>
```

## CSS

```
.oval {  
  width: 50px;  
  height: 80px;  
  background: rgb(246, 156, 85);  
  border-radius: 50%;  
}
```

## trapezio

Un trapezio può essere creato da un elemento di blocco con altezza zero (altezza di `0px`), una larghezza maggiore di zero e un bordo, che è trasparente ad eccezione di un lato:



HTML:

```
<div class="trapezoid"></div>
```

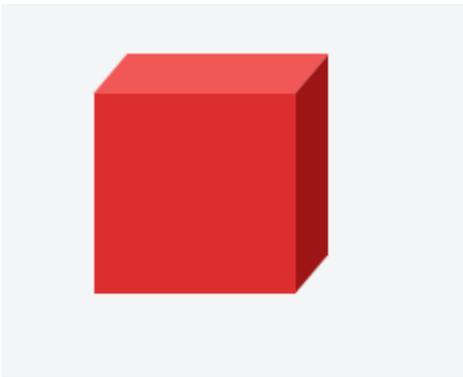
## CSS:

```
.trapezoid {  
  width: 50px;  
  height: 0;  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-bottom: 100px solid black;  
}
```

Con la modifica dei lati del bordo, è possibile regolare l'orientamento del trapezio.

## Cubo

Questo esempio mostra come creare un cubo usando i metodi di trasformazione 2D `skewX()` e `skewY()` su pseudo elementi.



## HTML:

```
<div class="cube"></div>
```

## CSS:

```
.cube {  
  background: #dc2e2e;  
  width: 100px;  
  height: 100px;  
  position: relative;  
  margin: 50px;  
}  
  
.cube::before {  
  content: '';  
  display: inline-block;  
  background: #f15757;  
  width: 100px;  
  height: 20px;  
  transform: skewX(-40deg);  
  position: absolute;  
  top: -20px;  
  left: 8px;
```

```

}

.cube::after {
  content: '';
  display: inline-block;
  background: #9e1515;
  width: 16px;
  height: 100px;
  transform: skewY(-50deg);
  position: absolute;
  top: -10px;
  left: 100%;
}

```

[Guarda la demo](#)

## Piramide

Questo esempio mostra come creare una **piramide** usando i bordi e i metodi di trasformazione 2D `skewY()` e `rotate()` su pseudo elementi.



### HTML:

```
<div class="pyramid"></div>
```

### CSS:

```

.pyramid {
  width: 100px;
  height: 200px;
  position: relative;
  margin: 50px;
}

.pyramid::before, .pyramid::after {
  content: '';
  display: inline-block;
  width: 0;
  height: 0;
  border: 50px solid;
  position: absolute;
}

.pyramid::before {
  border-color: transparent transparent #ff5656 transparent;
}

```

```
transform: scaleY(2) skewY(-40deg) rotate(45deg);
}

.pyramid::after {
border-color: transparent transparent #d64444 transparent;
transform: scaleY(2) skewY(40deg) rotate(-45deg);
}
```

Leggi Forme ad elementi singoli online: <https://riptutorial.com/it/css/topic/2862/forme-ad-elementi-singoli>

# Capitolo 23: Forme per galleggianti

## Sintassi

- fuori forma: nessuno | [<forma base> || <shape box>] | <Image>
- margine di forma: <lunghezza> | <Percentuale>
- shape-image-threshold: <numero>

## Parametri

| Parametro    | Dettagli  |
|--------------|---|
| nessuna      | Un valore pari a <code>none</code> significa che l'area float (l'area utilizzata per avvolgere il contenuto attorno a un elemento float) non è interessata. Questo è il valore predefinito / iniziale.  |
| basic-figura | Si riferisce a uno tra <code>inset()</code> , <code>circle()</code> , <code>ellipse()</code> o <code>polygon()</code> . Usando una di queste funzioni e i suoi valori viene definita la forma.  |
| Forma-box    | Si riferisce a uno tra <code>margin-box</code> , <code>border-box</code> , <code>padding-box</code> , <code>content-box</code> . Quando viene fornito solo <shape-box> (senza <basic-shape>) questa casella è la forma. Quando viene utilizzato insieme a <forma base>, funge da riquadro di riferimento. |
| Immagine     | Quando un'immagine viene fornita come valore, la forma viene calcolata in base al canale alfa dell'immagine specificata.  |

## Osservazioni

Il supporto del browser per il modulo CSS Shapes è molto limitato in questo momento.

È supportato in Chrome v37+ e Opera 24+ senza prefissi browser / fornitore. Safari lo supporta dalla v7.1+ ma con il prefisso `-webkit-`.

Non è ancora supportato in IE, Edge e Firefox.

## Examples

### Forma all'esterno con forma base - cerchio ()

Con la proprietà CSS `shape-outside` possibile definire i valori di forma per l'area float in modo che il contenuto inline ricopra la forma anziché la casella del float.

## CSS

```
img:nth-of-type(1) {
  shape-outside: circle(80px at 50% 50%);
  float: left;
  width: 200px;
}
img:nth-of-type(2) {
  shape-outside: circle(80px at 50% 50%);
  float: right;
  width: 200px;
}
p {
  text-align: center;
  line-height: 30px; /* purely for demo */
}
```

## HTML

```

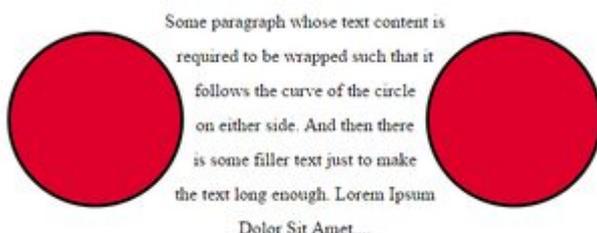

<p>Some paragraph whose text content is required to be wrapped such that it follows the curve
of the circle on either side. And then there is some filler text just to make the text long
enough. Lorem Ipsum Dolor Sit Amet....</p>
```

Nell'esempio sopra, entrambe le immagini sono in realtà immagini quadrate e quando il testo viene posizionato senza la proprietà `shape-outside`, non scorrerà attorno al cerchio su entrambi i lati. Scorrerà intorno alla scatola contenente solo l'immagine. Con la `shape-outside` dell'area del float viene ri-definita come un *cerchio* e il contenuto viene fatto fluire attorno a questo *cerchio immaginario* creato utilizzando la `shape-outside`.

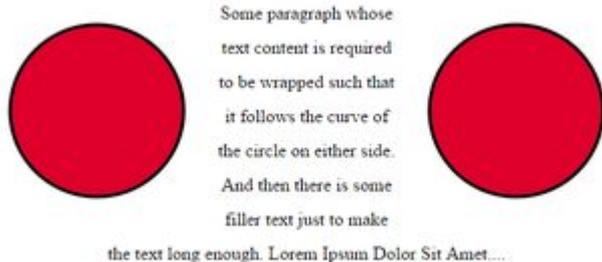
Il *cerchio immaginario* che viene utilizzato per ridefinire l'area fluttuante è un cerchio con raggio di 80 px disegnato dal centro-centro del riquadro di riferimento dell'immagine.

Di seguito sono riportati alcuni screenshot per illustrare in che modo il contenuto verrà avvolto quando viene utilizzato `shape-outside` e quando non viene utilizzato.

### Uscita con `shape-outside`



### Uscita senza `shape-outside`



## Margine di forma

La `shape-margin` proprietà CSS aggiunge un *margin*e di `shape-outside` .

## CSS

```
img:nth-of-type(1) {
  shape-outside: circle(80px at 50% 50%);
  shape-margin: 10px;
  float: left;
  width: 200px;
}
img:nth-of-type(2) {
  shape-outside: circle(80px at 50% 50%);
  shape-margin: 10px;
  float: right;
  width: 200px;
}
p {
  text-align: center;
  line-height: 30px; /* purely for demo */
}
```

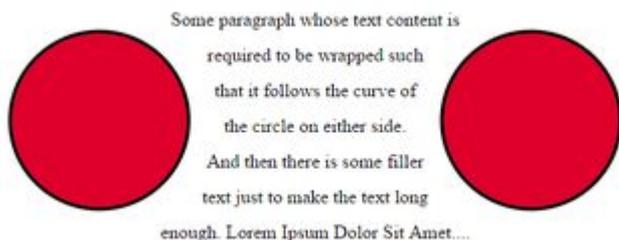
## HTML

```


<p>Some paragraph whose text content is required to be wrapped such that it follows the curve
of the circle on either side. And then there is some filler text just to make the text long
enough. Lorem Ipsum Dolor Sit Amet....</p>
```

In questo esempio, un margine di 10 pixel viene aggiunto attorno alla **forma** utilizzando il `shape-margin` . Questo crea un po 'più di spazio tra il *cerchio immaginario* che definisce l'area fluttuante e il contenuto reale che scorre intorno.

## Produzione:



Leggi Forme per galleggianti online: <https://riptutorial.com/it/css/topic/2034/forme-per-galleggianti>

# Capitolo 24: funzioni

## Sintassi

- `<calc()> = calc( <calc-sum> )`
- `<calc-sum> = <calc-product> [ [ '+' | '-' ] <calc-product> ]*`
- `<calc-product> = <calc-value> [ '*' <calc-value> | '/' <number> ]*`
- `<calc-value> = <number> | <dimension> | <percentage> | ( <calc-sum> )`

## Osservazioni

Per `calc()`, è richiesto uno spazio bianco attorno agli operatori " - " e " + ", ma non agli operatori " \* " o " / ".

Tutte le unità devono essere dello stesso tipo; tentare di moltiplicare l'altezza per una durata, ad esempio, non è valido.

## Examples

### funzione `calc()`

Accetta un'espressione matematica e restituisce un valore numerico.

È particolarmente utile quando si lavora con diversi tipi di unità (ad esempio, sottraendo un valore px da una percentuale) per calcolare il valore di un attributo.

È possibile utilizzare tutti gli operatori + , - , / , e \* e è possibile aggiungere parentesi per specificare l'ordine delle operazioni, se necessario.

Usa `calc()` per calcolare la larghezza di un elemento div:

```
#div1 {
  position: absolute;
  left: 50px;
  width: calc(100% - 100px);
  border: 1px solid black;
  background-color: yellow;
  padding: 5px;
  text-align: center;
}
```

Usa `calc()` per determinare la posizione di un'immagine di sfondo:

```
background-position: calc(50% + 17px) calc(50% + 10px), 50% 50%;
```

Usa `calc()` per determinare l'altezza di un elemento:

```
height: calc(100% - 20px);
```

## funzione attr ()

Restituisce il valore di un attributo dell'elemento selezionato.

Di seguito è riportato un elemento blockquote che contiene un carattere all'interno di un [attributo data-\\*](#) che il CSS può utilizzare (ad esempio all'interno dello [pseudo-elemento](#) `::before` e `::after`) utilizzando questa funzione.

```
<blockquote data-mark=""></blockquote>
```

Nel seguente blocco CSS, il carattere viene aggiunto prima e dopo il testo all'interno dell'elemento:

```
blockquote[data-mark]::before,  
blockquote[data-mark]::after {  
    content: attr(data-mark);  
}
```

## funzione linear-gradient ()

Crea un'immagine che rappresenta un gradiente lineare di colori.

```
linear-gradient( 0deg, red, yellow 50%, blue);
```

Questo crea una sfumatura che va dal basso verso l'alto, con i colori che iniziano con il rosso, poi il giallo al 50% e la finitura in blu.

## funzione radial-gradient ()

Crea un'immagine che rappresenta un gradiente di colori che si irradiano dal centro del gradiente

```
radial-gradient(red, orange, yellow) /*A gradient coming out from the middle of the  
gradient, red at the center, then orange, until it is finally yellow at the edges*/
```

## funzione var ()

La funzione var () consente l'accesso alle variabili CSS.

```
/* set a variable */  
:root {  
    --primary-color: blue;  
}  
  
/* access variable */  
selector {  
    color: var(--primary-color);  
}
```

Questa funzione è attualmente in fase di sviluppo. Controlla [caniuse.com](https://caniuse.com) per l'ultimo supporto del browser.

Leggi funzioni online: <https://riptutorial.com/it/css/topic/2214/funzioni>

---

# Capitolo 25: Galleggianti

## Sintassi

- chiaro: nessuno | sinistra | giusto | entrambi | inline-start | inline-end;
- float: sinistra | giusto | nessuno | inline-start | inline-end;

## Osservazioni

Poiché float implica l'uso del layout a blocchi, in alcuni casi modifica il valore calcolato dei valori di visualizzazione [1]

[1]: <https://developer.mozilla.org/en-US/docs/Web/CSS/float> MDN

## Examples

### Galleggiare un'immagine all'interno del testo

L'uso più basilare di un oggetto mobile è che il testo si avvolge attorno a un'immagine. Il codice seguente produrrà due paragrafi e un'immagine, con il secondo paragrafo che scorre intorno all'immagine. Si noti che è sempre contenuto *dopo* l'elemento floated che scorre attorno all'elemento floated.

HTML:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla. </p>



<p>Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur sodales ligula in libero. Sed dignissim lacinia nunc. Curabitur tortor. Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel, suscipit quis, luctus non, massa. Fusce ac turpis quis ligula lacinia aliquet. </p>
```

CSS:

```
img {
  float:left;
  margin-right:1rem;
}
```

Questo sarà l'output

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.



Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur sodales ligula in libero.

Sed dignissim lacinia nunc. Curabitur tortor. Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel, suscipit quis, luctus non, massa. Fusce ac turpis quis ligula lacinia aliquet.

[Codepen Link](#)

### Semplice layout a due colonne a larghezza fissa

Un semplice layout a due colonne è costituito da due elementi fluttuanti a larghezza fissa. Notare che la barra laterale e l'area del contenuto non hanno la stessa altezza in questo esempio. Questa è una delle parti più complesse con layout a più colonne che utilizzano i float e richiede soluzioni alternative per far apparire più colonne con la stessa altezza.

HTML:

```

<div class="wrapper">

<div class="sidebar">
  <h2>Sidebar</h2>

  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio.</p>
</div>

<div class="content">
  <h1>Content</h1>

  <p>Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos
himenaeos. Curabitur sodales ligula in libero. Sed dignissim lacinia nunc. Curabitur tortor.
Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis
tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel,
suscipit quis, luctus non, massa. Fusce ac turpis quis ligula lacinia aliquet. </p>
</div>

</div>

```

## CSS:

```

.wrapper {
  width:600px;
  padding:20px;
  background-color:pink;

  /* Floated elements don't use any height. Adding "overflow:hidden;" forces the
  parent element to expand to contain its floated children. */
  overflow:hidden;
}

.sidebar {
  width:150px;
  float:left;
  background-color:blue;
}

.content {
  width:450px;
  float:right;
  background-color:yellow;
}

```

## Semplice layout a tre colonne a larghezza fissa

### HTML:

```

<div class="wrapper">
  <div class="left-sidebar">
    <h1>Left Sidebar</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
  </div>
  <div class="content">
    <h1>Content</h1>
    <p>Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos
himenaeos. Curabitur sodales ligula in libero. Sed dignissim lacinia nunc. Curabitur tortor.
Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis

```

```

tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel,
suscipit quis, luctus non, massa. </p>
</div>
<div class="right-sidebar">
  <h1>Right Sidebar</h1>
  <p>Fusce ac turpis quis ligula lacinia aliquet.</p>
</div>
</div>

```

## CSS:

```

.wrapper {
  width:600px;
  background-color:pink;
  padding:20px;

  /* Floated elements don't use any height. Adding "overflow:hidden;" forces the
  parent element to expand to contain its floated children. */
  overflow:hidden;
}

.left-sidebar {
  width:150px;
  background-color:blue;
  float:left;
}

.content {
  width:300px;
  background-color:yellow;
  float:left;
}

.right-sidebar {
  width:150px;
  background-color:green;
  float:right;
}

```

## Layout pigro / avido a due colonne

Questo layout utilizza una colonna flottata per creare un layout a due colonne senza larghezze definite. In questo esempio la barra laterale sinistra è "pigra", in quanto occupa solo lo spazio necessario. Un altro modo per dire questo è che la barra laterale sinistra è "termoretraibile". La colonna del contenuto giusto è "golosa", in quanto occupa tutto lo spazio rimanente.

## HTML:

```

<div class="sidebar">
  <h1>Sidebar</h1>
  
</div>

<div class="content">
  <h1>Content</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero.

```

```
Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla. </p>
<p>Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur sodales ligula in libero. Sed dignissim lacinia nunc. Curabitur tortor. Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel, suscipit quis, luctus non, massa. Fusce ac turpis quis ligula lacinia aliquet. Mauris ipsum. Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh. </p>
</div>
```

## CSS:

```
.sidebar {
  /* `display:table;` shrink-wraps the column */
  display:table;
  float:left;
  background-color:blue;
}

.content {
  /* `overflow:hidden;` prevents `.content` from flowing under `.sidebar` */
  overflow:hidden;
  background-color:yellow;
}
```

## Violino

### chiara proprietà

La proprietà clear è direttamente correlata ai float. Valori di proprietà:

- nessuno - Predefinito. Consente elementi mobili su entrambi i lati
- sinistra - Nessun elemento flottante consentito sul lato sinistro
- a destra: non sono ammessi elementi mobili sul lato destro
- entrambi - Nessun elemento flottante consentito sul lato sinistro o destro
- iniziale: imposta questa proprietà sul valore predefinito. Leggi l'iniziale
- inherit: eredita questa proprietà dall'elemento padre. Leggi sull'eredità

```
<html>
<head>
<style>
img {
  float: left;
}

p.clear {
  clear: both;
}
</style>
</head>
<body>


<p>Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum
```

```

Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum </p>
<p class="clear">Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum
Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum Lorem ipsum </p>

</body>
</html>

```

## clearfix

L'hack clearfix è un modo popolare per contenere i float (N. Gallagher aka @necolas)

Da non confondere con la proprietà `clear`, clearfix è un *concetto* (che è anche legato ai float, quindi alla possibile confusione). Per *contenere i float*, devi aggiungere la classe `.cf` o `.clearfix` sul contenitore ( **il genitore** ) e `.clearfix` uno stile a questa classe con alcune regole descritte di seguito.

3 versioni con effetti leggermente diversi (fonti: [un nuovo micro clearfix hack](#) di N. Gallagher e [clearfix ricaricato](#) da TJ Koblentz):

---

## Clearfix (con il margine superiore di compressione dei float ancora presenti)

```

.cf:after {
  content: "";
  display: table;
}

.cf:after {
  clear: both;
}

```

---

## Clearfix impedisce anche il collasso dei margini superiori dei float contenuti

```

/**
 * For modern browsers
 * 1. The space content is one way to avoid an Opera bug when the
 *    contenteditable attribute is included anywhere else in the document.
 *    Otherwise it causes space to appear at the top and bottom of elements
 *    that are clearfixed.
 * 2. The use of `table` rather than `block` is only necessary if using
 *    `:before` to contain the top-margins of child elements.
 */
.cf:before,
.cf:after {
  content: " "; /* 1 */
  display: table; /* 2 */
}

```

```
.cf:after {
  clear: both;
}
```

---

## Clearfix con supporto di browser obsoleti IE6 e IE7

```
.cf:before,
.cf:after {
  content: " ";
  display: table;
}

.cf:after {
  clear: both;
}

/**
 * For IE 6/7 only
 * Include this rule to trigger hasLayout and contain floats.
 */
.cf {
  *zoom: 1;
}
```

---

[Codepen mostra l'effetto clearfix](#)

---

Altra risorsa: [tutto ciò che sai di clearfix è sbagliato](#) (clearfix e BFC - Block Formatting Context mentre hasLayout si riferisce a browser obsoleti IE6 forse 7)

### DIV in linea usando float

Il `div` è un elemento a livello di blocco, cioè occupa l'intera larghezza della pagina e i fratelli sono posizionati uno sotto l'altro indipendentemente dalla loro larghezza.

```
<div>
  <p>This is DIV 1</p>
</div>
<div>
  <p>This is DIV 2</p>
</div>
```

L'output del seguente codice sarà

This is DIV 1

This is DIV 2

Possiamo renderli in linea aggiungendo una proprietà `float` css al `div` .

HTML:

```
<div class="outer-div">
  <div class="inner-div1">
    <p>This is DIV 1</p>
  </div>
  <div class="inner-div2">
    <p>This is DIV 2</p>
  </div>
</div>
```

CSS

```
.inner-div1 {
  width: 50%;
  margin-right:0px;
  float:left;
```

```
background : #337ab7;
padding:50px 0px;
}

.inner-div2 {
width: 50%;
margin-right:0px;
float:left;
background : #dd2c00;
padding:50px 0px;
}

p {
text-align:center;
}
```

This is DIV 1

[Codepen Link](#)

## Uso della proprietà di overflow per cancellare i float

Impostando il valore di `overflow` su `hidden`, `auto` o `scroll` su un elemento, verranno cancellati tutti i float all'interno di quell'elemento.

**Nota:** utilizzando l' `overflow:scroll` mostrerà sempre la scrollbar

Leggi Galleggianti online: <https://riptutorial.com/it/css/topic/405/galleggianti>

# Capitolo 26: Griglia

## introduzione

Il layout a griglia è un nuovo e potente sistema di layout CSS che consente di dividere un contenuto di una pagina Web in righe e colonne in modo semplice.

## Osservazioni

Il modulo [Livello di griglia CSS Livello 1](#) è, a partire dal 9 settembre 2016, una Raccomandazione del Candidato W3C. È considerato in fase di test ( <https://www.w3.org/Style/CSS/current-work> ).

A partire dal 3 luglio 2017, i browser Internet Explorer 10 e 11 di Microsoft e Edge supportano solo una versione precedente delle specifiche utilizzando un prefisso del fornitore.

## Examples

### Esempio di base

| Proprietà | Valori possibili           |
|-----------|----------------------------|
| display   | griglia / griglia in linea |

La griglia CSS è definita come proprietà di visualizzazione. Si applica solo a un elemento genitore e ai suoi figli immediati.

Considera il seguente markup:

```
<section class="container">
  <div class="item1">item1</div>
  <div class="item2">item2</div>
  <div class="item3">item3</div>
  <div class="item4">item4</div>
</section>
```

Il modo più semplice per definire la struttura di markup sopra come una griglia è semplicemente impostare la proprietà di `display` sulla `grid` :

```
.container {
  display: grid;
}
```

Tuttavia, facendo ciò invariabilmente causerà il collasso di tutti gli elementi figlio uno sopra l'altro. Questo perché i bambini attualmente non sanno come posizionarsi all'interno della griglia. Ma possiamo dire esplicitamente a loro.

Per prima cosa dobbiamo dire all'elemento della griglia `.container` quante righe e colonne compongono la sua struttura e possiamo farlo usando le proprietà `grid-columns` e `grid-rows` (notare la pluralizzazione):

```
.container {
  display: grid;
  grid-columns: 50px 50px 50px;
  grid-rows: 50px 50px;
}
```

Tuttavia, questo ancora non ci aiuta molto perché dobbiamo dare un ordine a ciascun elemento figlio. Possiamo farlo specificando i valori della `grid-row` `grid-column` e della `grid-column` che indicano dove si trova nella griglia:

```
.container .item1 {
  grid-column: 1;
  grid-row: 1;
}
.container .item2 {
  grid-column: 2;
  grid-row: 1;
}
.container .item3 {
  grid-column: 1;
  grid-row: 2;
}
.container .item4 {
  grid-column: 2;
  grid-row: 2;
}
```

Dando a ciascun articolo un valore di colonna e riga identifica l'ordine degli articoli all'interno del contenitore.

Visualizza un esempio funzionante su [JSFiddle](#) . Dovrai visualizzarlo in IE10, IE11 o Edge perché funzioni come sono attualmente gli unici browser che supportano il Grid Layout (con il prefisso del fornitore `-ms-` ) o abilitano un flag in Chrome, Opera e Firefox in base a [caniuse](#) in ordine per testare con loro.

Leggi Griglia online: <https://riptutorial.com/it/css/topic/2152/griglia>

---

# Capitolo 27: Hack di Internet Explorer

## Osservazioni

Questi "hack" possono essere utilizzati per indirizzare un browser / client specifico. Questo può essere usato per aggirare le differenze di rendering del browser applicando gli stili in uno di quei wrapper elencati sopra.

## Examples

### Modalità Contrasto elevato in Internet Explorer 10 e versioni successive

In Internet Explorer 10+ e Edge, Microsoft offre il selettore di mezzi multimediali ad `-ms-high-contrast` per esporre l'impostazione "Contrasto elevato" dal browser, che consente al programmatore di regolare gli stili del proprio sito di conseguenza.

Il selettore `-ms-high-contrast` ha 3 stati: `active`, `black-on-white` e `white-on-black`. In IE10 + non aveva `none` stato, ma non è più supportato in Edge in futuro.

## Esempi

```
@media screen and (-ms-high-contrast: active), (-ms-high-contrast: black-on-white) {
  .header{
    background: #fff;
    color: #000;
  }
}
```

Ciò cambierà lo sfondo dell'intestazione in bianco e il colore del testo in nero quando la modalità a contrasto elevato è attiva e si trova in modalità `black-on-white`.

```
@media screen and (-ms-high-contrast: white-on-black) {
  .header{
    background: #000;
    color: #fff;
  }
}
```

Simile al primo esempio, ma in particolare seleziona solo lo stato `white-on-black` e inverte i colori dell'intestazione su uno sfondo nero con testo bianco.

---

## Maggiori informazioni:

[Documentazione Microsoft](#) su `-ms-high-contrast`

## Solo Internet Explorer 6 e Internet Explorer 7

Per scegliere come target Internet Explorer 6 e Internet Explorer 7, avvia le tue proprietà con `*` :

```
.hide-on-ie6-and-ie7 {  
    *display : none; // This line is processed only on IE6 and IE7  
}
```

I prefissi non alfanumerici (diversi da trattini e caratteri di sottolineatura) vengono ignorati in IE6 e IE7, quindi questa modifica funziona per qualsiasi `property: value unprefixed property: value` coppia di `property: value` .

## Solo Internet Explorer 8

Per `@media \0 screen { }` target Internet Explorer 8, racchiudi i selettori nella `@media \0 screen { }` :

```
@media \0 screen {  
    .hide-on-ie8 {  
        display : none;  
    }  
}
```

Tutto tra `@media \0 screen { }` viene elaborato solo da I

## Aggiunta del supporto Inline Block a IE6 e IE7

```
display: inline-block;
```

La proprietà di `display` con il valore di `inline-block` non è supportata da Internet Explorer 6 e 7. Una soluzione per questo è:

```
zoom: 1;  
*display: inline;
```

La proprietà `zoom` attiva la funzione `hasLayout` degli elementi ed è disponibile solo in Internet Explorer. Il `*display` assicura che la proprietà non valida venga eseguita solo sui browser interessati. Gli altri browser ignoreranno semplicemente la regola.

Leggi Hack di Internet Explorer online: <https://riptutorial.com/it/css/topic/5056/hack-di-internet-explorer>

---

# Capitolo 28: I selettori

## introduzione

I selettori CSS identificano specifici elementi HTML come target per gli stili CSS. Questo argomento spiega in che modo i selettori CSS hanno come target elementi HTML. I selezionatori usano una vasta gamma di oltre 50 metodi di selezione offerti dal linguaggio CSS, inclusi elementi, classi, ID, pseudo-elementi e pseudo-classi e modelli.

## Sintassi

- `# id`
- `. nome della classe`
- `: pseudo-classname`
- `:: pseudo-elementname`
- `[ attr ] / * ha l'attributo attr . * /`
- `[ attr = " valore " ] / * ha l'attributo attr , e il suo valore è esattamente " valore " . * /`
- `[ attr ~ = " valore " ] / * ha l'attributo attr , e il suo valore, quando diviso su spazi , contiene " valore " . * /`
- `[ attr | = " valore " ] / * ha l'attributo attr , e il suo valore è esattamente " valore " , o il suo valore inizia con " valore - " . * /`
- `[ attr ^ = " valore " ] / * ha l'attributo attr , e il suo valore inizia con " valore " . * /`
- `[ attr $ = " valore " ] / * ha l'attributo attr e il suo valore termina con " valore " . * /`
- `[ attr * = " valore " ] / * ha l'attributo attr , e il suo valore contiene " valore " . * /`
- `Elemento-name`
- `*`

## Osservazioni

- A volte vedrai i doppi due punti ( `::` ) invece di uno solo ( `:` . Questo è un modo per separare le [pseudo-classi](#) dagli [pseudo-elementi](#) .
- Vecchi browser, come Internet Explorer 8, supportano **solo** un singolo due punti ( `:` ) per la definizione di pseudo-elementi.
- A differenza delle pseudo-classi, può essere usato un solo pseudo-elemento per selettore, se presente deve apparire dopo la sequenza di selettori semplici che rappresentano i soggetti del selettore (una versione futura della [specificazione W3C](#) può consentire più pseudo-elementi per selettore ).

## Examples

### Selettori di attributi

---

# Panoramica

I selettori di attributi possono essere utilizzati con vari tipi di operatori che modificano di conseguenza i criteri di selezione. Selezionano un elemento utilizzando la presenza di un determinato attributo o valore di attributo.

| Selettore (1)  | Elemento abbinato          | Seleziona elementi ...   | Versione CSS |
|----------------|----------------------------|--|--------------|
| [attr]         | <div attr>                 | Con attributo attr   | 2            |
| [attr='val']   | <div attr="val">           | Dove l'attributo attr ha valore val  | 2            |
| [attr~='val']  | <div attr="val val2 val3"> | Dove val appare nel lista separata da spazi di attr  | 2            |
| [attr^='val']  | <div attr="val1 val2">     | Dove il valore di attr inizia con val  | 3            |
| [attr\$='val'] | <div attr="sth aval">      | Dove il valore di attr termina con val   | 3            |
| [attr*='val']  | <div attr="somevalhere">   | Dove attr contiene val ovunque   | 3            |
| [attr ='val']  | <div attr="val-sth etc">   | Dove il valore di attr è esattamente val , o inizia con val e immediatamente seguito da - (U + 002D) | 2            |
| [attr='val' i] | <div attr="val">           | Dove attr ha valore val , ignorando il rivestimento delle lettere di val .                           | 4 (2)        |

## Gli appunti:

1. Il valore dell'attributo può essere circondato da virgolette singole o doppie virgolette. Nessuna virgola può anche funzionare, ma non è valida secondo lo standard CSS ed è scoraggiata.
2. Non esiste un'unica specifica CSS4 integrata, perché è divisa in moduli separati. Tuttavia, ci sono moduli di "livello 4". [Vedi il supporto del browser](#) .

## Dettagli

[attribute]

Seleziona elementi con l'attributo dato.

```
div[data-color] {  
  color: red;  
}
```

```
<div data-color="red">This will be red</div>  
<div data-color="green">This will be red</div>  
<div data-background="red">This will NOT be red</div>
```

## [Live Demo su JSBin](#)

**[attribute="value"]**

Seleziona elementi con l'attributo e il valore specificati.

```
div[data-color="red"] {  
  color: red;  
}
```

```
<div data-color="red">This will be red</div>  
<div data-color="green">This will NOT be red</div>  
<div data-color="blue">This will NOT be red</div>
```

## [Live Demo su JSBin](#)

**[attribute\*="value"]**

Seleziona gli elementi con l'attributo e il valore specificati in cui l'attributo dato contiene il valore dato da qualsiasi parte (come sottostringa).

```
[class*="foo"] {  
  color: red;  
}
```

```
<div class="foo-123">This will be red</div>  
<div class="foo123">This will be red</div>  
<div class="bar123foo">This will be red</div>  
<div class="barfoo123">This will be red</div>  
<div class="barfo0">This will NOT be red</div>
```

## [Live Demo su JSBin](#)

**[attribute~="value"]**

Seleziona gli elementi con l'attributo e il valore specificati in cui il valore indicato viene visualizzato in un elenco separato da spazi.

```
[class~="color-red"] {  
  color: red;  
}
```

```
<div class="color-red foo-bar the-div">This will be red</div>
```

```
<div class="color-blue foo-bar the-div">This will NOT be red</div>
```

## Live Demo su JSBin

**[attribute^="value"]**

Seleziona gli elementi con l'attributo e il valore specificati in cui l'attributo dato inizia con il valore.

```
[class^="foo-"] {  
  color: red;  
}
```

```
<div class="foo-123">This will be red</div>  
<div class="foo-234">This will be red</div>  
<div class="bar-123">This will NOT be red</div>
```

## Live Demo su JSBin

**[attribute\$="value"]**

Seleziona gli elementi con l'attributo e il valore specificati in cui l'attributo dato termina con il valore dato.

```
[class$="file"] {  
  color: red;  
}
```

```
<div class="foobar-file">This will be red</div>  
<div class="foobar-file">This will be red</div>  
<div class="foobar-input">This will NOT be red</div>
```

## Live Demo su JSBin

**[attribute|="value"]**

Seleziona gli elementi con un determinato attributo e valore dove il valore dell'attributo è esattamente il valore dato o è esattamente il valore dato seguito da - (U + 002D)

```
[lang|="EN"] {  
  color: red;  
}
```

```
<div lang="EN-us">This will be red</div>  
<div lang="EN-gb">This will be red</div>  
<div lang="PT-pt">This will NOT be red</div>
```

## Live Demo su JSBin

**[attribute="value" i]**

Seleziona elementi con un dato attributo e un valore in cui il valore dell'attributo può essere rappresentato come `Value` , `VALUE` , `vAlUe` o qualsiasi altra possibilità maiuscole e minuscole.

```
[lang="EN" i] {  
  color: red;  
}
```

```
<div lang="EN">This will be red</div>  
<div lang="en">This will be red</div>  
<div lang="PT">This will NOT be red</div>
```

[Live Demo su JSBin](#)

## Specificità dei selettori di attributi

0-1-0

Come selettore di classe e pseudoclass.

```
*[type=checkbox] // 0-1-0
```

Nota che questo significa che un selettore di attributo può essere usato per selezionare un elemento dal suo ID a un livello inferiore di specificità che se fosse stato selezionato con un [selettore ID](#) : `[id="my-ID"]` lo stesso numero di `#my-ID` ma con specificità inferiore.

Vedi la [sezione Sintassi](#) per maggiori dettagli.

combinatori

## Panoramica

| Selettore                  | Descrizione  |
|----------------------------|--|
| <code>div span</code>      | Selettore discendente (tutti <code>&lt;span&gt;</code> s discendenti di <code>&lt;div&gt;</code> )                         |
| <code>div &gt; span</code> | Selettore di bambini (tutti <code>&lt;span&gt;</code> che sono figli diretti di <code>&lt;div&gt;</code> )                 |
| <code>a ~ span</code>      | Selettore General Sibling (tutti <code>&lt;span&gt;</code> s che sono fratelli dopo un <code>&lt;a&gt;</code> )            |
| <code>a + span</code>      | Selettore di Sibling adiacente (tutti i <code>&lt;span&gt;</code> che sono immediatamente dopo un <code>&lt;a&gt;</code> ) |

**Nota:** i selettori di Sibling scelgono come target gli elementi che vengono dopo di essi nel documento di origine. Il CSS, per sua natura (a cascata), non può bersagliare elementi *precedenti* o *genitore* . Tuttavia, usando la proprietà `order flex`, [un precedente](#)

selettore di pari livello può essere simulato su supporti visivi .

## Combinatore discendente: `selector selector`

Un combinatore discendente, rappresentato da almeno un carattere di spazio ( ), seleziona gli elementi che sono discendenti dell'elemento definito. Questo combinatore seleziona **tutti i** discendenti dell'elemento (dagli elementi figli in basso).

```
div p {
  color:red;
}
```

```
<div>
  <p>My text is red</p>
  <section>
    <p>My text is red</p>
  </section>
</div>

<p>My text is not red</p>
```

[Live Demo su JSBin](#)

Nell'esempio precedente, i primi due elementi `<p>` sono selezionati poiché sono entrambi i discendenti di `<div>` .

## Combinatore di bambini: `selector > selector`

Il combinatore child ( > ) viene utilizzato per selezionare elementi **figli** , o **discendenti diretti** , dell'elemento specificato.

```
div > p {
  color:red;
}
```

```
<div>
  <p>My text is red</p>
  <section>
    <p>My text is not red</p>
  </section>
</div>
```

[Live Demo su JSBin](#)

Il precedente CSS seleziona solo il primo elemento `<p>` , in quanto è l'unico paragrafo direttamente discendente da un `<div>` .

Il secondo elemento `<p>` non è selezionato perché non è un figlio diretto di `<div>` .

---

## Combinatore Sibling adiacente: `selector + selector`

Il combinatore adiacente di pari livello ( + ) seleziona un elemento fratello che segue immediatamente un elemento specificato.

```
p + p {
  color:red;
}
```

```
<p>My text is not red</p>
<p>My text is red</p>
<p>My text is red</p>
<hr>
<p>My text is not red</p>
```

[Live Demo su JSBin](#)

L'esempio sopra seleziona solo quegli elementi `<p>` che sono *direttamente preceduti* da un altro elemento `<p>` .

---

---

## General Sibling Combinator: `selector ~ selector`

Il combinatore generale ( ~ ) seleziona *tutti i fratelli* che seguono l'elemento specificato.

```
p ~ p {
  color:red;
}
```

```
<p>My text is not red</p>
<p>My text is red</p>
<hr>
<h1>And now a title</h1>
<p>My text is red</p>
```

[Live Demo su JSBin](#)

L'esempio sopra seleziona tutti gli elementi `<p>` *preceduti* da un altro elemento `<p>` , indipendentemente dal fatto che siano immediatamente adiacenti.

## Selettori del nome di classe

Il selettore del nome della classe seleziona tutti gli elementi con il nome della classe mirata. Ad esempio, il nome della classe `.warning` selezionerebbe il seguente elemento `<div>` :

```
<div class="warning">
  <p>This would be some warning copy.</p>
</div>
```

Puoi anche combinare i nomi delle classi con gli elementi target in modo più specifico. Costruiamo sull'esempio sopra per mostrare una selezione di classi più complicata.

## CSS

```
.important {
  color: orange;
}
.warning {
  color: blue;
}
.warning.important {
  color: red;
}
```

## HTML

```
<div class="warning">
  <p>This would be some warning copy.</p>
</div>

<div class="important warning">
  <p class="important">This is some really important warning copy.</p>
</div>
```

In questo esempio, tutti gli elementi con la classe `.warning` avranno un colore di testo blu, gli elementi con la classe `.important` avranno un colore di testo arancione e tutti gli elementi che hanno *sia* il nome di classe `.important` che `.warning` avranno un testo rosso colore.

Si noti che all'interno del CSS, la dichiarazione `.warning.important` non ha spazi tra i due nomi di classe. Ciò significa che troverà solo elementi che contengono sia i nomi di classe che `warning` e `important` nel loro attributo di `class`. Quei nomi di classe potrebbero essere in qualsiasi ordine sull'elemento.

Se fosse incluso uno spazio tra le due classi nella dichiarazione CSS, selezionerebbe solo elementi che hanno elementi parent con un `.warning` classe `.warning` ed elementi figlio con nomi di classe `.important`.

## Selettori ID

I selettori ID selezionano gli elementi DOM con l'ID target. Per selezionare un elemento con un ID specifico in CSS, viene utilizzato il prefisso `#`.

Ad esempio, il seguente elemento `div` HTML ...

```
<div id="exampleID">
  <p>Example</p>
</div>
```

... può essere selezionato da `#exampleID` in CSS come mostrato di seguito:

```
#exampleID {
  width: 20px;
}
```

**Nota** : le specifiche HTML non consentono più elementi con lo stesso ID

## Pseudo-classi

Le **pseudo-classi** sono **parole chiave** che consentono la selezione in base a informazioni che si trovano all'esterno dell'albero del documento o che non possono essere espresse da altri selettori o combinatori. Questa informazione può essere associato ad un certo stato ( **stato** e **dinamiche** pseudo-classi), a posizioni ( **strutturali** e **indirizzare** pseudo-classi), per negazioni della ex ( **negazione** pseudo-classe) o alle lingue ( **Lang** pseudo-classe). Gli esempi includono se un collegamento è stato seguito o meno ( `:visited` ), il mouse si trova su un elemento ( `:hover` ), una casella di controllo è selezionata ( `:checked` ), ecc.

## Sintassi

```
selector:pseudo-class {
  property: value;
}
```

## Elenco di pseudo-classi:

| Nome                   | Descrizione   |
|------------------------|---|
| <code>:active</code>   | Si applica a qualsiasi elemento che viene attivato (cioè cliccato) dall'utente.   |
| <code>:any</code>      | Ti permette di creare gruppi di selettori correlati creando gruppi che il gli articoli inclusi corrisponderanno. Questa è un'alternativa alla ripetizione di un intero selettore. |
| <code>:target</code>   | Seleziona l'elemento <code>#news</code> attivo corrente (cliccato su un URL contenente quel nome di ancoraggio)   |
| <code>:checked</code>  | Si applica alla radio, alla casella di controllo o agli elementi delle opzioni che sono selezionati o attivato in uno stato "on".   |
| <code>:default</code>  | Rappresenta qualsiasi elemento dell'interfaccia utente che è l'impostazione predefinita tra un gruppo di elementi simili.   |
| <code>:disabled</code> | Si applica a qualsiasi elemento dell'interfaccia utente che si trova in uno stato disabilitato.   |
| <code>:empty</code>    | Si applica a qualsiasi elemento che non ha figli.   |

| Nome                        | Descrizione   |
|-----------------------------|---|
| <code>:enabled</code>       | Si applica a qualsiasi elemento dell'interfaccia utente che si trova in uno stato abilitato.  |
| <code>:first</code>         | Utilizzato insieme alla regola <code>@page</code> , seleziona la prima pagina di a documento stampato   |
| <code>:first-child</code>   | Rappresenta qualsiasi elemento che è il primo elemento figlio del suo genitore.   |
| <code>:first-of-type</code> | Si applica quando un elemento è il primo del tipo di elemento selezionato all'interno del suo genitore. Questo può o non può essere il primo figlio.  |
| <code>:focus</code>         | Si applica a qualsiasi elemento che ha l'attenzione dell'utente. Questo può essere dato dal tastiera dell'utente, eventi del mouse o altre forme di input.  |
| <code>:focus-within</code>  | Può essere usato per evidenziare un'intera sezione quando un elemento al suo interno è focalizzato. Corrisponde a qualsiasi elemento che corrisponde alle corrispondenze pseudo-classe: <code>focus</code> o che ha un focus focalizzato.   |
| <code>:full-screen</code>   | Si applica a qualsiasi elemento visualizzato in modalità a schermo intero. Seleziona l'intero stack di elementi e non solo l'elemento di livello superiore.   |
| <code>:hover</code>         | Si applica a qualsiasi elemento al passaggio del mouse dal dispositivo di puntamento dell'utente, ma non attivato.  |
| <code>:indeterminate</code> | Applica elementi dell'interfaccia utente radio o casella di controllo che non sono né controllati né deselezionati, ma sono in uno stato indeterminato. Questo può essere dovuto a un attributo dell'elemento o manipolazione DOM.  |
| <code>:in-range</code>      | La <code>:in-range</code> pseudo-classe CSS <code>:in-range</code> corrisponde quando un elemento ha il suo attributo <code>value</code> all'interno delle limitazioni di intervallo specificate per questo elemento. Permette alla pagina di dare un feedback sul valore attualmente definito l'utilizzo dell'elemento è all'interno dei limiti dell'intervallo. |
| <code>:invalid</code>       | Si applica a <code>&lt;input&gt;</code> elementi i cui valori non sono validi in base a il tipo specificato nell'attributo <code>type=</code> .   |
| <code>:lang</code>          | Si applica a qualsiasi elemento che include l'elemento <code>&lt;body&gt;</code> correttamente <code>lang=</code> designato <code>lang=</code> attributo. Perché la pseudo-classe sia valida, deve contenere un <a href="#">codice lingua valido di due o tre lettere</a> .   |

| Nome                       | Descrizione   |
|----------------------------|---|
| <code>:last-child</code>   | Rappresenta qualsiasi elemento che è l'ultimo elemento figlio del suo genitore.   |
| <code>:last-of-type</code> | Si applica quando un elemento è l'ultimo del tipo di elemento selezionato all'interno suo genitore. Questo può o non può essere l'ultimo figlio.  |
| <code>:left</code>         | Utilizzato in congiunzione con la regola <code>@page</code> , seleziona tutto il sinistro pagine in un documento stampato.  |
| <code>:link</code>         | Si applica a tutti i collegamenti che non sono stati visitati dall'utente.  |
| <code>:not()</code>        | Si applica a tutti gli elementi che <b>non</b> corrispondono al valore passato a ( <code>:not(p)</code> o <code>:not(.class-name)</code> ) per esempio. Deve avere un valore per essere valido e può contenere solo un selettore. Tuttavia, puoi concatenare più elementi <code>:not</code> selettori insieme.  |
| <code>:nth-child</code>    | Si applica quando un elemento è l' $n$ -esimo elemento del suo genitore, dove $n$ può essere un numero intero, un'espressione matematica (ad es. $n+3$ ) o le parole chiave <code>odd</code> o <code>even</code> .  |
| <code>:nth-of-type</code>  | Si applica quando un elemento è l' $n$ -esimo elemento del suo genitore di stesso tipo di elemento, dove $n$ può essere un numero intero, un matematico espressione (ad esempio $n+3$ ), o le parole chiave <code>odd</code> o <code>even</code> .  |
| <code>:only-child</code>   | La pseudo-classe CSS <code>:only-child</code> rappresenta qualsiasi elemento che è l'unico figlio del suo genitore. Questo è lo stesso di <code>:first-child:last-child</code> o <code>:nth-child(1):nth-last-child(1)</code> , ma con una specificità inferiore.   |
| <code>:optional</code>     | La <code>:optional</code> pseudo-classe CSS <code>:optional</code> rappresenta qualsiasi elemento su cui non è impostato l'attributo richiesto. Questo permette forme per indicare facilmente campi opzionali e per modellarli di conseguenza.  |
| <code>:out-of-range</code> | La pseudo-classe CSS <code>:out-of-range</code> coincide quando un elemento ha la sua attributo valore al di fuori dei limiti di intervallo specificati per questo elemento. Permette alla pagina di dare un feedback sul valore attualmente definito usando il elemento non rientra nei limiti dell'intervallo. Un valore può essere al di fuori di un intervallo, se lo è |

| Nome                            | Descrizione  |
|---------------------------------|--|
|                                 | o più piccoli o più grandi dei valori massimi e minimi.  |
| <code>:placeholder-shown</code> | <b>Sperimentale.</b> Si applica a qualsiasi elemento del modulo che attualmente visualizza testo segnaposto.     |
| <code>:read-only</code>         | Si applica a qualsiasi elemento che non è modificabile dall'utente.  |
| <code>:read-write</code>        | Si applica a qualsiasi elemento modificabile da un utente, come ad esempio <code>&lt;input&gt;</code> elementi.  |
| <code>:right</code>             | Utilizzato insieme alla regola <code>@page</code> , seleziona tutte le pagine giuste in a documento stampato     |
| <code>:root</code>              | corrisponde all'elemento radice di un albero che rappresenta il documento.                                       |
| <code>:scope</code>             | La pseudo-classe CSS corrisponde agli elementi che sono un riferimento punto per selezionare i selettori contro. |
| <code>:target</code>            | Seleziona l'elemento <code>#news</code> attivo corrente (cliccato su un URL contenente quel nome di ancoraggio)  |
| <code>:visited</code>           | Si applica a tutti i collegamenti che sono stati visitati dall'utente.   |

Lo pseudoclass `:visited` non può più essere utilizzato per la maggior parte degli stili in molti browser moderni perché è un buco di sicurezza. Vedi questo [link](#) per riferimento.

## Selettori di base

| Selettore                     | Descrizione  |
|-------------------------------|--|
| <code>*</code>                | Selettore universale (tutti gli elementi)  |
| <code>div</code>              | Selettore di tag (tutti gli elementi <code>&lt;div&gt;</code> )                                    |
| <code>.blue</code>            | Selettore di classe (tutti gli elementi con classe <code>blue</code> )                             |
| <code>.blue.red</code>        | Tutti gli elementi con classe <code>blue</code> e <code>red</code> (un tipo di selettore Compound) |
| <code>#headline</code>        | Selettore ID (l'elemento con l'attributo "id" impostato sul <code>headline</code> )                |
| <code>:pseudo-class</code>    | Tutti gli elementi con pseudo-classe   |
| <code>::pseudo-element</code> | Elemento che corrisponde allo pseudo-elemento  |
| <code>:lang(en)</code>        | Elemento che corrisponde a: lang declaration, ad esempio <code>&lt;span lang="en"&gt;</code>       |
| <code>div &gt; p</code>       | selettore di bambini   |

**Nota:** il valore di un ID deve essere univoco in una pagina Web. È una violazione dello [standard HTML](#) utilizzare il valore di un ID più di una volta nella stessa struttura del documento.

Un elenco completo di selettori può essere trovato nella specifica [CSS Selectors Level 3](#).

## Come plasmare un input Range

### HTML

```
<input type="range"></input>
```

### CSS

| Effetto                       | Pseudo selettore   |
|-------------------------------|--|
| Pollice                       | <code>input[type=range]::-webkit-slider-thumb, input[type=range]::-moz-range-thumb, input[type=range]::-ms-thumb</code>                                  |
| Traccia                       | <code>input[type=range]::-webkit-slider-runnable-track, input[type=range]::-moz-range-track, input[type=range]::-ms-track</code>                         |
| OnFocus                       | <code>input[type=range]:focus</code>   |
| Parte inferiore della traccia | <code>input[type=range]::-moz-range-progress, input[type=range]::-ms-fill-lower</code><br>(non possibile nei browser WebKit attualmente - JS necessario) |

## Global boolean con checkbox: checked e ~ (combinatore di fratellanza generale)

Con il selettore ~, puoi facilmente implementare un booleano accessibile globale senza utilizzare JavaScript.

## Aggiungi booleano come casella di controllo

All'inizio del tuo documento, aggiungi tutti i booleani che vuoi con un `id` univoco e il set di attributi `hidden`:

```
<input type="checkbox" id="sidebarShown" hidden />
<input type="checkbox" id="darkThemeUsed" hidden />

<!-- here begins actual content, for example: -->
<div id="container">
  <div id="sidebar">
    <!-- Menu, Search, ... -->
  </div>

  <!-- Some more content ... -->
</div>
```

```
<div id="footer">
  <!-- ... -->
</div>
```

## Cambia il valore del booleano

Puoi cambiare il valore booleano aggiungendo `label` con il set di attributi `for` :

```
<label for="sidebarShown">Show/Hide the sidebar!</label>
```

## Accesso al valore booleano con CSS

Il normale selettore (come `.color-red`) specifica le proprietà predefinite. Possono essere sovrascritti seguendo i selettori `true` / `false` :

```
/* true: */
<checkbox>:checked ~ [sibling of checkbox & parent of target] <target>

/* false: */
<checkbox>:not(:checked) ~ [sibling of checkbox & parent of target] <target>
```

Nota che `<checkbox>`, `[sibling ...]` e `<target>` dovrebbero essere sostituiti dai selettori appropriati. `[sibling ...]` può essere un selettore specifico, (spesso se sei pigro) semplicemente `*` o nulla se l'obiettivo è già un fratello della casella di controllo.

Esempi per la struttura HTML di cui sopra sarebbero:

```
#sidebarShown:checked ~ #container #sidebar {
  margin-left: 300px;
}

#darkThemeUsed:checked ~ #container,
#darkThemeUsed:checked ~ #footer {
  background: #333;
}
```

## In azione

Vedi [questo violino](#) per l'implementazione di questi booleani globali.

### CSS3: esempio di selettore all'interno del campo

```
<style>
input:in-range {
  border: 1px solid blue;
}
```

```
</style>
```

```
<input type="number" min="10" max="20" value="15">  
<p>The border for this value will be blue</p>
```

La `:in-range` pseudo-classe CSS `:in-range` corrisponde quando un elemento ha il suo attributo `value` all'interno delle limitazioni di intervallo specificato per questo elemento. Consente alla pagina di fornire un feedback sul fatto che il valore attualmente definito utilizzando l'elemento si trova all'interno dei limiti dell'intervallo. [1]

## Pseudo classe bambino

"La pseudo-classe CSS: `nth-child(an + b)` corrisponde ad un elemento che ha un fratello  $+ b - 1$  prima nell'albero del documento, per un dato **valore** positivo **o zero** per  $n$ " - [MDN: nth-child](#)

| pseudo-selettore                | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---------------------------------|---|---|---|---|---|---|---|---|---|----|
| <code>:first-child</code>       | ✓ |   |   |   |   |   |   |   |   |    |
| <code>:nth-child(3)</code>      |   |   | ✓ |   |   |   |   |   |   |    |
| <code>:nth-child(n+3)</code>    |   |   | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓  |
| <code>:nth-child(3n)</code>     |   |   | ✓ |   |   | ✓ |   |   | ✓ |    |
| <code>:nth-child(3n+1)</code>   | ✓ |   |   | ✓ |   |   | ✓ |   |   | ✓  |
| <code>:nth-child(-n+3)</code>   | ✓ | ✓ | ✓ |   |   |   |   |   |   |    |
| <code>:nth-child(odd)</code>    | ✓ |   | ✓ |   | ✓ |   | ✓ |   | ✓ |    |
| <code>:nth-child(even)</code>   |   | ✓ |   | ✓ |   | ✓ |   | ✓ |   | ✓  |
| <code>:last-child</code>        |   |   |   |   |   |   |   |   |   | ✓  |
| <code>:nth-last-child(3)</code> |   |   |   |   |   |   | ✓ |   |   |    |

## Seleziona l'elemento usando il suo ID senza l'alta specificità del selettore ID

Questo trucco ti aiuta a selezionare un elemento usando l'ID come valore per un selettore di attributi per evitare l'alta specificità del selettore ID.

HTML:

```
<div id="element">...</div>
```

## CSS

```
#element { ... } /* High specificity will override many selectors */  
[id="element"] { ... } /* Low specificity, can be overridden easily */
```

## A. L'esempio: non pseudo-classe e B.: focus-all'interno della pseudo-classe CSS

A. La sintassi è presentata sopra.

Il seguente selettore corrisponde a tutti gli elementi `<input>` in un documento HTML che non sono disabilitati e non hanno la classe `.example` :

HTML:

```
<form>  
  Phone: <input type="tel" class="example">  
  E-mail: <input type="email" disabled="disabled">  
  Password: <input type="password">  
</form>
```

CSS:

```
input:not([disabled]):not(.example){  
  background-color: #ccc;  
}
```

La pseudo-classe `:not()` supporterà anche i selettori separati da virgola in Selectors Livello 4:

CSS:

```
input:not([disabled], .example){  
  background-color: #ccc;  
}
```

[Live Demo su JSBin](#)

Vedi la sintassi dello sfondo [qui](#) .

## B. II: focus-all'interno della pseudo-classe CSS

HTML:

```
<h3>Background is blue if the input is focused .</p>  
<div>  
  <input type="text">  
</div>
```

CSS:

```
div {
  height: 80px;
}
input {
  margin: 30px;
}
div:focus-within {
  background-color: #1565C0;
}
```

```
div {
  height: 80px;
}
input {
  margin: 30px;
}
div:focus-within {
  background-color: #1565C0;
}
```

Background is blue if the input is focused .



## # :focus-within CSS pseudo-class UNOFF

The **:focus-within** pseudo-class matches elements that either themselves match **:focus** or that have descendants which match **:focus**.

Current aligned
Usage relative
Date relative
Show all

| IE | Edge <sup>*</sup> | Firefox | Chrome          | Safari | Op             |
|----|-------------------|---------|-----------------|--------|----------------|
|    |                   | 52      | 49              |        |                |
|    | 14                | 53      | 58              |        | 4              |
| 11 | 15                | 54      | 59 <sup>1</sup> | 10.1   | 4 <sup>1</sup> |
|    | 16                | 55      | 60              | 11     | 4 <sup>1</sup> |
|    |                   | 56      | 61              | TP     | 4 <sup>1</sup> |
|    |                   | 57      | 62              |        |                |

[Notes](#)
[Known issues \(0\)](#)
[Resources \(11\)](#)
[Feedback](#)

<sup>1</sup> Can be enabled via the "Experimental Web Platform Features" flag

## L'esempio del selettore pseudo-classe: only-child

La pseudo-classe CSS `:only-child` rappresenta qualsiasi elemento che è l'unico figlio del suo genitore.

HTML:

```
<div>
  <p>This paragraph is the only child of the div, it will have the color blue</p>
</div>

<div>
  <p>This paragraph is one of the two children of the div</p>
  <p>This paragraph is one of the two children of its parent</p>
</div>
```

CSS:

```
p:only-child {
  color: blue;
}
```

L'esempio sopra seleziona l'elemento `<p>` che è il figlio unico dal suo genitore, in questo caso un `<div>`.

[Live Demo su JSBin](#)

## II: ultimo tipo di selettore

Il `:last-of-type` seleziona l'elemento che è l'ultimo figlio, di un particolare tipo, del suo genitore. Nell'esempio seguente, il css seleziona l'ultimo paragrafo e l'ultima intestazione `h1`.

```
p:last-of-type {
  background: #C5CAE9;
}
h1:last-of-type {
  background: #CDDC39;
}
```

```
<div class="container">
  <p>First paragraph</p>
  <p>Second paragraph</p>
  <p>Last paragraph</p>
  <h1>Heading 1</h1>
  <h2>First heading 2</h2>
  <h2>Last heading 2</h2>
</div>
```

First paragraph  
Second paragraph  
Last paragraph  
Heading 1  
First heading 2  
Last heading 2

[jsFiddle](#)

Leggi i selettori online: <https://riptutorial.com/it/css/topic/611/i-selettori>

---

# Capitolo 29: Il modello di scatola

## Sintassi

- dimensionamento del riquadro: *parametro* ;

## Parametri

| Parametro                | Dettaglio  |
|--------------------------|--|
| <code>content-box</code> | La larghezza e l'altezza dell'elemento includono solo l'area del contenuto.      |
| <code>padding-box</code> | La larghezza e l'altezza dell'elemento includono il contenuto e il riempimento.  |
| <code>border-box</code>  | La larghezza e l'altezza dell'elemento includono contenuto, riempimento e bordo. |
| <code>initial</code>     | Imposta il modello box al suo stato predefinito.                                 |
| <code>inherit</code>     | Inherisce il modello di box dell'elemento genitore.                              |

## Osservazioni

---

### A proposito di `padding-box`

Questo valore è stato implementato solo da **Firefox** e quindi non dovrebbe essere utilizzato. È stato rimosso in Firefox versione 50.0.

## Examples

Qual è il modello di scatola?

---

### I bordi

Il browser crea un rettangolo per ogni elemento nel documento HTML. Il modello di casella descrive come il riempimento, il bordo e il margine vengono aggiunti al contenuto per creare questo rettangolo.

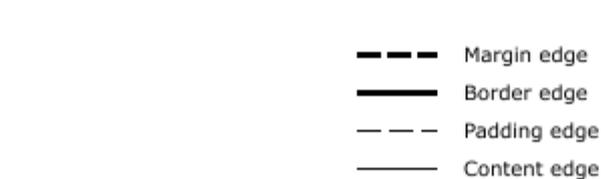
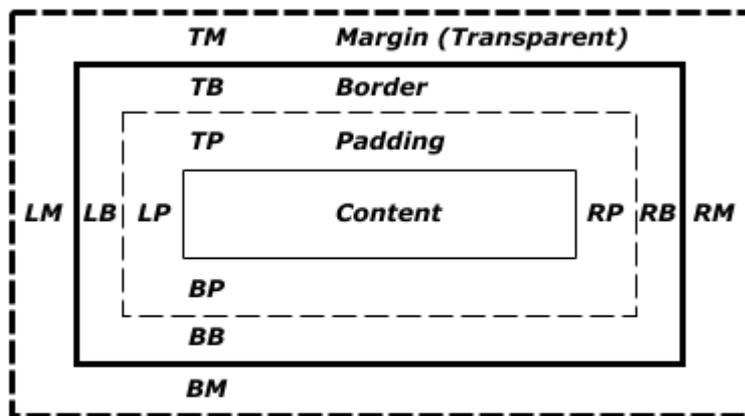


Diagramma da [CSS2.2 Bozza di lavoro](#)

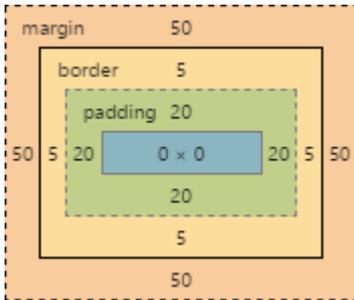
Il perimetro di ciascuna delle quattro aree è chiamato *bordo*. Ogni bordo definisce una *scatola*.

- Il rettangolo più interno è il **riquadro dei contenuti**. La larghezza e l'altezza di questo dipendono dal contenuto reso dell'elemento (testo, immagini e qualsiasi elemento figlio che possa avere).
- Il prossimo è il **riquadro di riempimento**, come definito dalla proprietà `padding`. Se non è definita alcuna larghezza di `padding`, il bordo di riempimento è uguale al bordo del contenuto.
- Quindi abbiamo la **casella di confine**, come definito dalla proprietà `border`. Se non c'è `border` larghezza definita, il limite del bordo è uguale al limite del cuscinetto.
- Il rettangolo più esterno è il **margin**, come definito dalla proprietà del `margin`. Se non v'è alcun `margin` larghezza definita, il bordo margin è uguale al limite del bordo.

## Esempio

```
div {
  border: 5px solid red;
  margin: 50px;
  padding: 20px;
}
```

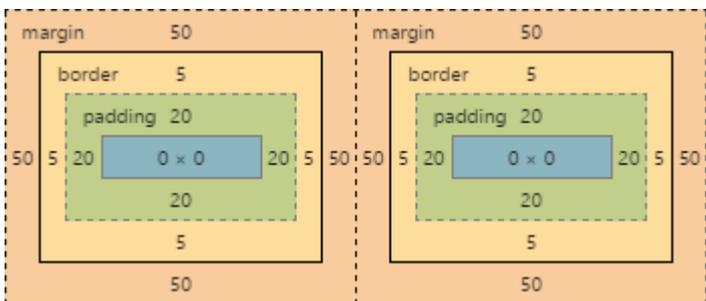
Questo CSS usa tutti gli elementi `div` per avere un bordo superiore, destro, inferiore e sinistro di `5px` in larghezza; un margine superiore, destro, inferiore e sinistro di `50px`; e una imbottitura superiore, destra, inferiore e sinistra di `20px`. Ignorando il contenuto, la nostra casella generata sarà simile a questa:



Screenshot del pannello Stili degli elementi di Google Chrome

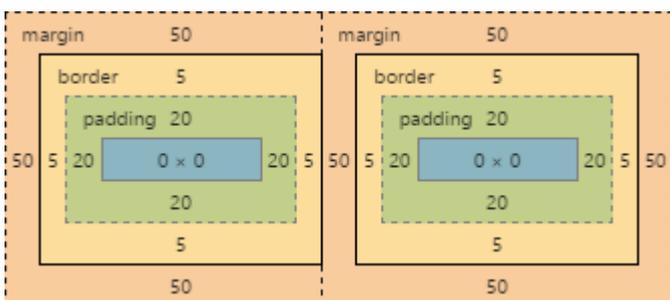
- Poiché non c'è contenuto, l'area del contenuto (la casella blu al centro) non ha altezza o larghezza (0px per 0px).
- La casella di riempimento di default ha le stesse dimensioni del riquadro del contenuto, più la larghezza di 20px su tutti e quattro i bordi che stiamo definendo sopra con la proprietà `padding` (40px per 40px).
- La casella del bordo ha le stesse dimensioni del riquadro di riempimento, più la larghezza di 5 px che stiamo definendo sopra con la proprietà `border` (50px per 50px).
- Infine, la casella del margine ha le stesse dimensioni del riquadro del bordo, più la larghezza 50px che stiamo definendo sopra con la proprietà del `margin` (dando al nostro elemento una dimensione totale di 150 px per 150 px).

Ora diamo al nostro elemento un fratello con lo stesso stile. Il browser guarda il Box Model di entrambi gli elementi per capire dove, in relazione al contenuto dell'elemento precedente, il nuovo elemento dovrebbe essere posizionato:



Il contenuto di ciascun elemento è separato da uno spazio di 150 px, ma le caselle dei due elementi si toccano.

Se poi modifichiamo il nostro primo elemento per non avere il margine destro, il margine del margine destro si troverà nella stessa posizione del bordo del bordo destro, e i nostri due elementi ora assomigliano a questo:



## box-sizing

Il modello di box predefinito ( `content-box` ) può essere contro-intuitivo, poiché la `width` / `height` per un elemento non rappresenterà la sua larghezza o altezza effettiva sullo schermo non appena inizierai ad aggiungere `padding` e stili di `border` all'elemento.

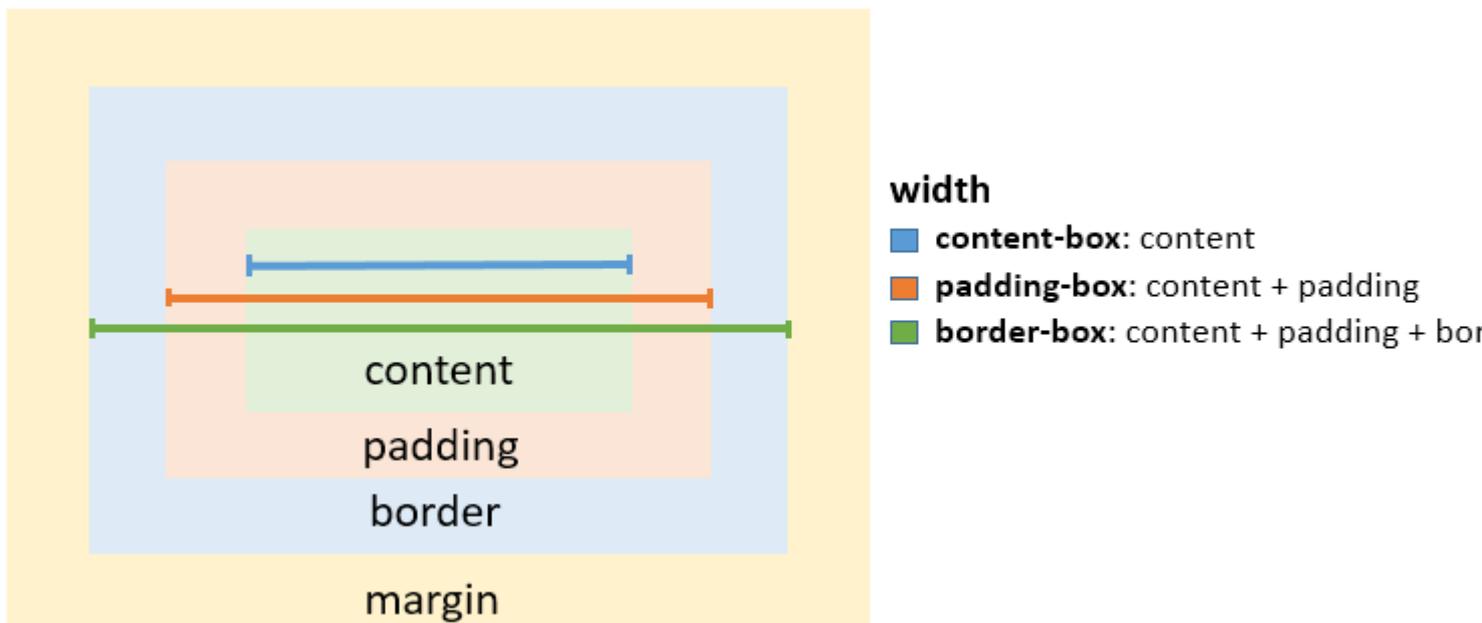
L'esempio seguente dimostra questo potenziale problema con il `content-box` :

```
textarea {
  width: 100%;
  padding: 3px;
  box-sizing: content-box; /* default value */
}
```

Poiché il `padding` verrà aggiunto alla larghezza della `textarea`, l'elemento risultante è una `textarea` che è più larga del 100%.

Fortunatamente, il CSS ci consente di cambiare il modello di box con la proprietà di `box-sizing` del `box-sizing` per un elemento. Sono disponibili tre diversi valori per la proprietà:

- `content-box` : il modello di box comune - larghezza e altezza include solo il contenuto, non il `padding` o il bordo
- `padding-box` : larghezza e altezza includono il contenuto e il `padding`, ma non il bordo
- `border-box` : larghezza e altezza include il contenuto, il `padding` e il bordo



Per risolvere il problema della `textarea` sopra, puoi semplicemente modificare la proprietà di `box-sizing` della `box-sizing` in `padding-box` o `border-box`. `border-box` è più comunemente usato.

```
textarea {
  width: 100%;
```

```
padding: 3px;
box-sizing: border-box;
}
```

Per applicare un modello di box specifico a ogni elemento della pagina, utilizza il seguente frammento:

```
html {
  box-sizing: border-box;
}

*, *:before, *:after {
  box-sizing: inherit;
}
```

In questo `box-sizing: border-box;` codifica `box-sizing: border-box;` non è direttamente applicato a `*`, quindi puoi facilmente sovrascrivere questa proprietà su singoli elementi.

Leggi Il modello di scatola online: <https://riptutorial.com/it/css/topic/646/il-modello-di-scatola>

---

# Capitolo 30: Imbottitura

## Sintassi

- padding: *length* | initial | inherit | unset;
- padding-top: *length* | initial | inherit | unset;
- padding-right: *length* | initial | inherit | unset;
- padding-bottom: *length* | initial | inherit | unset;
- padding-left: *length* | initial | inherit | unset;

## Osservazioni

La proprietà padding imposta lo spazio di riempimento su tutti i lati di un elemento. L'area di riempimento è lo spazio tra il contenuto dell'elemento e il suo bordo. **I valori negativi non sono ammessi** .

1 : <https://developer.mozilla.org/en/docs/Web/CSS/padding> MDN

Vedi anche questa [domanda](#) , "Perché i CSS non supportano il padding negativo?" e le sue risposte.

Inoltre, si prega di prendere in considerazione [The Box Model](#) quando si utilizza il padding. A seconda del valore di dimensionamento della casella, il riempimento su un elemento può o aggiungere all'altezza / larghezza precedentemente definita di un elemento o meno.

Proprietà correlate:

[margine](#)

Il riempimento degli elementi in linea si applica solo a sinistra ea destra dell'elemento e non in alto e in basso, a causa delle proprietà di visualizzazione intrinseche degli elementi in linea.

## Examples

### Imbottitura su un lato

La proprietà padding imposta lo spazio di riempimento su tutti i lati di un elemento. L'area di riempimento è lo spazio tra il contenuto dell'elemento e il suo bordo. I valori negativi non sono ammessi.

Puoi specificare una parte individualmente:

- padding-top
- padding-right
- padding-bottom
- padding-left

Il seguente codice aggiungerebbe un padding di 5px all'inizio del div:

```
<style>
.myClass {
  padding-top: 5px;
}
</style>

<div class="myClass"></div>
```

## Stenografia di imbottitura

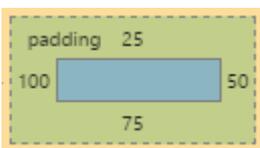
La proprietà padding imposta lo spazio di riempimento su tutti i lati di un elemento. L'area di riempimento è lo spazio tra il contenuto dell'elemento e il suo bordo. I valori negativi non sono ammessi.

Per salvare l'aggiunta di padding su ciascun lato individualmente (usando `padding-top`, `padding-left` ecc.) Puoi scriverlo come una stenografia, come di seguito:

### Quattro valori :

```
<style>
.myDiv {
  padding: 25px 50px 75px 100px; /* top right bottom left; */
}
</style>

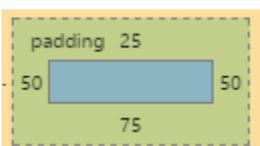
<div class="myDiv"></div>
```



### Tre valori :

```
<style>
.myDiv {
  padding: 25px 50px 75px; /* top left/right bottom */
}
</style>

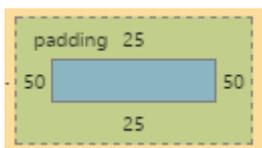
<div class="myDiv"></div>
```



### Due valori :

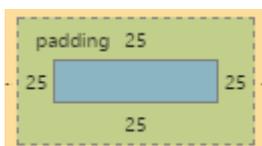
```
<style>
.myDiv {
  padding: 25px 50px; /* top/bottom left/right */
}
</style>
```

```
</style>
<div class="myDiv"></div>
```



## Un valore :

```
<style>
  .myDiv {
    padding: 25px; /* top/right/bottom/left */
  }
</style>
<div class="myDiv"></div>
```



Leggi Imbottitura online: <https://riptutorial.com/it/css/topic/1255/imbottitura>

# Capitolo 31: La frammentazione

## Sintassi

- `page-break-after`: auto | sempre | evitare | sinistra | giusto | iniziale | ereditare;
- `page-break-before`: auto | sempre | evitare | sinistra | giusto | iniziale | ereditare;
- `page-break-inside`: auto | evitare | iniziale | ereditare;

## Parametri

| Valore    | Descrizione   |
|-----------|---|
| auto      | Predefinito. Interruzioni di pagina automatiche   |
| sempre    | Inserisci sempre un'interruzione di pagina  |
| evitare   | Evita l'interruzione di pagina (se possibile)   |
| sinistra  | Inserisci interruzioni di pagina in modo che la pagina successiva sia formattata come pagina sinistra |
| destra    | Inserisci interruzioni di pagina in modo che la pagina successiva sia formattata come pagina destra   |
| iniziale  | Imposta questa proprietà sul valore predefinito.  |
| ereditare | Eredita questa proprietà dal suo elemento padre   |

## Osservazioni

Non c'è alcuna proprietà di interruzione di pagina nei CSS. Solo le 3 proprietà ( **page-break-before** , **page-break-after** , **page-break-inside** ).

Correlati: `orphans` , `widows` .

## Examples

### Media stampa page-break

```
@media print {  
  p {  
    page-break-inside: avoid;  
  }  
  h1 {  
    page-break-before: always;  
  }  
}
```

```
}  
h2 {  
  page-break-after: avoid;  
}  
}
```

Questo codice fa 3 cose:

- impedisce un'interruzione di pagina all'interno di qualsiasi tag p, il che significa che un paragrafo non verrà mai spezzato in due pagine, se possibile.
- forza una page-break-before in tutte le intestazioni h1, il che significa che prima di ogni occorrenza h1, ci sarà un'interruzione di pagina.
- impedisce le interruzioni di pagina subito dopo ogni h2

Leggi [La frammentazione online](https://riptutorial.com/it/css/topic/4316/la-frammentazione): <https://riptutorial.com/it/css/topic/4316/la-frammentazione>

---

# Capitolo 32: Layout di blocco in linea

## Examples

### Barra di navigazione giustificata

La barra di navigazione (menu) orizzontalmente giustificata presenta un numero di elementi che dovrebbe essere giustificato. La prima voce (a sinistra) non ha margine sinistro all'interno del contenitore, l'ultima (a destra) elemento non ha margine destro all'interno del contenitore. La distanza tra gli articoli è uguale, indipendente dalla larghezza dell'oggetto singolo.

---

## HTML

```
<nav>
  <ul>
    <li>abc</li>
    <li>abcdefghijkl</li>
    <li>abcdef</li>
  </ul>
</nav>
```

---

## CSS

```
nav {
  width: 100%;
  line-height: 1.4em;
}
ul {
  list-style: none;
  display: block;
  width: 100%;
  margin: 0;
  padding: 0;
  text-align: justify;
  margin-bottom: -1.4em;
}
ul:after {
  content: "";
  display: inline-block;
  width: 100%;
}
li {
  display: inline-block;
}
```

---

## Gli appunti

- I tag `nav`, `ul` e `li` sono stati scelti per il loro significato semantico di 'una lista di voci di navigazione (menu)'. Naturalmente è possibile utilizzare anche altri tag.
- Lo pseudo-elemento `:after` causa una "linea" in più nella `ul` e quindi un'altezza extra vuota di questo blocco, spingendo verso il basso altri contenuti. Questo è risolto dal `margin-bottom` negativo `margin-bottom`, che deve avere la stessa grandezza `line-height` della `line-height` (ma negativo).
- Se la pagina diventa troppo stretta per tutti gli elementi da adattare, gli elementi si romperanno su una nuova riga (partendo da destra) e saranno giustificati su questa linea. L'altezza totale del menu aumenterà in base alle esigenze.

Leggi **Layout di blocco in linea online**: <https://riptutorial.com/it/css/topic/3308/layout-di-blocco-in-linea>

---

# Capitolo 33: Layout flessibile della scatola (Flexbox)

## introduzione

Il modulo Flexible Box, o semplicemente "flexbox" in breve, è un modello box progettato per interfacce utente e consente agli utenti di allineare e distribuire lo spazio tra gli elementi in un contenitore in modo tale che gli elementi si comportino in modo prevedibile quando il layout della pagina deve adattarsi a diversi, sconosciuti dimensioni dello schermo. Un contenitore flessibile espande gli oggetti per riempire lo spazio disponibile e li riduce per evitare il trabocco.

## Sintassi

- `display: flex;`
- direzione della flessione: `riga` | `fila-indietro` | `colonna` | `colonna-retromarcia`;
- `flex-wrap`: `nowrap` | `avvolgere` | `wrap-retromarcia`;
- `flex-flow`: `<'flex-direction'> || <'Flex-wrap'>`
- `justify-content`: `flex-start` | `flex-end` | `centro` | `spazio-tra` | `spazio intorno`;
- `align-items`: `flex-start` | `flex-end` | `centro` | `linea di base` | `allungare`;
- `allinea-contenuto`: `flex-start` | `flex-end` | `centro` | `spazio-tra` | `spazio-in giro` | `allungare`;
- `ordine`: `<intero>`;
- `flex-grow`: `<numero>`; / \* default 0 \* /
- `flex-shrink`: `<numero>`; / \* predefinito 1 \* /
- `flex-base`: `<lunghezza>` | `auto`; / \* predefinito auto \* /
- `flex`: `nessuno` | [`<'flex-grow'>` `<'flex-shrink'>`?] || `<'flex-base'>`]
- `align-self`: `auto` | `flex-start` | `flex-end` | `centro` | `linea di base` | `allungare`;

## Osservazioni

---

## Prefissi di Vender

- `mostra`: `-webkit-box`; / \* Chrome <20 \* /
- `display`: `-webkit-flex`; / \* Chrome 20+ \* /
- `display`: `-moz-box`; / \* Firefox \* /
- `display`: `-ms-flexbox`; / \* IE \* /
- `display`: `flex`; / \* Browser moderni \* /

---

## risorse

- [Una guida completa per Flexbox](#)
- [Risolto da Flexbox](#)

- [Che cos'è il Flexbox ?!](#)
- [Flexbox in 5 minuti](#)
- [Flexbugs](#)

## Examples

### Piè di pagina appiccicoso a altezza variabile

Questo codice crea un piè di pagina appiccicoso. Quando il contenuto non raggiunge la fine della finestra, il piè di pagina rimane nella parte inferiore della finestra. Quando il contenuto si estende oltre la parte inferiore della finestra, anche il piè di pagina viene espulso dal viewport. [Visualizza risultato](#)

HTML:

```
<div class="header">
  <h2>Header</h2>
</div>

<div class="content">
  <h1>Content</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur sodales ligula in libero. </p>
</div>

<div class="footer">
  <h4>Footer</h4>
</div>
```

CSS:

```
html, body {
  height: 100%;
}

body {
  display: flex;
  flex-direction: column;
}

.content {
  /* Include `0 auto` for best browser compatibility. */
  flex: 1 0 auto;
}

.header, .footer {
  background-color: grey;
  color: white;
  flex: none;
}
```

## Layout di Holy Grail usando Flexbox

Il layout di Holy Grail è un layout con intestazione e piè di pagina ad altezza fissa e un centro con 3 colonne. Le 3 colonne includono un sidenav a larghezza fissa, un centro fluido e una colonna per altri contenuti come gli annunci (il centro del fluido appare per primo nel markup). CSS Flexbox può essere utilizzato per ottenere questo risultato con un markup molto semplice:

### Markup HTML:

```
<div class="container">
  <header class="header">Header</header>
  <div class="content-body">
    <main class="content">Content</main>
    <nav class="sidenav">Nav</nav>
    <aside class="ads">Ads</aside>
  </div>
  <footer class="footer">Footer</footer>
</div>
```

### CSS:

```
body {
  margin: 0;
  padding: 0;
}

.container {
  display: flex;
  flex-direction: column;
  height: 100vh;
}

.header {
  flex: 0 0 50px;
}

.content-body {
  flex: 1 1 auto;

  display: flex;
  flex-direction: row;
}

.content-body .content {
  flex: 1 1 auto;
  overflow: auto;
}

.content-body .sidenav {
  order: -1;
  flex: 0 0 100px;
  overflow: auto;
}

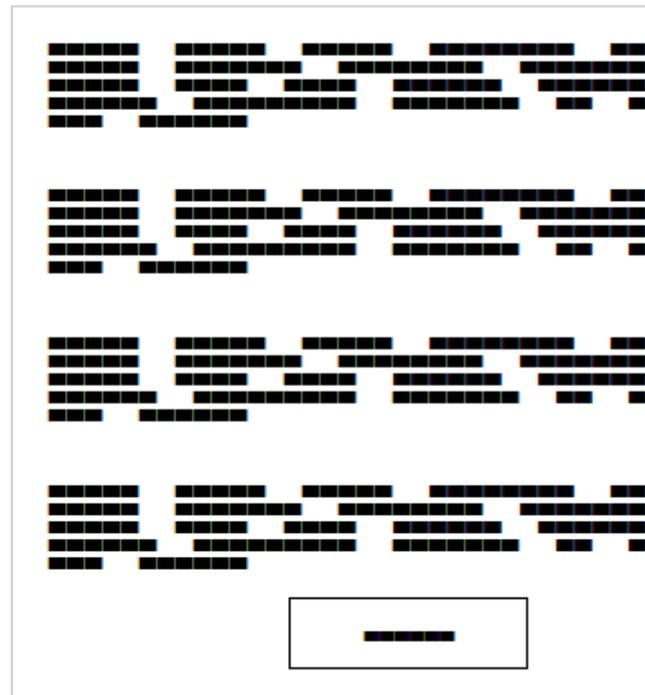
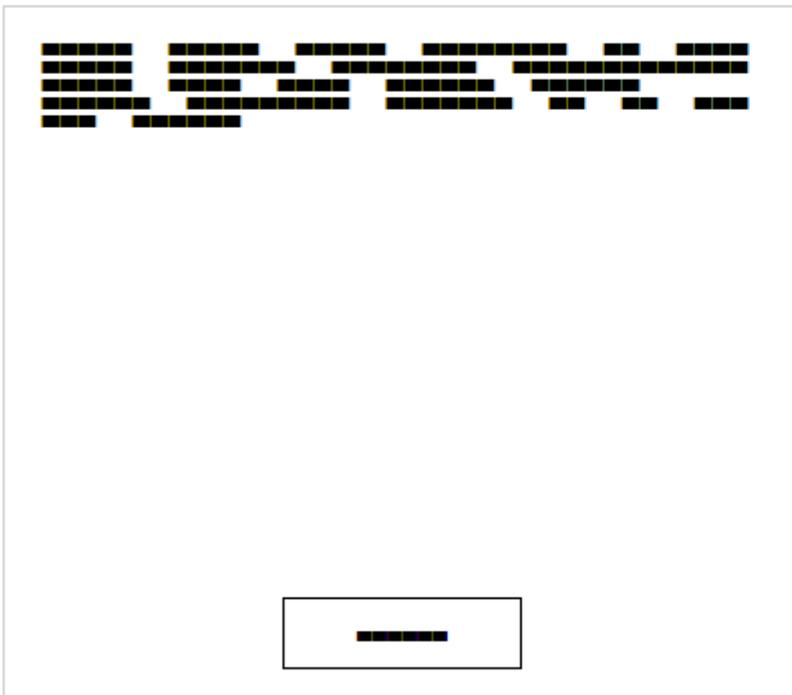
.content-body .ads {
  flex: 0 0 100px;
  overflow: auto;
}
```

```
}  
  
.footer {  
  flex: 0 0 50px;  
}
```

## dimostrazione

### Pulsanti perfettamente allineati all'interno di schede con flexbox

In questi giorni è un modello regolare in design per allineare verticalmente l' **invito alle azioni** all'interno delle sue carte contenenti come questo:



Questo può essere ottenuto utilizzando un trucco speciale con `flexbox`

## HTML

```
<div class="cards">  
  <div class="card">  
    <p>Lorem ipsum Magna proident ex anim dolor ullamco pariatur reprehenderit culpa esse enim  
mollit labore dolore voluptate ullamco et ut sed qui minim.</p>  
    <p><button>Action</button></p>  
  </div>  
  <div class="card">  
    <p>Lorem ipsum Magna proident ex anim dolor ullamco pariatur reprehenderit culpa esse enim  
mollit labore dolore voluptate ullamco et ut sed qui minim.</p>  
    <p>Lorem ipsum Magna proident ex anim dolor ullamco pariatur reprehenderit culpa esse enim  
mollit labore dolore voluptate ullamco et ut sed qui minim.</p>  
    <p>Lorem ipsum Magna proident ex anim dolor ullamco pariatur reprehenderit culpa esse enim  
mollit labore dolore voluptate ullamco et ut sed qui minim.</p>  
    <p>Lorem ipsum Magna proident ex anim dolor ullamco pariatur reprehenderit culpa esse enim  
mollit labore dolore voluptate ullamco et ut sed qui minim.</p>  
    <p><button>Action</button></p>
```

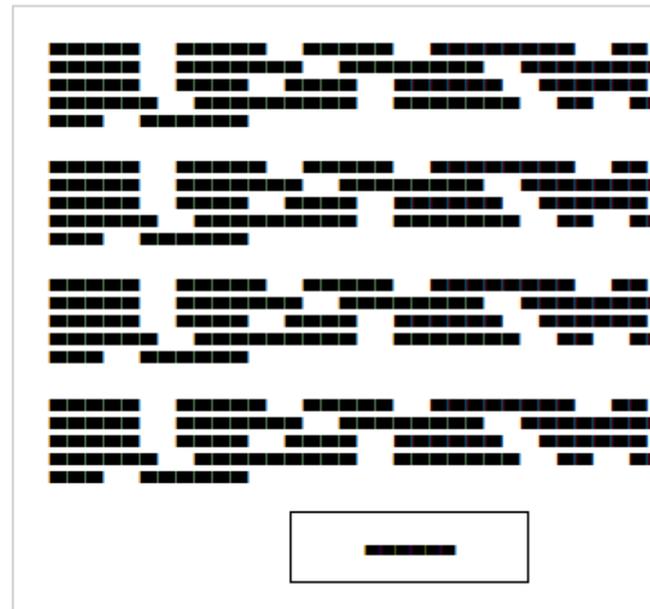
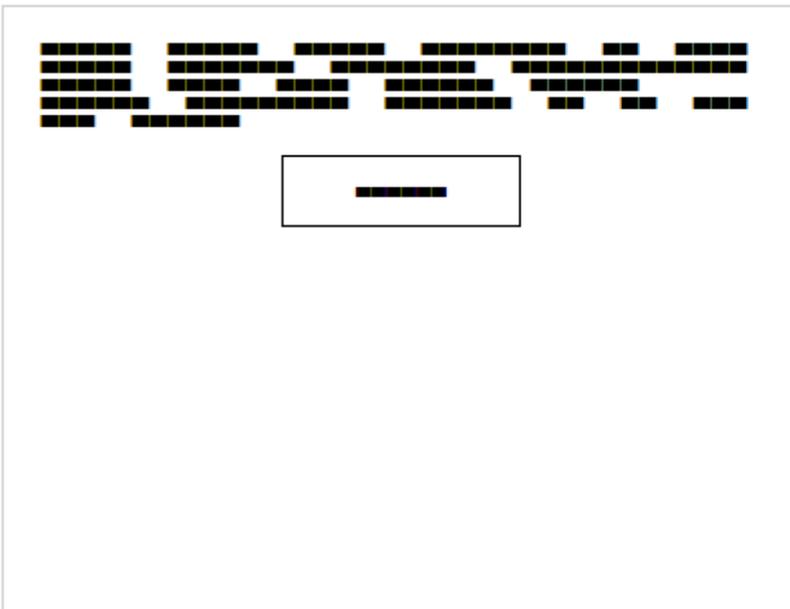
```
</div>
</div>
```

Prima di tutto, usiamo i CSS per applicare `display: flex;` al contenitore. Questo creerà 2 colonne uguali in altezza con il contenuto che scorre naturalmente al suo interno

## CSS

```
.cards {
  display: flex;
}
.card {
  border: 1px solid #ccc;
  margin: 10px 10px;
  padding: 0 20px;
}
button {
  height: 40px;
  background: #fff;
  padding: 0 40px;
  border: 1px solid #000;
}
p:last-child {
  text-align: center;
}
```

Il layout cambierà e diventerà così:



Per spostare i pulsanti nella parte inferiore del blocco, è necessario applicare la `display: flex;` alla carta stessa con la direzione impostata sulla `column`. Dopodiché, dovremmo selezionare l'ultimo elemento all'interno della carta e impostare il `margin-top` su `auto`. Questo spingerà l'ultimo paragrafo in fondo alla carta e raggiungerà il risultato richiesto.

CSS finale:

```
.cards {
  display: flex;
}
.card {
  border: 1px solid #ccc;
  margin: 10px 10px;
  padding: 0 20px;
  display: flex;
  flex-direction: column;
}
button {
  height: 40px;
  background: #fff;
  padding: 0 40px;
  border: 1px solid #000;
}
p:last-child {
  text-align: center;
  margin-top: auto;
}
```

Centrata verticale e orizzontale dinamica (allineare elementi, giustificare contenuto)

## Esempio semplice (centrata di un singolo elemento)

### HTML

```
<div class="aligner">
  <div class="aligner-item">...</div>
</div>
```

### CSS

```
.aligner {
  display: flex;
  align-items: center;
  justify-content: center;
}

.aligner-item {
  max-width: 50%; /*for demo. Use actual width instead.*/
}
```

Ecco una [demo](#) .

# Ragionamento

| Proprietà                    | Valore              | Descrizione   |
|------------------------------|---------------------|---|
| <code>align-items</code>     | <code>center</code> | Questo centra gli elementi lungo l'asse diverso da quello specificato dalla <code>flex-direction</code> , cioè il centraggio verticale per un flexbox orizzontale e il centraggio orizzontale per un flexbox verticale.   |
| <code>justify-content</code> | <code>center</code> | Questo centra gli elementi lungo l'asse specificato dalla <code>flex-direction</code> . Vale a dire, per un flexbox orizzontale ( <code>flex-direction: row</code> ), questo si concentra orizzontalmente, e per un flexbox verticale ( <code>flex-direction: column</code> ), questo centra verticalmente) |

## Esempi di proprietà individuali

Tutti gli stili seguenti sono applicati a questo layout semplice:

```
<div id="container">
  <div></div>
  <div></div>
  <div></div>
</div>
```

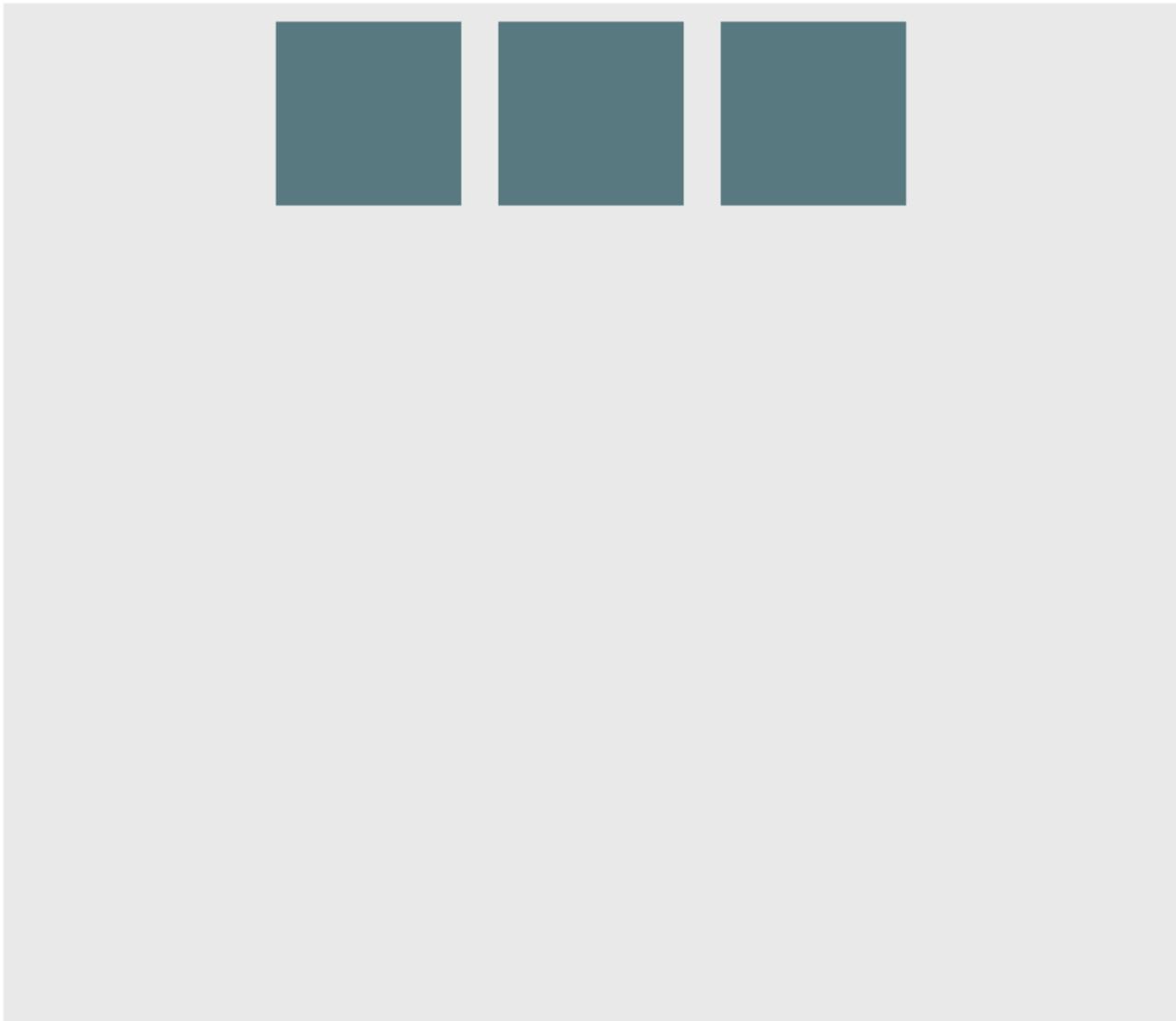
dove `#container` è il `flex-box`.

### Esempio: `justify-content: center` su un flexbox orizzontale

CSS:

```
div#container {
  display: flex;
  flex-direction: row;
  justify-content: center;
}
```

Risultato:



Ecco una [demo](#) .

## Esempio: `justify-content: center` su un flexbox verticale

### CSS:

```
div#container {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
}
```

### Risultato:



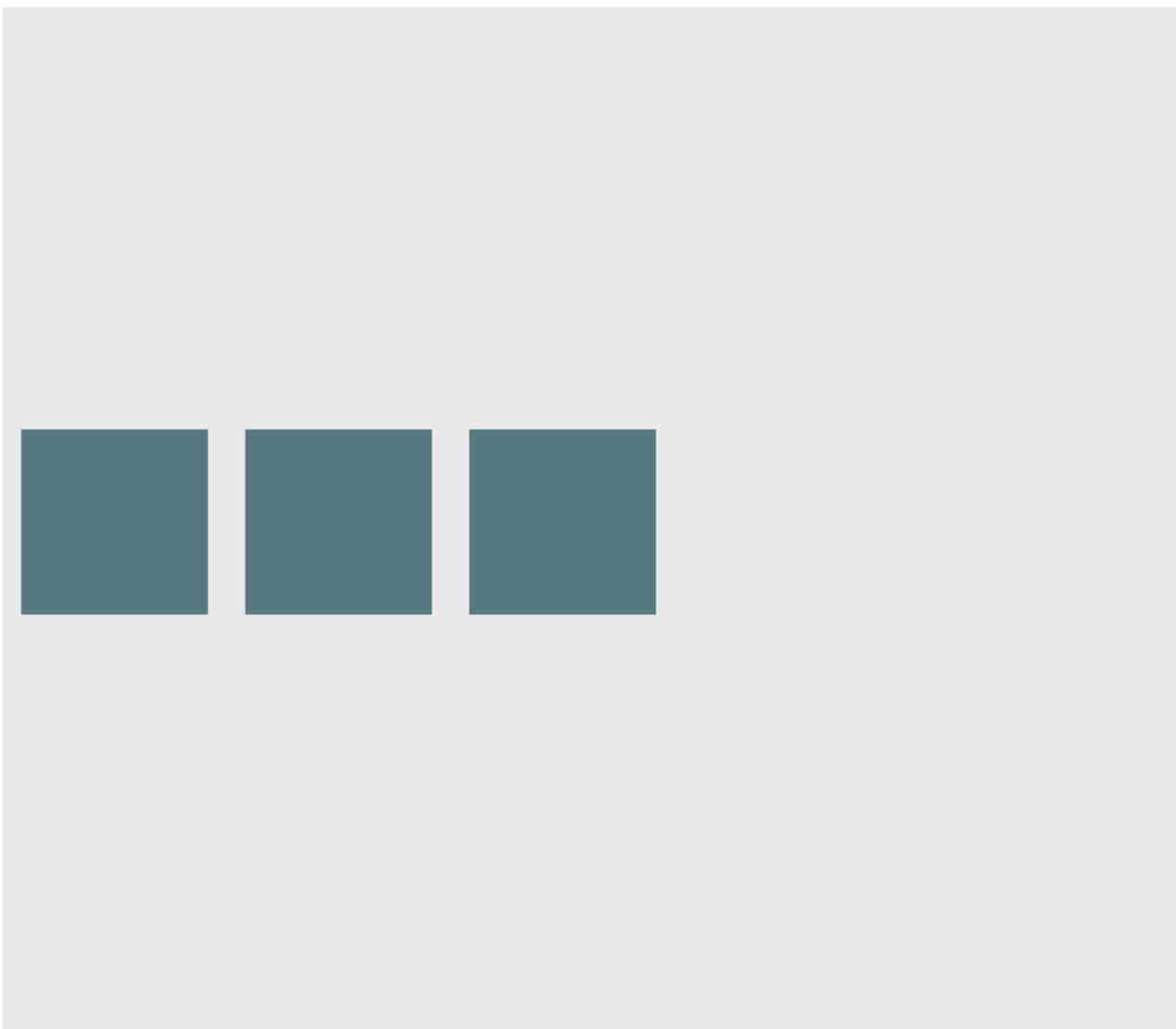
Ecco una [demo](#) .

## Esempio: `align-content: center` su una flexbox orizzontale

### CSS:

```
div#container {  
  display: flex;  
  flex-direction: row;  
  align-items: center;  
}
```

### Risultato:



Ecco una [demo](#) .

## Esempio: `align-content: center` su una flexbox verticale

### CSS:

```
div#container {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
}
```

### Risultato:



Ecco una [demo](#) .

## Esempio: combinazione per il centraggio sia sulla flexbox orizzontale

```
div#container {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
  align-items: center;  
}
```

**Risultato:**



Ecco una [demo](#) .

## Esempio: combinazione per il centraggio sia sulla flexbox verticale

```
div#container {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: center;  
}
```

**Risultato:**



Ecco una [demo](#) .

## Stessa altezza sui contenitori nidificati

Questo codice garantisce che tutti i contenitori annidati abbiano sempre la stessa altezza. Questo viene fatto assicurando che tutti gli elementi annidati abbiano la stessa altezza del div di contenimento del contenitore. [Guarda l'esempio di lavoro](https://jsfiddle.net/3wwh7ewp/) : <https://jsfiddle.net/3wwh7ewp/>

Questo effetto si ottiene a causa del fatto che gli `align-items` proprietà vengono impostati per `stretch` per impostazione predefinita.

## HTML

```
<div class="container">
  <div style="background-color: red">
    Some <br />
    data <br />
    to make<br />
    a height <br />
  </div>
```

```
<div style="background-color: blue">
  Fewer <br />
  lines <br />
</div>
</div>
```

## CSS

```
.container {
  display: flex;
  align-items: stretch; // Default value
}
```

Nota: [non funziona su versioni IE inferiori a 10](#)

## Adatta in modo ottimale gli elementi al loro contenitore

Una delle più belle funzionalità di Flexbox è quella di consentire l'adattamento ottimale dei contenitori all'elemento principale.

[Demo dal vivo](#)

HTML:

```
<div class="flex-container">
  <div class="flex-item">1</div>
  <div class="flex-item">2</div>
  <div class="flex-item">3</div>
  <div class="flex-item">4</div>
  <div class="flex-item">5</div>
</div>
```

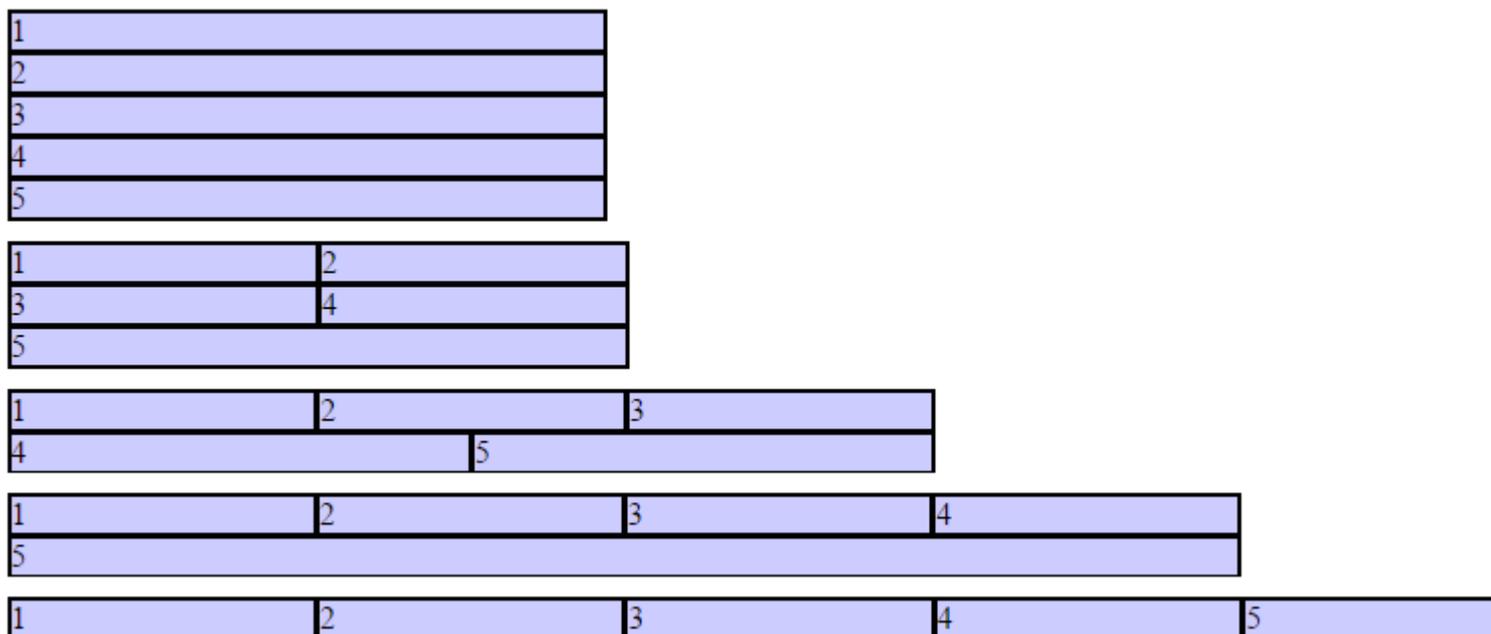
CSS:

```
.flex-container {
  background-color: #000;
  height: 100%;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  justify-content: flex-start;
  align-content: stretch;
  align-items: stretch;
}

.flex-item {
  background-color: #ccf;
  margin: 0.1em;
  flex-grow: 1;
  flex-shrink: 0;
  flex-basis: 200px; /* or % could be used to ensure a specific layout */
}
```

**Risultato:**

Le colonne si adattano quando lo schermo viene ridimensionato.



Leggi Layout flessibile della scatola (Flexbox) online: <https://riptutorial.com/it/css/topic/445/layout-flessibile-della-scatola--flexbox->

# Capitolo 34: Lineamenti

## Sintassi

- `outline`: `outline-color outline-style outline-width` | iniziale | ereditare;
- larghezza del contorno: `medio` | `sottile` | `spesso` | `lunghezza` | iniziale | ereditare;
- stile schema: `nessuno` | `nascosto` | `punteggiato` | `tratteggiata` | `solido` | `doppio` | `groove` | `cresta` | `inserto` | `inizio` | iniziale | ereditare;

## Parametri

| Parametro                 | Dettagli  |
|---------------------------|---|
| <code>tratteggiata</code> | contorno punteggiato  |
| <code>tratteggiata</code> | contorno tratteggiato   |
| <code>solido</code>       | contorno solido   |
| <code>Doppio</code>       | doppio schema   |
| <code>scanalatura</code>  | Il contorno scanalato 3D dipende dal valore del contorno del colore     |
| <code>cresta</code>       | Il contorno tridimensionale, dipende dal valore del contorno del colore |
| <code>inserire</code>     | Il contorno dell'inserto 3D dipende dal valore del contorno del colore  |
| <code>inizio</code>       | Il contorno 3D, dipende dal valore del contorno del colore              |
| <code>nessuna</code>      | senza contorni  |
| <code>nascosto</code>     | contorno nascosto   |

## Osservazioni

`outline` è ora descritto nell'interfaccia [utente di base](#), un modulo CSS Livello 3 (era già descritto in REC CSS2.1)

La proprietà struttura è definita per impostazione predefinita nei browser per gli elementi attivabili in `:focus` stato di attivazione.

Non dovrebbe essere rimosso, vedi <http://outlinenone.com> che afferma:

### Cosa fa la struttura del profilo?

Fornisce feedback visivo per i collegamenti che hanno "focus" durante la navigazione

di un documento Web utilizzando il tasto TAB (o equivalente). Ciò è particolarmente utile per le persone che non possono usare un mouse o avere un danno visivo. Se rimuovi la struttura, stai rendendo il tuo sito inaccessibile per queste persone. (...)

Esempi correlati interessanti su Stack Overflow:

- [Come rimuovere l'evidenziazione del bordo su un elemento di testo di input](#)
- [Come rimuovere il contorno tratteggiato di Firefox su BOTTONI e collegamenti?](#)

## Examples

### Panoramica

Struttura è una linea che circonda l'elemento, al di fuori del confine. A differenza del `border`, i contorni non occupano spazio nel modello di riquadro. Quindi aggiungere una struttura a un elemento non influisce sulla posizione dell'elemento o di altri elementi.

Inoltre, i contorni possono essere non rettangolari in alcuni browser. Questo può accadere se `outline` viene applicato su un elemento `span` che contiene testo con proprietà di `font-size` diverse al suo interno. A differenza dei bordi, i contorni *non possono avere angoli arrotondati*.

Le parti essenziali del `outline` sono `outline-color` `outline-style` `outline-width`.

La definizione di una struttura è equivalente alla definizione di una frontiera:

Un contorno è una linea attorno a un elemento. Viene visualizzato attorno al margine dell'elemento. Tuttavia, è diverso dalla proprietà del confine.

```
outline: 1px solid black;
```

### outline-style

La proprietà `outline-style` viene utilizzata per impostare lo stile del contorno di un elemento.

```
p {
  border: 1px solid black;
  outline-color:blue;
  line-height:30px;
}
.p1{
  outline-style: dotted;
}
.p2{
  outline-style: dashed;
}
.p3{
  outline-style: solid;
}
.p4{
  outline-style: double;
}
```

```
.p5{
  outline-style: groove;
}
.p6{
  outline-style: ridge;
}
.p7{
  outline-style: inset;
}
.p8{
  outline-style: outset;
}
```

## HTML

```
<p class="p1">A dotted outline</p>
<p class="p2">A dashed outline</p>
<p class="p3">A solid outline</p>
<p class="p4">A double outline</p>
<p class="p5">A groove outline</p>
<p class="p6">A ridge outline</p>
<p class="p7">An inset outline</p>
<p class="p8">An outset outline</p>
```

A dotted outline

A dashed outline

A solid outline

A double outline

A groove outline

A ridge outline

An inset outline

An outset outline

Leggi Lineamenti online: <https://riptutorial.com/it/css/topic/4258/lineamenti>

---

# Capitolo 35: margini

## Sintassi

- margine: `<top & right & bottom & left>` ;
- margine: `<top>` , `<left & right>` , `<bottom>` ;
- margine: `<alto e basso>` , `<sinistra e destra>` ;
- margine: `<top>` , `<right>` , `<bottom>` , `<left>` ;
- margin-top: `<top>` ;
- margin-right: `<right>` ;
- margin-bottom: `<bottom>` ;
- margin-left: `<left>` ;

## Parametri

| Parametro      | Dettagli   |
|----------------|--|
| 0              | imposta il margine su nessuno  |
| auto           | usato per centrare, impostando uniformemente i valori su ciascun lato            |
| unità (es. px) | vedi la sezione parametri in <a href="#">Unità</a> per un elenco di unità valide |
| ereditare      | eredita il valore del margine dall'elemento genitore                             |
| iniziale       | ripristinare al valore iniziale  |

## Osservazioni

Altro su "Margini collassanti": [qui](#) .

## Examples

Applica margine su un dato lato

---

## Proprietà specifiche della direzione

CSS ti consente di specificare un lato specifico per applicare il margine a. Le quattro proprietà fornite a questo scopo sono:

- margin-left
- margin-right
- margin-top

- `margin-bottom`

Il seguente codice applicherebbe un margine di 30 pixel sul lato sinistro del div selezionato.

[Visualizza risultato](#)

## HTML

```
<div id="myDiv"></div>
```

## CSS

```
#myDiv {  
  margin-left: 30px;  
  height: 40px;  
  width: 40px;  
  background-color: red;  
}
```

| Parametro                | Dettagli  |
|--------------------------|---|
| <code>margin-left</code> | La direzione in cui il margine dovrebbe essere applicato. |
| <code>30px</code>        | La larghezza del margine.                                 |

# Specifica della direzione utilizzando la proprietà di stenografia

La proprietà del `margin` standard può essere espansa per specificare diverse larghezze su ciascun lato degli elementi selezionati. La sintassi per fare ciò è la seguente:

```
margin: <top> <right> <bottom> <left>;
```

L'esempio seguente applica un margine di larghezza zero all'inizio del div, un margine di 10 pixel sul lato destro, un margine di 50 pixel sul lato sinistro e un margine di 100 pixel sul lato sinistro.

[Visualizza risultato](#)

## HTML

```
<div id="myDiv"></div>
```

## CSS

```
#myDiv {  
  margin: 0 10px 50px 100px;  
  height: 40px;  
  width: 40px;  
  background-color: red;  
}
```

```
}
```

## Margine che crolla

Quando due margini si toccano l'un l'altro verticalmente, vengono compressi. Quando due margini toccano orizzontalmente, non collassano.

### Esempio di margini verticali adiacenti:

Considera i seguenti stili e markup:

```
div{
  margin: 10px;
}
```

```
<div>
  some content
</div>
<div>
  some more content
</div>
```

Saranno 10px a parte poiché i margini verticali crollano sopra l'uno e l'altro. (La spaziatura non sarà la somma di due margini).

### Esempio di margini orizzontali adiacenti:

Considera i seguenti stili e markup:

```
span{
  margin: 10px;
}
```

```
<span>some</span><span>content</span>
```

Saranno 20px a parte poiché i margini orizzontali non collassano sull'uno e sull'altro. (La spaziatura sarà la somma di due margini).

## Sovrapposizione con dimensioni diverse

```
.top{
  margin: 10px;
}
.bottom{
  margin: 15px;
}
```

```
<div class="top">
  some content
</div>
<div class="bottom">
```

```
    some more content
</div>
```

Questi elementi saranno distanziati di 15px verticalmente. I margini si sovrappongono il più possibile, ma il margine più ampio determinerà la spaziatura tra gli elementi.

### Gotcha margine di sovrapposizione

```
.outer-top{
  margin: 10px;
}
.inner-top{
  margin: 15px;
}
.outer-bottom{
  margin: 20px;
}
.inner-bottom{
  margin: 25px;
}
```

```
<div class="outer-top">
  <div class="inner-top">
    some content
  </div>
</div>
<div class="outer-bottom">
  <div class="inner-bottom">
    some more content
  </div>
</div>
```

Quale sarà la spaziatura tra i due testi? (passa il mouse per vedere la risposta)

La spaziatura sarà 25px. Poiché tutti e quattro i margini si toccano, collassano, utilizzando quindi il margine maggiore dei quattro.

Ora, cosa succede se aggiungiamo alcuni bordi al markup sopra.

```
div{
  border: 1px solid red;
}
```

Quale sarà la spaziatura tra i due testi? (passa il mouse per vedere la risposta)

La spaziatura sarà di 59 px! Ora solo i margini di .outer-top e .outer-bottom si toccano e sono gli unici margini compressi. I margini rimanenti sono separati dai bordi. Quindi abbiamo 1px + 10px + 1px + 45px + 20px + 1px + 25px + 1px. (I 1px sono i confini ...)

### Margini collassabili tra elementi padre e figlio:

HTML:

```
<h1>Title</h1>
<div>
  <p>Paragraph</p>
</div>
```

## CSS

```
h1 {
  margin: 0;
  background: #cff;
}
div {
  margin: 50px 0 0 0;
  background: #cfc;
}
p {
  margin: 25px 0 0 0;
  background: #cf9;
}
```

Nell'esempio sopra riportato, si applica solo il margine più ampio. Ci si può aspettare che il paragrafo si trovi a 60px da h1 (poiché l'elemento div ha un margine superiore di 40px e il p ha un margine superiore di 20px). Ciò non accade perché i margini collasano insieme per formare un margine.

## Centra orizzontalmente gli elementi su una pagina usando il margine

Finché l'elemento è un **blocco** e ha un **valore di larghezza impostato in modo esplicito**, è possibile utilizzare i margini per centrare gli elementi di blocco su una pagina in senso orizzontale.

Aggiungiamo un valore di larghezza inferiore alla larghezza della finestra e la proprietà automatica del margine quindi distribuisce lo spazio rimanente a sinistra ea destra:

```
#myDiv {
  width:80%;
  margin:0 auto;
}
```

Nell'esempio sopra, utilizziamo la dichiarazione del `margin` stenografico per impostare prima `0` sui valori di margine superiore e inferiore (sebbene questo possa essere qualsiasi valore) e quindi usiamo `auto` per consentire al browser di allocare automaticamente lo spazio ai valori del margine sinistro e destro.

Nell'esempio sopra, l'elemento `#myDiv` è impostato su una larghezza dell'80% che lascia il rimanente del 20%. Il browser distribuisce questo valore ai lati rimanenti in modo tale:

$$(100\% - 80\%) / 2 = 10\%$$

## Semplificazione della proprietà del margine

```
p {
```

```

margin:1px;                /* 1px margin in all directions */

/*equals to:*/

margin:1px 1px;

/*equals to:*/

margin:1px 1px 1px;

/*equals to:*/

margin:1px 1px 1px 1px;
}

```

## Un altro esempio:

```

p{
margin:10px 15px;          /* 10px margin-top & bottom And 15px margin-right & left*/

/*equals to:*/

margin:10px 15px 10px 15px;

/*equals to:*/

margin:10px 15px 10px;
/* margin left will be calculated from the margin right value (=15px) */
}

```

## Margini negativi

Margine è una delle poche proprietà CSS che può essere impostata su valori negativi. Questa proprietà può essere utilizzata per **sovrapporre elementi senza posizionamento assoluto** .

```

div{
display: inline;
}

#over{
margin-left: -20px;
}

<div>Base div</div>
<div id="over">Overlapping div</div>

```

## Esempio 1:

È ovvio supporre che il valore percentuale del margine a `margin-left` e `margin-right` sia relativo al suo elemento genitore.

```

.parent {
width : 500px;
height: 300px;
}

```

```
.child {
  width : 100px;
  height: 100px;
  margin-left: 10%; /* (parentWidth * 10/100) => 50px */
}
```

Ma non è questo il caso, quando si tratta di `margin-top` e `margin-bottom`. Entrambe queste proprietà, in percentuale, non sono relative all'altezza del contenitore genitore ma alla **larghezza** del contenitore genitore.

Così,

```
.parent {
  width : 500px;
  height: 300px;
}

.child {
  width : 100px;
  height: 100px;
  margin-left: 10%; /* (parentWidth * 10/100) => 50px */
  margin-top: 20%; /* (parentWidth * 20/100) => 100px */
}
```

Leggi margini online: <https://riptutorial.com/it/css/topic/305/margini>

# Capitolo 36: Media Queries

## Sintassi

- `@media [not | only] mediatype e (media feature) {/ * Regole CSS da applicare * /}`

## Parametri

| Parametro                   | Dettagli  |
|-----------------------------|---|
| <code>mediatype</code>      | (Opzionale) Questo è il tipo di media. Potrebbe essere qualsiasi cosa nell'intervallo di <code>all</code> per lo <code>screen</code> .          |
| <code>not</code>            | (Opzionale) Non applica il CSS per questo particolare tipo di media e si applica a tutto il resto.  |
| <code>media feature</code>  | Logica per identificare casi d'uso per CSS. Opzioni descritte di seguito.   |
| Caratteristica multimediale | Dettagli  |
| <code>aspect-ratio</code>   | Descrive le proporzioni dell'area di visualizzazione target del dispositivo di output.  |
| <code>color</code>          | Indica il numero di bit per componente colore del dispositivo di output. Se il dispositivo non è un dispositivo a colori, questo valore è zero. |
| <code>color-index</code>    | Indica il numero di voci nella tabella di ricerca colore per il dispositivo di output.  |
| <code>grid</code>           | Determina se il dispositivo di output è un dispositivo a griglia o un dispositivo bitmap.   |
| <code>height</code>         | La funzione media dell'altezza descrive l'altezza della superficie di rendering del dispositivo di output.                                      |
| <code>max-width</code>      | Il CSS non si applica su una larghezza dello schermo più ampia di quella specificata.   |
| <code>min-width</code>      | Il CSS non si applica su una larghezza dello schermo più stretta di quella specificata.   |
| <code>max-height</code>     | Il CSS non si applica su un'altezza dello schermo più alta di quella specificata.   |
| <code>min-height</code>     | Il CSS non si applica su un'altezza dello schermo inferiore a quella specificata.   |

| Parametro              | Dettagli   |
|------------------------|--|
| monochrome             | Indica il numero di bit per pixel su un dispositivo monocromatico (in scala di grigi).   |
| orientation            | I CSS verranno visualizzati solo se il dispositivo utilizza l'orientamento specificato. Vedi le osservazioni per maggiori dettagli.  |
| resolution             | Indica la risoluzione (densità di pixel) del dispositivo di output.  |
| scan                   | Descrive il processo di scansione dei dispositivi di output televisivi.  |
| width                  | La funzionalità di larghezza del supporto descrive la larghezza della superficie di rendering del dispositivo di output (ad esempio la larghezza della finestra del documento o la larghezza del riquadro di pagina su una stampante).               |
| Funzionalità deprecate | Dettagli   |
| device-aspect-ratio    | CSS <code>device-aspect-ratio</code> verrà visualizzato solo sui dispositivi il cui rapporto altezza / larghezza corrisponde al rapporto specificato. Questa è una funzionalità <code>device-aspect-ratio</code> e non è garantito il funzionamento. |
| max-device-width       | <code>max-device-width</code> uguale alla <code>max-width</code> ma misura la larghezza fisica dello schermo, piuttosto che la larghezza di visualizzazione del browser.   |
| min-device-width       | <code>min-device-width</code> Uguale a <code>min-width</code> ma misura la larghezza fisica dello schermo, piuttosto che la larghezza di visualizzazione del browser.  |
| max-device-height      | <code>max-device-height</code> uguale a <code>max-height</code> ma misura la larghezza fisica dello schermo, piuttosto che la larghezza di visualizzazione del browser.  |
| min-device-height      | <code>min-device-height</code> uguale a <code>min-height</code> ma misura la larghezza fisica dello schermo, piuttosto che la larghezza di visualizzazione del browser.  |

## Osservazioni

Le query multimediali sono supportate in tutti i browser moderni, tra cui Chrome, Firefox, Opera e Internet Explorer 9 e versioni successive.

È importante notare che la funzione dei supporti di `orientation` non è limitata ai dispositivi mobili. Si basa sulla larghezza e altezza del viewport (non sulla finestra o sui dispositivi).

*La modalità Paesaggio* è quando la larghezza della finestra è più grande dell'altezza della finestra.

*La modalità verticale* è quando l'altezza del viewport è più grande della larghezza della finestra.

Questo di solito si traduce in un monitor desktop in modalità orizzontale, ma a volte può essere ritratto.

---

Nella maggior parte dei casi, i dispositivi mobili segnalano la loro risoluzione e non la loro dimensione reale dei pixel che può differire a causa della densità dei pixel. Per costringerli a segnalare la loro reale dimensione dei pixel aggiungi quanto segue all'interno del tuo `head` tag:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

---

## Examples

### Esempio di base

```
@media screen and (min-width: 720px) {  
  body {  
    background-color: skyblue;  
  }  
}
```

La query multimediale precedente specifica due condizioni:

1. La pagina deve essere visualizzata su uno schermo normale (non una pagina stampata, un proiettore, ecc.).
2. La larghezza della porta di visualizzazione dell'utente deve essere di almeno 720 pixel.

Se queste condizioni sono soddisfatte, gli stili all'interno della query multimediale saranno attivi e il colore di sfondo della pagina sarà blu cielo.

Le query multimediali vengono applicate in modo dinamico. Se sulla pagina di caricamento vengono soddisfatte le condizioni specificate nella query multimediale, il CSS verrà applicato, ma verrà immediatamente disabilitato nel caso in cui le condizioni cessino di essere soddisfatte. Al contrario, se le condizioni inizialmente non sono soddisfatte, il CSS non sarà applicato fino a quando non saranno soddisfatte le condizioni specificate.

Nel nostro esempio, se la larghezza della porta di visualizzazione dell'utente è inizialmente maggiore di 720 pixel, ma l'utente riduce la larghezza del browser, il colore di sfondo cesserà di essere blu cielo non appena l'utente ha ridimensionato la porta di visualizzazione a meno di 720 pixel in larghezza.

### Utilizzare sul tag link

```
<link rel="stylesheet" media="min-width: 600px" href="example.css" />
```

Questo foglio di stile è ancora scaricato ma viene applicato solo su dispositivi con una larghezza dello schermo superiore a 600 px.

## tipo di supporto

Le query multimediali hanno un parametro `mediatype` facoltativo. Questo parametro viene posizionato direttamente dopo la `@media` dichiarazione ( `@media mediatype` ), ad esempio:

```
@media print {
  html {
    background-color: white;
  }
}
```

Il precedente codice CSS darà all'elemento `HTML` DOM un colore di sfondo bianco durante la stampa.

Il parametro `mediatype` ha un prefisso `not` o `only` facoltativo che applicherà gli stili a tutto tranne il tipo di media specificato o solo il tipo di supporto specificato, rispettivamente. Ad esempio, il seguente esempio di codice applicherà lo stile a ogni tipo di supporto tranne la `print` .

```
@media not print {
  html {
    background-color: green;
  }
}
```

E allo stesso modo, per mostrarlo solo sullo schermo, questo può essere usato:

```
@media only screen {
  .fadeInEffects {
    display: block;
  }
}
```

L'elenco di `mediatype` può essere compreso meglio con la seguente tabella:

| Tipo di supporto        | Descrizione  |
|-------------------------|--|
| <code>all</code>        | Applica a tutti i dispositivi  |
| <code>screen</code>     | Computer predefiniti   |
| <code>print</code>      | Stampanti in generale. Utilizzato per lo stile delle versioni di stampa dei siti Web |
| <code>handheld</code>   | PDA, cellulari e dispositivi portatili con schermo piccolo                           |
| <code>projection</code> | Per presentazioni proiettate, ad esempio proiettori                                  |
| <code>aural</code>      | Sistemi vocali   |
| <code>braille</code>    | Dispositivi tattili braille  |

| Tipo di supporto | Descrizione   |
|------------------|---|
| embossed         | Stampanti braille paginate  |
| tv               | Dispositivi di tipo televisivo  |
| tty              | Dispositivi con una griglia di caratteri a passo fisso. Terminali, portatili. |

## Utilizzo di query multimediali per scegliere come target dimensioni schermo diverse

Spesso, il web design reattivo coinvolge le query multimediali, che sono blocchi CSS che vengono eseguiti solo se una condizione è soddisfatta. Questo è utile per il web design reattivo perché puoi utilizzare le query multimediali per specificare diversi stili CSS per la versione mobile del tuo sito web rispetto alla versione desktop.

```
@media only screen and (min-width: 300px) and (max-width: 767px) {
  .site-title {
    font-size: 80%;
  }

  /* Styles in this block are only applied if the screen size is atleast 300px wide, but no
  more than 767px */
}

@media only screen and (min-width: 768px) and (max-width: 1023px) {
  .site-title {
    font-size: 90%;
  }

  /* Styles in this block are only applied if the screen size is atleast 768px wide, but no
  more than 1023px */
}

@media only screen and (min-width: 1024px) {
  .site-title {
    font-size: 120%;
  }

  /* Styles in this block are only applied if the screen size is over 1024px wide. */
}
```

## Larghezza vs Viewport

Quando utilizziamo la "larghezza" con le query multimediali, è importante impostare correttamente il metatag. Il meta tag di base ha questo aspetto e deve essere inserito nel tag `<head>`.

```
<meta name="viewport" content="width=device-width,initial-scale=1">
```

### Perché questo è importante?

Sulla base della definizione di MDN "larghezza" è

La funzionalità di larghezza del supporto descrive la larghezza della superficie di rendering del dispositivo di output (ad esempio la larghezza della finestra del documento o la larghezza del riquadro di pagina su una stampante).

Cosa significa?

View-port è la larghezza del dispositivo stesso. Se la risoluzione dello schermo indica che la risoluzione è 1280 x 720, la larghezza della porta di visualizzazione è "1280px".

Più spesso molti dispositivi allocano una quantità di pixel diversa per visualizzare un pixel. Ad esempio iPhone 6 Plus ha una risoluzione di 1242 x 2208. Ma la larghezza della finestra e l'altezza della finestra sono 414 x 736 effettivi. Ciò significa che vengono utilizzati 3 pixel per creare 1 pixel.

Ma se non hai impostato correttamente il `meta`, tenderà di mostrare la tua pagina web con la sua risoluzione nativa che si traduce in una visualizzazione ingrandita (testi e immagini più piccoli).

## Media Queries per Retina e Non Retina Screens

Anche se questo funziona solo per i browser basati su WebKit, questo è utile:

```
/* ----- Non-Retina Screens ----- */
@media screen
  and (min-width: 1200px)
  and (max-width: 1600px)
  and (-webkit-min-device-pixel-ratio: 1) {
}

/* ----- Retina Screens ----- */
@media screen
  and (min-width: 1200px)
  and (max-width: 1600px)
  and (-webkit-min-device-pixel-ratio: 2)
  and (min-resolution: 192dpi) {
}
```

## Informazioni di base

Ci sono due tipi di pixel nel display. Uno è i pixel logici e l'altro è i pixel fisici. Principalmente, i pixel fisici rimangono sempre gli stessi, perché è lo stesso per tutti i dispositivi di visualizzazione. I pixel logici cambiano in base alla risoluzione dei dispositivi per visualizzare pixel di qualità superiore. Il rapporto pixel del dispositivo è il rapporto tra pixel fisici e pixel logici. Ad esempio, MacBook Pro Retina, iPhone 4 e versioni successive riportano un rapporto pixel del dispositivo pari a 2, in quanto la risoluzione lineare fisica è il doppio della risoluzione logica.

Il motivo per cui funziona solo con i browser basati su WebKit è dovuto a:

- Il prefisso del venditore `-webkit-` prima della regola.
- Questo non è stato implementato in motori diversi da WebKit e Blink.

## Terminologia e struttura

Le **query multimediali** consentono di applicare le regole CSS in base al tipo di dispositivo / supporto (ad es. Schermo, stampa o palmare) chiamato **tipo di supporto** , vengono descritti aspetti aggiuntivi del dispositivo con **funzioni multimediali** quali la disponibilità di dimensioni di colore o viewport.

---

## Struttura generale di una query multimediale

```
@media [...] {  
    /* One or more CSS rules to apply when the query is satisfied */  
}
```

---

## Una query multimediale contenente un tipo di media

```
@media print {  
    /* One or more CSS rules to apply when the query is satisfied */  
}
```

---

## Una query multimediale contenente un tipo di supporto e una funzione multimediale

```
@media screen and (max-width: 600px) {  
    /* One or more CSS rules to apply when the query is satisfied */  
}
```

---

## Una query multimediale contenente una funzione multimediale (e un tipo di media implicito di "tutti")

```
@media (orientation: portrait) {  
    /* One or more CSS rules to apply when the query is satisfied */  
}
```

### Media query e IE8

Le **query multimediali** non sono supportate affatto in IE8 e versioni precedenti.

## Una soluzione alternativa basata su Javascript

Per aggiungere il supporto per IE8, è possibile utilizzare una delle diverse soluzioni JS. Ad esempio, è possibile aggiungere **Rispondi** per aggiungere il supporto per le query multimediali per IE8 solo con il seguente codice:

```
<!--[if lt IE 9]>
<script
  src="respond.min.js">
</script>
<![endif]-->
```

**CSS Mediaquery** è un'altra libreria che fa la stessa cosa. Il codice per aggiungere quella libreria al tuo HTML sarebbe identico:

```
<!--[if lt IE 9]>
<script
  src="css3-mediaqueries.js">
</script>
<![endif]-->
```

---

## L'alternativa

Se non ti piace una soluzione basata su JS, dovresti anche considerare di aggiungere un foglio di stile IE <9 solo dove aggiusti lo stile specifico per IE <9. Per questo, dovresti aggiungere il seguente codice HTML al tuo codice:

```
<!--[if lt IE 9]>
<link rel="stylesheet" type="text/css" media="all" href="style-ielt9.css"/>
<![endif]-->
```

---

### Nota :

Tecnicamente è un'altra alternativa: usare gli **hack CSS** per scegliere IE <9. Ha lo stesso impatto di un foglio di stile IE <9, ma per questo non è necessario un foglio di stile separato. Non raccomando questa opzione, dato che producono codice CSS non valido (che è solo una delle numerose ragioni per cui l'uso degli hack CSS è generalmente disapprovato oggi).

Leggi Media Queries online: <https://riptutorial.com/it/css/topic/317/media-queries>

---

# Capitolo 37: Modelli di progettazione CSS

## introduzione

Questi esempi servono per documentare modelli di progettazione specifici per CSS come [BEM](#) , [OOCSS](#) e [SMACSS](#) .

Questi esempi NON sono per documentare framework CSS come [Bootstrap](#) o [Foundation](#) .

## Osservazioni

Questi esempi sono per la documentazione di metodologie / schemi di progettazione specifici per CSS.

Queste metodologie includono ma non sono esclusive di quanto segue:

- [BEM](#)
- [OOCSS](#)
- [SMACSS](#)

Questi esempi NON sono per documentare framework CSS come [Bootstrap](#) o [Foundation](#) . Sebbene si possano includere esempi su come applicare uno o più pattern / pattern di progettazione CSS con un framework CSS, questi esempi devono focalizzarsi sulle metodologie / schemi di progettazione con quel particolare framework e sull'uso del framework stesso.

## Examples

### BEM

[BEM](#) sta per `Blocks, Elements and Modifiers` . È una metodologia inizialmente concepita dalla società tecnologica russa [Yandex](#) , ma che ha ottenuto un certo successo tra gli sviluppatori web americani e dell'Europa occidentale.

Come lo stesso implica, la metodologia BEM si basa esclusivamente sulla componentizzazione del codice HTML e CSS in tre tipi di componenti:

- **Blocchi:** entità indipendenti che sono significative per conto proprio

Gli esempi sono `header` , `container` , `menu` , `checkbox` e `textbox`

- **Elementi:** parte di blocchi che non hanno un significato autonomo e sono legati semanticamente ai loro blocchi.

Esempi sono `menu item` , `menu item list item` , `checkbox caption` e `header title`

- **Modificatori:** contrassegni su un blocco o un elemento, utilizzati per modificare l'aspetto o il

comportamento

Gli esempi sono `disabled`, `highlighted`, `checked`, `fixed`, `size big` e `color yellow`

---

L'obiettivo di BEM è quello di ottimizzare la leggibilità, la manutenibilità e la flessibilità del codice CSS. Il modo per raggiungere questo obiettivo è applicare le seguenti regole.

- Gli stili di blocco non dipendono mai da altri elementi in una pagina
- I blocchi dovrebbero avere un nome breve e semplice ed evitare caratteri `_` o `-`
- Quando si disegnano elementi, utilizzare i selettori di formato `blockname__elementname`
- Quando `blockname--modifiername` stile, usa i selettori di formato `blockname--modifiername` e `blockname__elementname--modifiername`
- Elementi o blocchi che hanno modificatori dovrebbero ereditare tutto dal blocco o elemento che sta modificando eccetto le proprietà che il modificatore dovrebbe modificare

## Esempio di codice

Se applichi BEM ai tuoi elementi del modulo, i selettori CSS dovrebbero assomigliare a questo:

```
.form { } // Block
.form--theme-xmas { } // Block + modifier
.form--simple { } // Block + modifier
.form__input { } // Block > element
.form__submit { } // Block > element
.form__submit--disabled { } // Block > element + modifier
```

L'HTML corrispondente dovrebbe assomigliare a questo:

```
<form class="form form--theme-xmas form--simple">
  <input class="form__input" type="text" />
  <input class="form__submit form__submit--disabled" type="submit" />
</form>
```

Leggi Modelli di progettazione CSS online: <https://riptutorial.com/it/css/topic/10823/modelli-di-progettazione-css>

---

# Capitolo 38: Normalizzazione degli stili del browser

## introduzione

Ogni browser ha un set predefinito di stili CSS che utilizza per il rendering di elementi. Questi stili predefiniti potrebbero non essere coerenti tra i vari browser perché: le specifiche della lingua non sono chiare, quindi gli stili di base sono interpretabili, i browser potrebbero non seguire le specifiche fornite o i browser potrebbero non avere stili predefiniti per gli elementi HTML più recenti. Di conseguenza, le persone potrebbero voler normalizzare gli stili di default su quanti più browser possibile.

## Osservazioni

Il reset di Meyer, se efficace, apporta le stesse modifiche a quasi tutti gli elementi comunemente usati. Questo ha il risultato di inquinare le finestre di ispezione del browser Web con gli stessi stili applicati più e più volte e crea più lavoro per il browser (più regole da applicare a più elementi). La tecnica Normalize, d'altra parte, è molto più focalizzata e meno una tecnica a pennello largo. Ciò semplifica il lavoro sulla parte del browser e risulta meno ingombrante negli strumenti di ispezione del browser.

## Examples

### normalize.css

I browser hanno un set predefinito di stili CSS che usano per gli elementi di rendering. Alcuni di questi stili possono anche essere personalizzati utilizzando le impostazioni del browser per modificare, ad esempio, le definizioni predefinite di faccia e dimensioni dei caratteri. Gli stili contengono la definizione di quali elementi dovrebbero essere a livello di blocco o in linea, tra le altre cose.

Poiché questi stili predefiniti hanno un certo margine di manovra secondo le specifiche del linguaggio e poiché i browser potrebbero non seguire correttamente le specifiche possono differire da browser a browser.

È qui che entra in gioco [normalize.css](#). Sovrascrive le incoerenze più comuni e corregge i bug noti.

## Che cosa fa

- Preserva i valori predefiniti utili, a differenza di molti resettaggi CSS.
- Normalizza gli stili per una vasta gamma di elementi.
- Corregge bug e inconsistenze comuni del browser.

- Migliora l'usabilità con modifiche sottili.
- Spiega quale codice utilizza i commenti dettagliati.

Quindi, includendo `normalize.css` nel tuo progetto, il tuo design apparirà più simile e coerente tra i diversi browser.

## Differenza di `reset.css`

Potresti aver sentito di `reset.css`. Qual è la differenza tra i due?

Mentre `normalize.css` fornisce consistenza impostando proprietà diverse su valori predefiniti unificati, `reset.css` raggiunge la coerenza **rimuovendo** tutto lo stile di base che può essere applicato da un browser. Anche se all'inizio potrebbe sembrare una buona idea, in realtà ciò significa che devi scrivere **tutte le** regole tu stesso, il che è contrario al fatto di avere uno standard solido.

## Approcci ed esempi

I ripristini CSS adottano approcci separati ai valori predefiniti del browser. Reset CSS di Eric Meyer è in circolazione da un po' di tempo. Il suo approccio annulla molti degli elementi del browser che sono noti per causare problemi fin da subito. Quanto segue è tratto dalla sua versione (v2.0 | 20110126) Reset CSS.

```
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
```

## Ripristina CSS di Eric Meyer

Normalizza i CSS sull'altro e tratta molti di questi separatamente. Quello che segue è un esempio della versione (v4.2.0) del codice.

```
/**
 * 1. Change the default font family in all browsers (opinionated).
 * 2. Correct the line height in all browsers.
```

```

* 3. Prevent adjustments of font size after orientation changes in IE and iOS.
*/

/* Document
===== */

html {
  font-family: sans-serif; /* 1 */
  line-height: 1.15; /* 2 */
  -ms-text-size-adjust: 100%; /* 3 */
  -webkit-text-size-adjust: 100%; /* 3 */
}

/* Sections
===== */

/**
 * Remove the margin in all browsers (opinionated).
 */

body {
  margin: 0;
}

/**
 * Add the correct display in IE 9-.
 */

article,
aside,
footer,
header,
nav,
section {
  display: block;
}

/**
 * Correct the font size and margin on `h1` elements within `section` and
 * `article` contexts in Chrome, Firefox, and Safari.
 */

h1 {
  font-size: 2em;
  margin: 0.67em 0;
}

```

## Normalizza CSS

Leggi Normalizzazione degli stili del browser online:

<https://riptutorial.com/it/css/topic/1211/normalizzazione-degli-stili-del-browser>

# Capitolo 39: Opacità

## Sintassi

- opacità: numero (\* strettamente compreso tra 0 e 1) | ereditare | iniziale | unset;

## Osservazioni

Se non vuoi applicare l'opacità, puoi usare questo invece:

```
background: rgba (255, 255, 255, 0,6);
```

risorse:

- MDN: <https://developer.mozilla.org/en/docs/Web/CSS/opacity> ;
- W3C Transparency: la proprietà 'opacity': <https://www.w3.org/TR/css3-color/#transparency>
- Supporto per browser: <http://caniuse.com/#feat=css-opacity>

## Examples

### Proprietà Opacità

L'opacità di un elemento può essere impostata usando la proprietà `opacity` . I valori possono essere ovunque da `0.0` (trasparente) a `1.0` (opaco).

### Esempio di utilizzo

```
<div style="opacity:0.8;">  
  This is a partially transparent element  
</div>
```

| Costo dell'immobile         | Trasparenza                 |
|-----------------------------|-----------------------------|
| <code>opacity: 1.0;</code>  | Opaco                       |
| <code>opacity: 0.75;</code> | 25% trasparente (75% opaco) |
| <code>opacity: 0.5;</code>  | 50% trasparente (50% opaco) |
| <code>opacity: 0.25;</code> | 75% trasparente (25% opaco) |
| <code>opacity: 0.0;</code>  | Trasparente                 |

### Compatibilità IE per `opacity`

Per utilizzare l' `opacity` in tutte le versioni di IE, l'ordine è:

```
.transparent-element {  
  /* for IE 8 & 9 */  
  -ms-filter:"progid:DXImageTransform.Microsoft.Alpha(Opacity=60)"; // IE8  
  /* works in IE 8 & 9 too, but also 5, 6, 7 */  
  filter: alpha(opacity=60); // IE 5-7  
  /* Modern Browsers */  
  opacity: 0.6;  
}
```

Leggi Opacità online: <https://riptutorial.com/it/css/topic/2864/opacita>

# Capitolo 40: Posizionamento

## Sintassi

- posizione: statico | absolute | fixed | relativo | sticky | initial | inherit | unset;
- z-index: auto | *numero* | iniziale | eredita;

## Parametri

| Parametro   | Dettagli   |
|-------------|--|
| statico     | Valore predefinito. Gli elementi vengono visualizzati in ordine, così come appaiono nel flusso del documento. Le proprietà superiore, destra, inferiore, sinistra e z-index non si applicano.                    |
| parente     | L'elemento è posizionato rispetto alla sua posizione normale, quindi a <code>left:20px</code> aggiunge 20 pixel alla posizione SINISTRA dell'elemento  |
| fisso       | L'elemento è posizionato rispetto alla finestra del browser  |
| assoluto    | L'elemento è posizionato rispetto al suo primo elemento antenato posizionato (non statico)   |
| iniziale    | Imposta questa proprietà sul valore predefinito.   |
| ereditare   | Eredita questa proprietà dal suo elemento padre  |
| appiccicoso | Caratteristica sperimentale Si comporta come la <code>position: static</code> nel suo genitore fino a quando viene raggiunta una determinata soglia di offset, quindi agisce come <code>position: fixed</code> . |
| unset       | Combinazione di iniziale ed ereditaria. Maggiori informazioni <a href="#">qui</a> .  |

## Osservazioni

**Normal Flow** è il flusso di elementi se la posizione dell'elemento è *statica*.

1. definire la *larghezza* è utile perché in alcuni casi impedisce la sovrapposizione del contenuto dell'elemento.

## Examples

### Posizione fissa

Definendo la posizione come fissa possiamo rimuovere un elemento dal flusso del documento e impostarne la posizione relativamente alla finestra del browser. Un uso ovvio è quando vogliamo che qualcosa sia visibile quando scorriamo fino alla fine di una lunga pagina.

```
#stickyDiv {
  position:fixed;
  top:10px;
  left:10px;
}
```

## Elementi sovrapposti con z-index

Per modificare gli elementi posizionati [nell'ordine di stack](#) predefinito (proprietà della `position` impostata su `relative`, `absolute` o `fixed`), utilizzare la proprietà `z-index`.

Più alto è lo z-index, più in alto nel contesto di stacking (sull'asse z) è posizionato.

---

## Esempio

Nell'esempio seguente, un valore z-index di 3 mette il verde in cima, uno z-index di 2 mette il rosso appena sotto di esso, e un indice z di 1 mette blu sotto quello.

## HTML

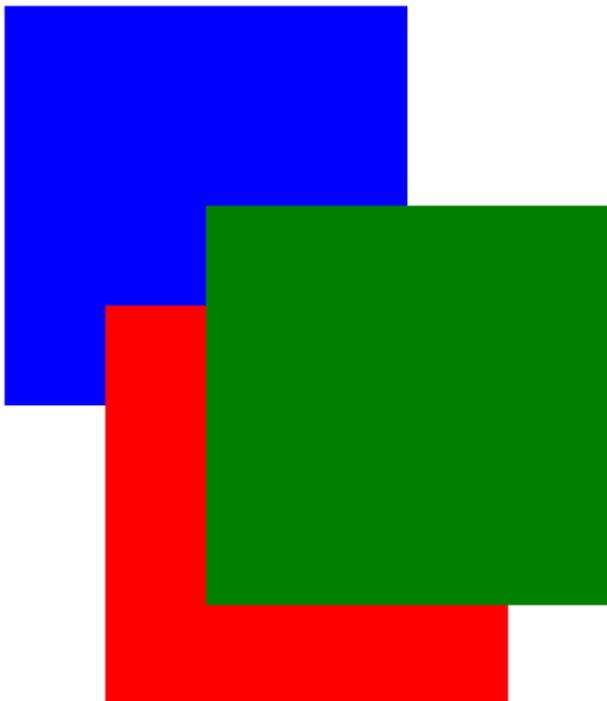
```
<div id="div1"></div>
<div id="div2"></div>
<div id="div3"></div>
```

## CSS

```
div {
  position: absolute;
  height: 200px;
  width: 200px;
}
div#div1 {
  z-index: 1;
  left: 0px;
  top: 0px;
  background-color: blue;
}
div#div2 {
  z-index: 3;
  left: 100px;
  top: 100px;
  background-color: green;
}
div#div3 {
  z-index: 2;
```

```
left: 50px;
top: 150px;
background-color: red;
}
```

Questo crea il seguente effetto:



Guarda un esempio funzionante su [JSFiddle](#) .

---

## Sintassi

```
z-index: [ number ] | auto;
```

| Parametro | Dettagli  |
|-----------|---|
| number    | Un valore intero. Un numero più alto è più alto nella pila <code>z-index</code> . 0 è il valore predefinito. Valori negativi sono ammessi |
| auto      | Dà all'elemento lo stesso contesto di stack del suo genitore. ( <b>Predefinito</b> )  |

---

## Osservazioni

Tutti gli elementi sono disposti in un asse 3D in CSS, incluso un asse di profondità, misurato dalla proprietà `z-index` . `z-index` funziona solo sugli elementi posizionati: (vedi: [Perché z-index necessita di una posizione definita per funzionare?](#) ). L'unico valore in cui viene ignorato è il valore

predefinito, `static` .

Ulteriori informazioni sulla proprietà z-index e sui contesti di impilamento nella [Specifica CSS](#) sulla presentazione a livelli e sulla [rete di sviluppatori Mozilla](#) .

## Posizione relativa

Il posizionamento relativo sposta l'elemento in relazione a dove sarebbe stato nel *flusso normale* .  
Proprietà offset:

1. superiore
2. sinistra
3. destra
4. parte inferiore

sono usati per indicare quanto lontano spostare l'elemento da dove sarebbe stato nel flusso normale.

```
.relpos{
  position:relative;
  top:20px;
  left:30px;
}
```

Questo codice sposterà la casella contenente l'elemento con attributo `class = "relpos"` 20px down e 30px a destra da dove sarebbe stato nel flusso normale.

## Posizione assoluta

Quando viene utilizzato il posizionamento assoluto, la casella dell'elemento desiderato viene eliminata dal *flusso normale* e non influisce più sulla posizione degli altri elementi sulla pagina.  
Proprietà offset:

1. superiore
2. sinistra
3. destra
4. parte inferiore

specificare che l'elemento dovrebbe apparire in relazione al suo prossimo elemento non statico contenente.

```
.abspos{
  position:absolute;
  top:0px;
  left:500px;
}
```

Questo codice sposterà la casella contenente l'elemento con attributo `class="abspos"` basso a 0px e a destra 500px rispetto al relativo elemento contenitore.

## Posizionamento statico

La posizione di default di un elemento è `static` . Per citare [MDN](#) :

Questa parola chiave consente all'elemento di utilizzare il comportamento normale, ovvero è disposto nella sua posizione corrente nel flusso. Le proprietà superiore, destra, inferiore, sinistra e z-index non si applicano.

```
.element {  
  position:static;  
}
```

Leggi Posizionamento online: <https://riptutorial.com/it/css/topic/935/posizionamento>

---

# Capitolo 41: Prestazione

## Examples

### Usa la trasformazione e l'opacità per evitare il layout del trigger

La modifica di alcuni attributi CSS fa sì che il browser calcoli in modo sincrono lo stile e il layout, il che è un aspetto negativo quando è necessario animare a 60 fps.

---

## NON

Animazione con layout di trigger `left` e `top`.

```
#box {
  left: 0;
  top: 0;
  transition: left 0.5s, top 0.5s;
  position: absolute;
  width: 50px;
  height: 50px;
  background-color: gray;
}

#box.active {
  left: 100px;
  top: 100px;
}
```

**La demo ha** preso **11.7ms** per il rendering, **9.8ms** per la pittura



Capture:



Network



JS Pro

500 ms

1000 ms

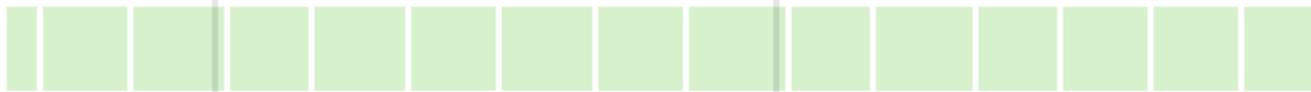
500 ms



800 ms

900 ms

1000 ms



Interactions

Main



Summary

Bottom-Up

Call Tree

Event Log

Group by Category



Self Time



Total Time



Activ

11.7 ms 54.2 %

11.7 ms 54.2 %



4.2 ms 19.5 %

4.2 ms 19.5 %

3.9 ms 18.2 %

3.9 ms 18.2 %

1.8 ms 8.3 %

1.8 ms 8.3 %

# Capitolo 42: Proprietà filtro

## Sintassi

- filtro: nessuno (valore predefinito)
- filter: initial (predefinito su none);
- filter: inherit (valore predefinito al valore padre);
- filtro: sfocatura (px)
- filtro: luminosità (numero |%)
- filtro: contrasto (numero |%)
- filtro: drop-shadow (orizzontale-ombra-px verticale-ombra-px ombra-sfocatura-px ombra- colore diffuso)
- filtro: scala di grigi (numero |%)
- filtro: hue-rotate (deg)
- filtro: invert (numero |%)
- filtro: opacità (numero |%)
- filtro: saturare (numero |%)
- filtro: seppia (numero |%)

## Parametri

| Valore                      | Descrizione   |
|-----------------------------|---|
| sfocatura (x)               | Sfoca l'immagine di x pixel.  |
| luminosità (x)              | Illumina l'immagine con qualsiasi valore superiore a 1.0 o 100%. Al di sotto di questo, l'immagine sarà oscurata.                         |
| contrasto (x)               | Fornisce più contrasto all'immagine con qualsiasi valore superiore a 1.0 o 100%. Al di sotto di questo, l'immagine diventerà meno satura. |
| drop-shadow (h, v, x, y, z) | Dà all'immagine un'ombra. h e v possono avere valori negativi. x, y e z sono opzionali.   |
| scala di grigi (x)          | Mostra l'immagine in scala di grigi, con un valore massimo di 1.0 o 100%.   |
| tonalità rotazione (x)      | Applica una rotazione della tonalità all'immagine.  |
| invertire (x)               | Inverte il colore dell'immagine con un valore massimo di 1.0 o 100%.  |
| opacità (x)                 | Imposta l'opaco / trasparente dell'immagine con un valore massimo di 1.0 o 100%.  |
| saturare (x)                | Saturizza l'immagine con qualsiasi valore superiore a 1.0 o 100%. Sotto di ciò, l'immagine inizierà a desaturare.                         |

| Valore     | Descrizione  |
|------------|--|
| seppia (x) | Converte l'immagine in seppia con un valore massimo di 1.0 o 100%. |

## Osservazioni

1. Poiché il filtro è una funzione sperimentale, è necessario utilizzare il prefisso `-webkit`. Potrebbe cambiare in sintassi e comportamento, ma probabilmente le modifiche saranno ridotte.
2. Potrebbe non essere supportato nelle versioni precedenti dei principali browser. Potrebbe non essere completamente supportato nei browser per dispositivi mobili.
3. A causa del suo supporto relativamente limitato, prova a usare `box-shadow` anziché `filter: drop-shadow()`. Usa l' `opacity` invece del `filter: opacity()`.
4. Può essere animato tramite Javascript / jQuery. Per Javascript, usa `object.style.WebkitFilter`.
5. Controlla [W3Schools](#) o [MDN](#) per maggiori informazioni.
6. W3Schools ha anche una [pagina demo](#) per tutti i diversi tipi di valori dei filtri.

## Examples

### Ombra esterna (usa invece `box-shadow` se possibile)

#### HTML

```
<p>My shadow always follows me.</p>
```

#### CSS

```
p {  
  -webkit-filter: drop-shadow(10px 10px 1px green);  
  filter: drop-shadow(10px 10px 1px green);  
}
```

#### Risultato

My shadow always follows me.  
*My shadow always follows me.*

### Più valori di filtro

Per utilizzare più filtri, separare ogni valore con uno spazio.

#### HTML

```
<img src='donald-duck.png' alt='Donald Duck' title='Donald Duck' />
```

## CSS

```
img {  
  -webkit-filter: brightness(200%) grayscale(100%) sepia(100%) invert(100%);  
  filter: brightness(200%) grayscale(100%) sepia(100%) invert(100%);  
}
```

## Risultato



## Tonalità Ruota

## HTML

```
<img src='donald-duck.png' alt='Donald Duck' title='Donald Duck' />
```

## CSS

```
img {  
  -webkit-filter: hue-rotate(120deg);  
  filter: hue-rotate(120deg);  
}
```

## Risultato



## Inverti colore

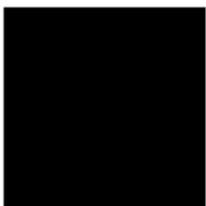
### HTML

```
<div></div>
```

### CSS

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: white;  
  -webkit-filter: invert(100%);  
  filter: invert(100%);  
}
```

### Risultato



Trasforma da bianco a nero.

### blur

### HTML

```
<img src='donald-duck.png' alt='Donald Duck' title='Donald Duck' />
```

### CSS

```
img {  
  -webkit-filter: blur(1px);  
  filter: blur(1px);  
}
```

## Risultato



Ti fa venire voglia di fregarti gli occhiali.

Leggi **Proprietà filtro** online: <https://riptutorial.com/it/css/topic/1567/proprieta-filtro>

---

# Capitolo 43: Proprietà personalizzate (variabili)

## introduzione

Le variabili CSS consentono agli autori di creare valori riutilizzabili che possono essere utilizzati in tutto un documento CSS.

Ad esempio, è comune nei CSS riutilizzare un singolo colore nel documento. Prima delle variabili CSS questo significherebbe riutilizzare lo stesso valore di colore più volte in un documento. Con le variabili CSS il valore del colore può essere assegnato a una variabile e referenziato in più punti. Questo rende più semplice cambiare i valori ed è più semantico rispetto all'utilizzo di valori CSS tradizionali.

## Sintassi

- `: root {}` / `*` pseudo-classe che consente una maggiore definizione globale di variabili `*` /
- `--variable-name: value ;` / `*` definisce la variabile `*` /
- `var (- nome-variabile, valore-predefinito )` / `*` usa la variabile definita con il valore di default `fallback` `*` /

## Osservazioni

**Le variabili CSS sono attualmente considerate una tecnologia sperimentale.**

---

## SUPPORTO AL BROWSER / COMPATIBILITÀ

**Firefox:** versione **31** + (*abilitato per impostazione predefinita*)

[Maggiori informazioni da Mozilla](#)

**Chrome:** versione **49** + (*abilitata per impostazione predefinita*) .

*"Questa funzione può essere abilitata nella versione 48 di Chrome per i test abilitando la funzionalità `experimental Web Platform` . Inserisci `chrome://flags/` nella barra degli indirizzi di Chrome per accedere a questa impostazione."*

**IE:** non supportato.

**Edge:** [Under Development](#)

**Safari:** versione **9.1+**

# Examples

## Colore variabile

```
:root {
  --red: #b00;
  --blue: #4679bd;
  --grey: #ddd;
}
.Bx1 {
  color: var(--red);
  background: var(--grey);
  border: 1px solid var(--red);
}
```

## Dimensioni variabili

```
:root {
  --W200: 200px;
  --W10: 10px;
}
.Bx2 {
  width: var(--W200);
  height: var(--W200);
  margin: var(--W10);
}
```

## Cascading variabile

Le variabili CSS si sovrappongono in modo simile alle altre proprietà e possono essere ripristinate in modo sicuro.

È possibile definire le variabili più volte e solo la definizione con la massima specificità verrà applicata all'elemento selezionato.

Supponendo questo HTML:

```
<a class="button">Button Green</a>
<a class="button button_red">Button Red</a>
<a class="button">Button Hovered On</a>
```

Possiamo scrivere questo CSS:

```
.button {
  --color: green;
  padding: .5rem;
  border: 1px solid var(--color);
  color: var(--color);
}

.button:hover {
  --color: blue;
```

```
}  
  
.button_red {  
  --color: red;  
}
```

E ottieni questo risultato:



## Validi / invalidi

**Denominazione** Quando si nominano le variabili CSS, contiene solo lettere e trattini, proprio come le altre proprietà CSS (es: line-height, -moz-box-sizing) ma dovrebbe iniziare con due trattini (-)

```
//These are Invalids variable names  
--123color: blue;  
--#color: red;  
--bg_color: yellow  
--$width: 100px;  
  
//Valid variable names  
--color: red;  
--bg-color: yellow  
--width: 100px;
```

**Le variabili CSS sono sensibili al maiuscolo / minuscolo.**

```
/* The variable names below are all different variables */  
--pcolor: ;  
--Pcolor: ;  
--pColor: ;
```

**Spazio vuoto vs.**

```
/* Invalid */  
  --color;;  
  
/* Valid */  
  --color: ; /* space is assigned */
```

**concatenazioni**

```
/* Invalid - CSS doesn't support concatenation*/  
.logo{  
  --logo-url: 'logo';  
  background: url('assets/img/' var(--logo-url) '.png');  
}  
  
/* Invalid - CSS bug */
```

```

.logo{
  --logo-url: 'assets/img/logo.png';
  background: url(var(--logo-url));
}

/* Valid */
.logo{
  --logo-url: url('assets/img/logo.png');
  background: var(--logo-url);
}

```

## Attento quando si usano le unità

```

/* Invalid */
--width: 10;
width: var(--width)px;

/* Valid */
--width: 10px;
width: var(--width);

/* Valid */
--width: 10;
width: calc(1px * var(--width)); /* multiply by 1 unit to convert */
width: calc(1em * var(--width));

```

## Con le domande dei media

È possibile reimpostare le variabili all'interno delle query multimediali e avere quei nuovi valori in cascata ovunque vengano utilizzati, cosa che non è possibile con le variabili del preprocessore.

Qui, una query multimediale modifica le variabili utilizzate per impostare una griglia molto semplice:

### HTML

```

<div></div>
<div></div>
<div></div>
<div></div>

```

### CSS

```

:root{
  --width: 25%;
  --content: 'This is desktop';
}
@media only screen and (max-width: 767px){
  :root{
    --width:50%;
    --content: 'This is mobile';
  }
}
@media only screen and (max-width: 480px){
  :root{

```

```
        --width:100%;
    }
}

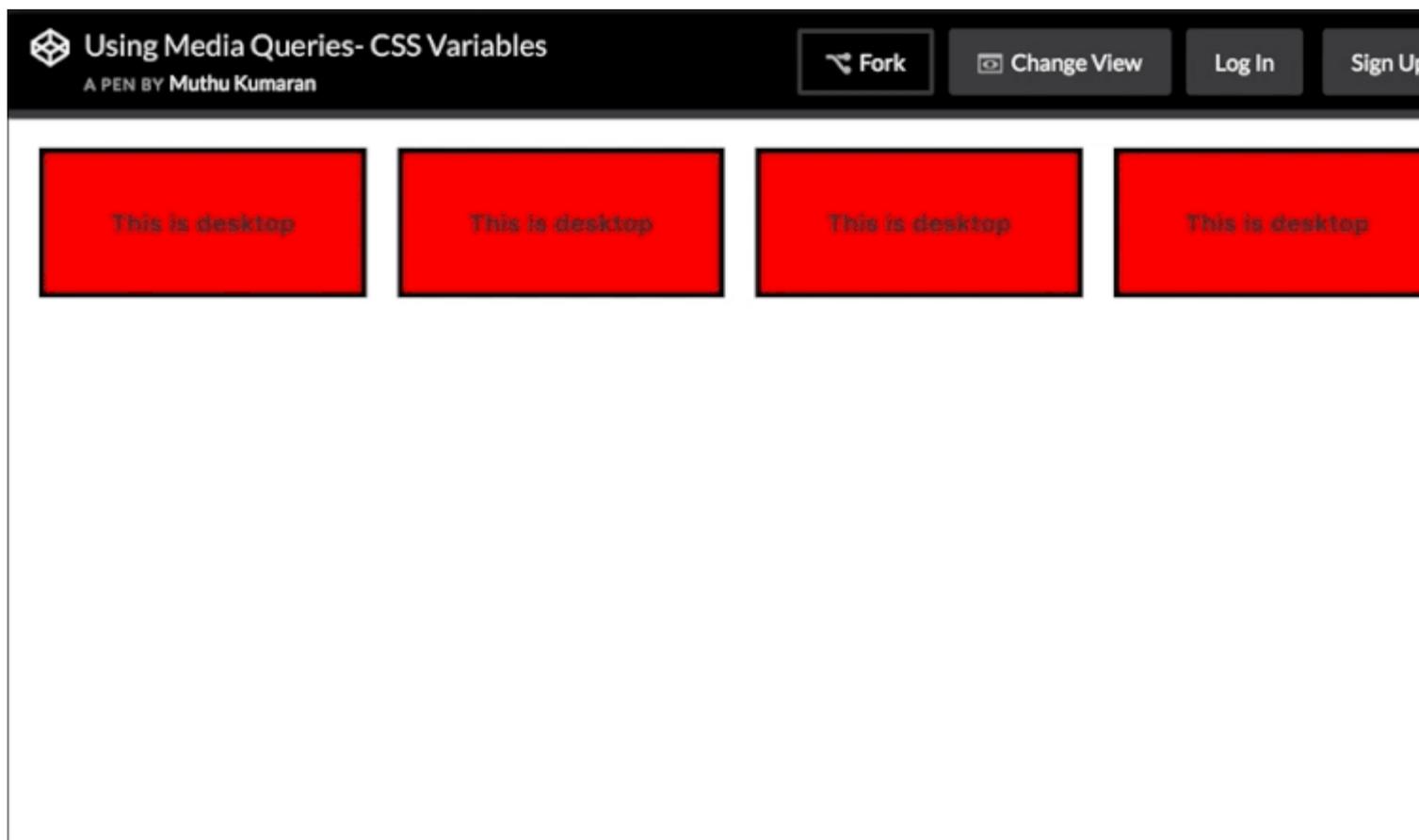
div{
    width: calc(var(--width) - 20px);
    height: 100px;
}
div:before{
    content: var(--content);
}

/* Other Styles */
body {
    padding: 10px;
}

div{
    display: flex;
    align-items: center;
    justify-content: center;
    font-weight:bold;
    float:left;
    margin: 10px;
    border: 4px solid black;
    background: red;
}
```

Puoi provare a ridimensionare la finestra in questa [demo di CodePen](#)

Ecco uno screenshot animato del ridimensionamento in azione:



Leggi Proprietà personalizzate (variabili) online: <https://riptutorial.com/it/css/topic/1755/proprietà-personalizzate--variabili->

# Capitolo 44: Pseudo-Elementi

## introduzione

Gli pseudo-elementi, proprio come le pseudo-classi, vengono aggiunti a un selettore CSS ma, invece di descrivere uno stato speciale, consentono di definire e definire determinate parti di un elemento html.

Ad esempio, lo pseudoelemento `:: first-letter` ha come target solo la prima lettera di un elemento di blocco specificato dal selettore.

## Sintassi

- `selector :: pseudo-element {property: value}`

## Parametri

| pseudo-elemento               | Descrizione  |
|-------------------------------|--|
| <code>::after</code>          | Inserisci il contenuto dopo il contenuto di un elemento  |
| <code>::before</code>         | Inserisci il contenuto prima del contenuto di un elemento  |
| <code>::first-letter</code>   | Seleziona la prima lettera di ogni elemento  |
| <code>::first-line</code>     | Seleziona la prima riga di ciascun elemento  |
| <code>::selection</code>      | Corrisponde alla parte di un elemento selezionato da un utente   |
| <code>::backdrop</code>       | Utilizzato per creare uno sfondo che nasconde il documento sottostante per un elemento nello stack del livello superiore |
| <code>::placeholder</code>    | Permette di modellare il testo segnaposto di un elemento del modulo (Sperimentale)                                       |
| <code>::marker</code>         | Per applicare gli attributi in stile elenco su un determinato elemento (Sperimentale)                                    |
| <code>::spelling-error</code> | Rappresenta un segmento di testo contrassegnato erroneamente dal browser (sperimentale)                                  |
| <code>::grammar-error</code>  | Rappresenta un segmento di testo che il browser ha contrassegnato come grammaticamente scorretto (sperimentale)          |

# Osservazioni

- A volte vedrai i doppi due punti ( :: ) invece di uno solo ( : . Questo è un modo per separare pseudo-classi da pseudo-elementi, ma alcuni vecchi browser come Internet Explorer 8 supporta **solo** singolo due punti ( : ) per la pseudo-elementi.
- Si può usare solo uno pseudo-elemento in un selettore. Deve apparire dopo i selettori semplici nella dichiarazione.
- Gli pseudo-elementi non fanno parte del DOM, quindi non sono targettizzabili da `:hover` o altri eventi utente.

## Examples

### Pseudo-Elementi

Gli pseudo-elementi vengono aggiunti ai selettori ma, invece di descrivere uno stato speciale, consentono di definire alcune parti di un documento.

L'attributo `content` è richiesto per gli pseudo-elementi da renderizzare; tuttavia, l'attributo può avere un valore vuoto (ad es. `content: ""` ).

```
div::after {
  content: 'after';
  color: red;
  border: 1px solid red;
}

div {
  color: black;
  border: 1px solid black;
  padding: 1px;
}

div::before {
  content: 'before';
  color: green;
  border: 1px solid green;
}
```



### Pseudo-elementi nelle liste

Gli pseudo-elementi sono spesso usati per cambiare l'aspetto delle liste (principalmente per liste non ordinate, `ul` ).

Il primo passo è rimuovere i punti elenco predefiniti:

```
ul {
```

```
list-style-type: none;
}
```

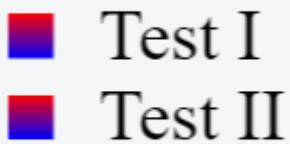
Quindi aggiungi lo stile personalizzato. In questo esempio, creeremo caselle sfumate per i punti elenco.

```
li:before {
  content: "";
  display: inline-block;
  margin-right: 10px;
  height: 10px;
  width: 10px;
  background: linear-gradient(red, blue);
}
```

## HTML

```
<ul>
  <li>Test I</li>
  <li>Test II</li>
</ul>
```

## Risultato



Leggi Pseudo-Elementi online: <https://riptutorial.com/it/css/topic/911/pseudo-elementi>

# Capitolo 45: Query di caratteristiche

## Sintassi

- `@supports [condizione] {/ * Regole CSS da applicare * /}`

## Parametri

| Parametro                      | Dettagli  |
|--------------------------------|---|
| <code>(property: value)</code> | Valuta true se il browser è in grado di gestire la regola CSS. Sono richieste le parentesi attorno alla regola. |
| <code>and</code>               | Restituisce vero solo se entrambe le condizioni precedenti e successive sono vere.                              |
| <code>not</code>               | Annulla la prossima condizione  |
| <code>or</code>                | Restituisce vero se la condizione precedente o successiva è vera.   |
| <code>(...)</code>             | Condizioni di gruppo  |

## Osservazioni

Il rilevamento delle funzioni tramite `@supports` è supportato in Edge, Chrome, Firefox, Opera e Safari 9 e `@supports` successive.

## Examples

### L'utilizzo di base `@supports`

```
@supports (display: flex) {  
  /* Flexbox is available, so use it */  
  .my-container {  
    display: flex;  
  }  
}
```

In termini di sintassi, `@supports` è molto simile a `@media`, ma invece di rilevare la dimensione e l'orientamento dello schermo, `@supports` rileva se il browser può gestire una determinata regola CSS.

Invece di fare qualcosa come `@supports (flex)`, nota che la regola è `@supports (display: flex)`.

### Rilevazione delle feature di concatenazione

Per rilevare più funzioni contemporaneamente, utilizzare l'operatore `and` .

```
@supports (transform: translateZ(1px)) and (transform-style: preserve-3d) and (perspective: 1px) {
  /* Probably do some fancy 3d stuff here */
}
```

C'è anche un operatore `or` un operatore `not` :

```
@supports (display: flex) or (display: table-cell) {
  /* Will be used if the browser supports flexbox or display: table-cell */
}
@supports not (-webkit-transform: translate(0, 0, 0)) {
  /* Will *not* be used if the browser supports -webkit-transform: translate(...) */
}
```

Per la migliore esperienza di `@supports` , prova a raggruppare le espressioni logiche con le parentesi:

```
@supports ((display: block) and (zoom: 1)) or ((display: flex) and (not (display: table-cell))) or (transform: translateX(1px)) {
  /* ... */
}
```

Funzionerà se il browser

1. Supporti `display: block` **AND** `zoom: 1` , **o**
2. `display: flex` **supportati** `display: flex` **AND NOT** `display: table-cell` , **o**
3. Supporta la `transform: translateX(1px)` .

Leggi Query di caratteristiche online: <https://riptutorial.com/it/css/topic/5024/query-di-caratteristiche>

# Capitolo 46: Ritaglio e mascheratura

## Sintassi

- **Ritaglio**
- clip-path: <clip-source> | [<forma base> || <clip-geometry-box>] | nessuna
- **mascheramento**
- immagine-maschera: [nessuno | <referimento-maschera>] #
- mask-mode: [<mask-mode>] #
- mask-repeat: [<stile di ripetizione>] #
- posizione della maschera: [<posizione>] #
- mask-clip: [<geometry-box> | no-clip] #
- origine maschera: [<geometria-box>] #
- dimensione maschera: [<bg-size>] #
- mask-composite: [<compositing-operator>] #
- maschera: [<referimento maschera> <modalità mascheratura>? || <posizione> [/ <bg-size>]? || <stile di ripetizione> || <geometry-box> || [<geometry-box> | no-clip] || <compositing-operator>] #

## Parametri

| Parametro            | Dettagli   |
|----------------------|--|
| clip-fonte           | Un URL che può puntare a un elemento SVG in linea (o) a un elemento SVG in un file esterno che contiene la definizione del percorso della clip.  |
| basic-figura         | Si riferisce a uno tra <code>inset()</code> , <code>circle()</code> , <code>ellipse()</code> o <code>polygon()</code> . Usando una di queste funzioni viene definito il tracciato di ritaglio. Queste funzioni di forma funzionano esattamente nello stesso modo in <a href="#">Shapes per float</a>   |
| clip-geometria-box   | Questo può avere uno tra <code>content-box</code> , <code>padding-box</code> , <code>border-box</code> , <code>margin-box</code> , <code>fill-box</code> , <code>stroke-box</code> , <code>view-box</code> come valori. Quando questo viene fornito senza alcun valore per <forma base>, i bordi della casella corrispondente vengono utilizzati come percorso per il ritaglio. Quando viene utilizzato con una <forma base>, funge da riquadro di riferimento per la forma. |
| maschera-riferimento | Questo può essere <code>none</code> o un'immagine o un URL di riferimento a un'origine di immagine maschera.   |
| repeat-style         | Specifica come la maschera deve essere ripetuta o affiancata negli assi X e Y. I valori supportati sono <code>repeat-x</code> , <code>repeat-y</code> , <code>repeat</code> , <code>space</code> , <code>round</code> , <code>no-repeat</code> .   |
| maschera-mode        | Può essere <code>alpha</code> o <code>luminance</code> o <code>auto</code> e indica se la maschera deve essere trattata come maschera alfa o maschera di luminanza. Se non viene fornito   |

| Parametro              | Dettagli   |
|------------------------|--|
|                        | alcun valore e il riferimento alla maschera è un'immagine diretta, verrà considerato come maschera alfa (o) se il riferimento alla maschera è un URL, allora sarebbe considerato come maschera di luminanza.   |
| posizione              | Specifica la posizione di ogni livello maschera ed è simile nel comportamento alla proprietà <code>background-position</code> . Il valore può essere fornito in 1 sintassi del valore (come in <code>top, 10%</code> ) o in 2 sintassi del valore (come in <code>top right, 50% 50%</code> ).  |
| Geometria-box          | Specifica la casella in cui la maschera deve essere ritagliata ( <i>area di pittura maschera</i> ) o la casella che deve essere utilizzata come riferimento per l'origine della <i>maschera</i> ( <i>area di posizionamento maschera</i> ) in base alla proprietà. L'elenco di valori possibili è <code>content-box, padding-box, border-box, margin-box, fill-box, stroke-box, view-box</code> . Spiegazione dettagliata su come funzionano ciascuno di questi valori è disponibile nella specifica <a href="#">W3C</a> . |
| bg-size                | Rappresenta la dimensione di ogni livello maschera-immagine e ha la stessa sintassi <code>background-size</code> . Il valore può essere lunghezza o percentuale o auto o copertura o contenere. Lunghezza, percentuale e auto possono essere forniti come valore singolo o come uno per ciascun asse.  |
| composizione-operatore | Può essere uno qualsiasi tra <code>add, subtract, exclude, multiply</code> per layer e definisce il tipo di operazione di compositing che dovrebbe essere usata per questo layer con quelli sotto di esso. Spiegazione dettagliata su ciascun valore è disponibile nelle <a href="#">specifiche W3C</a> .  |

## Osservazioni

**CSS Clipping and Masking** sono concetti molto nuovi e quindi il supporto del browser per queste proprietà è piuttosto basso.

## Maschere:

Al momento della stesura di questo articolo (16 luglio), Chrome, Safari e Opera supportano queste proprietà con il prefisso `-webkit-`.

Firefox non richiede prefissi ma supporta le maschere solo se utilizzato con elementi `mask SVG`. Per gli elementi `mask SVG` in linea, la sintassi è `mask: url(#msk)` mentre per l'utilizzo di elementi `mask` in un file SVG esterno la sintassi è `mask: url('yourfilepath/yourfilename.svg#msk')`. `#msk` in entrambi i casi si riferisce alla `id` della `mask` elemento che ci si riferisce. Come indicato in [questa risposta](#), al momento Firefox non supporta alcun parametro diverso dal `mask-reference` alla proprietà `mask`.

Internet Explorer (e Edge) non offre ancora alcun supporto per questa proprietà.

La proprietà in `mask-mode` è attualmente supportata da alcuni browser **con o senza** prefissi.

---

## Clip-path:

Al momento della scrittura (16 luglio), Chrome, Safari e Opera supportano il `clip-path` quando il percorso viene creato utilizzando forme di base (come `circle`, `polygon`) o la sintassi `url(#clipper)` con SVG in linea. Non supportano il ritaglio basato su forme che fanno parte di file SVG esterni. Inoltre, richiedono il prefisso `-webkit` per essere presenti.

Firefox supporta solo la sintassi `url()` per `clip-path` mentre Internet Explorer (e Edge) non offrono supporto.

## Examples

### Ritaglio (poligono)

---

## CSS:

```
div{
  width:200px;
  height:200px;
  background:teal;
  clip-path: polygon(0 0, 0 100%, 100% 50%); /* refer remarks before usage */
}
```

---

## HTML:

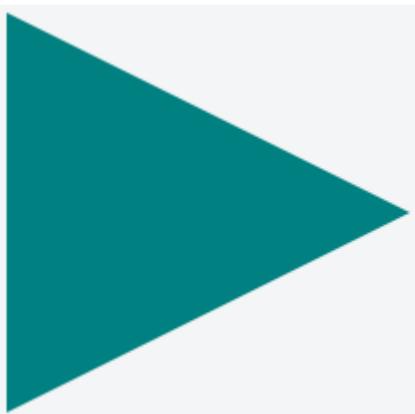
```
<div></div>
```

Nell'esempio sopra, un tracciato di ritaglio **poligonale** viene utilizzato per ritagliare l'elemento quadrato (200 x 200) in una forma triangolare. La forma di output è un triangolo perché il percorso inizia da (ovvero, le prime coordinate sono a) `0 0` - che è l'angolo in alto a sinistra della casella, quindi passa a `0 100%` - che è l'angolo in basso a sinistra della casella e infine al `100% 50%` che non è altro che il punto medio-centrale della scatola. Questi percorsi sono auto-chiudenti (cioè, il punto di partenza sarà il punto finale) e quindi la forma finale è quella di un triangolo.

Questo può anche essere usato su un elemento con un'immagine o un gradiente come sfondo.

[Visualizza esempio](#)

**Produzione:**



## Ritaglio (Cerchio)

### CSS:

```
div{
  width: 200px;
  height: 200px;
  background: teal;
  clip-path: circle(30% at 50% 50%); /* refer remarks before usage */
}
```

### HTML

```
<div></div>
```

Questo esempio mostra come tagliare un div a un cerchio. L'elemento è ritagliato in un cerchio il cui raggio è il 30% in base alle dimensioni della casella di riferimento con il suo punto centrale al centro della casella di riferimento. Qui dal momento che non viene fornita la <casella di clip-geometria> (in altre parole, casella di riferimento), la `border-box` di `border-box` dell'elemento verrà utilizzata come casella di riferimento.

La forma del cerchio deve avere un raggio e un centro con le coordinate  $(x, y)$  :

```
circle(radius at x y)
```

### Visualizza esempio

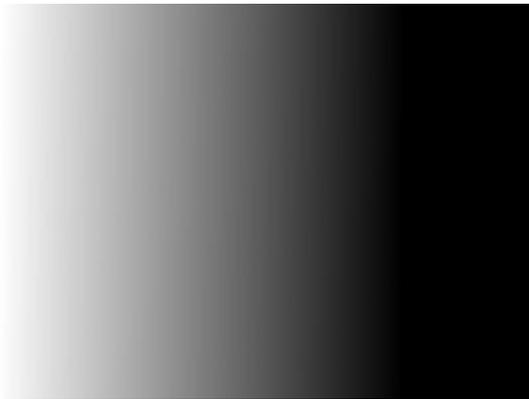
### Produzione:



quali livelli si sovrappongono all'elemento. Puoi immaginare questa maschera come un'immagine che consiste principalmente di due colori: bianco e nero.

**Maschera di luminanza** : il nero indica che la regione è opaca e bianca che è trasparente, ma esiste anche un'area grigia che è semitrasparente, quindi è possibile effettuare transizioni omogenee.

**Maschera Alpha** : solo sulle aree trasparenti della maschera l'elemento sarà opaco.



Questa immagine, ad esempio, può essere utilizzata come maschera di luminanza per rendere un elemento una transizione molto fluida da destra a sinistra e da opaca a trasparente.

La proprietà `mask` consente di specificare il tipo di maschera e un'immagine da utilizzare come layer.

Esempio

```
mask: url(masks.svg#rectangle) luminance;
```

Un elemento chiamato `rectangle` definito in `masks.svg` verrà utilizzato come **maschera di luminanza** sull'elemento.

## Maschera semplice che dissolve un'immagine da solida a trasparente

### CSS

```
div {
  height: 200px;
  width: 200px;
  background: url(http://loempixel.com/200/200/nature/1);
  mask-image: linear-gradient(to right, white, transparent);
}
```

### HTML

```
<div></div>
```

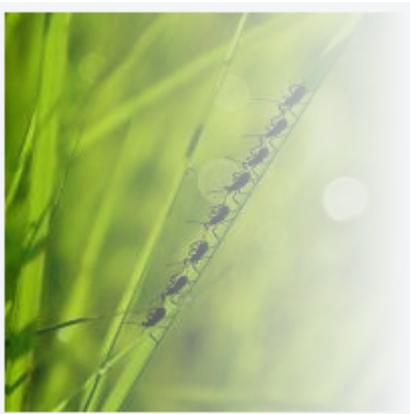
Nell'esempio sopra c'è un elemento con un'immagine come sfondo. La maschera applicata all'immagine (usando i CSS) fa sembrare che si stia dissolvendo da sinistra a destra.

Il mascheramento si ottiene utilizzando un `linear-gradient` che va dal bianco (a sinistra) a trasparente (a destra) come maschera. Essendo una maschera alfa, l'immagine diventa trasparente dove la maschera è trasparente.

### Uscita senza maschera:



### Uscita con la maschera:



**Nota:** come accennato in precedenza, l'esempio precedente funzionerebbe in Chrome, Safari e Opera solo se usato con il prefisso `-webkit`. Questo esempio (con un `linear-gradient` come immagine maschera) non è ancora supportato in Firefox.

## Utilizzo di maschere per tagliare un buco nel mezzo di un'immagine

### CSS

```
div {
  width: 200px;
  height: 200px;
  background: url(http://lorempixel.com/200/200/abstract/6);
  mask-image: radial-gradient(circle farthest-side at center, transparent 49%, white 50%); /*
  check remarks before using */
}
```

---

# HTML

Nell'esempio sopra, un cerchio trasparente viene creato al centro usando il `radial-gradient` e viene quindi utilizzato come maschera per produrre l'effetto di un cerchio che viene ritagliato dal centro di un'immagine.

**Immagine senza maschera:**



**Immagine con maschera:**



**Utilizzo di maschere per creare immagini con forme irregolari**

---

# CSS

```
div { /* check remarks before usage */
  height: 200px;
  width: 400px;
  background-image: url(http://lorempixel.com/400/200/nature/4);
  mask-image: linear-gradient(to top right, transparent 49.5%, white 50.5%), linear-
  gradient(to top left, transparent 49.5%, white 50.5%), linear-gradient(white, white);
  mask-size: 75% 25%, 25% 25%, 100% 75%;
  mask-position: bottom left, bottom right, top left;
  mask-repeat: no-repeat;
}
```

# HTML

```
<div></div>
```

Nell'esempio sopra, tre immagini a `linear-gradient` (che quando sono posizionate nelle posizioni appropriate coprono il 100% x 100% delle dimensioni del contenitore) vengono utilizzate come maschere per produrre un taglio triangolare trasparente nella parte inferiore dell'immagine.

**Immagine senza maschera:**



**Immagine con la maschera:**



Leggi Ritaglio e mascheratura online: <https://riptutorial.com/it/css/topic/3721/ritaglio-e-mascheratura>

---

# Capitolo 47: sfondi

## introduzione

Con i CSS puoi impostare colori, sfumature e immagini come sfondo di un elemento.

È possibile specificare varie combinazioni di immagini, colori e sfumature e regolare le dimensioni, il posizionamento e la ripetizione (tra gli altri) di questi.

## Sintassi

- colore di sfondo: colore | trasparente | iniziale | ereditare;
- background-image: url | nessuno | iniziale | ereditare;
- background-position: value;
- dimensione dello sfondo: <bg-size> [<bg-size>]
- <bg-size>: auto | lunghezza | copertina | contenere | iniziale | ereditare;
- background-repeat: repeat | repeat-x | repeat-y | nessuna ripetizione | iniziale | ereditare;
- background-origin: padding-box | scatola di confine | casella di contenuto | iniziale | ereditare;
- sfondo-clip: border-box | imbottitura-scatola | casella di contenuto | iniziale | ereditare;
- background-attachment: scroll | fisso | locale | iniziale | ereditare;
- sfondo: bg-color bg-image posizione / bg-size bg-repeat bg-origine bg-clip bg-attachment iniziale | ereditare;

## Osservazioni

- I gradienti CSS3 non funzioneranno su versioni di Internet Explorer inferiori a 10.

## Examples

### Colore di sfondo

La proprietà `background-color` imposta il colore di sfondo di un elemento usando un valore di colore o tramite parole chiave, come `transparent`, `inherit` o `initial`.

- **trasparente**, specifica che il colore di sfondo deve essere trasparente. Questo è il valore predefinito.
- **eredita**, eredita questa proprietà dal suo elemento genitore.
- **iniziale**, imposta questa proprietà sul valore predefinito.

Questo può essere applicato a tutti gli elementi e agli [pseudo-elementi](#) `::first-letter` / `::first-line`.

I colori in CSS possono essere specificati con [metodi diversi](#) .

---

## Nomi di colori

### CSS

```
div {
  background-color: red; /* red */
}
```

### HTML

```
<div>This will have a red background</div>
```

- L'esempio sopra riportato è uno dei molti modi in cui il CSS deve rappresentare un singolo colore.

---

## Codici colore esadecimali

Il codice esadecimale viene utilizzato per indicare le componenti RGB di un colore in notazione esadecimale in base 16. #ff0000, ad esempio, è rosso acceso, dove il componente rosso del colore è 256 bit (ff) e le corrispondenti parti verde e blu del colore sono 0 (00).

Se entrambi i valori in ciascuno dei tre accoppiamenti RGB (R, G e B) sono gli stessi, il codice colore può essere abbreviato in tre caratteri (la prima cifra di ciascun accoppiamento). #ff0000 può essere abbreviato in #f00 e #ffffff può essere abbreviato in #fff .

La notazione esadecimale non fa distinzione tra maiuscole e minuscole.

```
body {
  background-color: #de1205; /* red */
}

.main {
  background-color: #00f; /* blue */
}
```

---

## RGB / RGBa

Un altro modo per dichiarare un colore è usare RGB o RGBa.

RGB sta per Rosso, Verde e Blu e richiede tre valori separati tra 0 e 255, tra parentesi, che corrispondono ai valori dei colori decimali rispettivamente per rosso, verde e blu.

RGBa consente di aggiungere un ulteriore parametro alfa tra 0.0 e 1.0 per definire l'opacità.

```
header {
  background-color: rgb(0, 0, 0); /* black */
}

footer {
  background-color: rgba(0, 0, 0, 0.5); /* black with 50% opacity */
}
```

---

## HSL / HSLa

Un altro modo per dichiarare un colore è usare HSL o HSLa ed è simile a RGB e RGBA.

HSL è sinonimo di tonalità, saturazione e luminosità, ed è spesso chiamato anche HLS:

- La tonalità è un grado sulla ruota dei colori (da 0 a 360).
- La saturazione è una percentuale compresa tra 0% e 100%.
- La luminosità è anche una percentuale compresa tra 0% e 100%.

HSLa consente di aggiungere un ulteriore parametro alfa tra 0.0 e 1.0 per definire l'opacità.

```
li a {
  background-color: hsl(120, 100%, 50%); /* green */
}

#p1 {
  background-color: hsla(120, 100%, 50%, .3); /* green with 30% opacity */
}
```

---

## Interazione con l'immagine di sfondo

Le seguenti affermazioni sono tutte equivalenti:

```
body {
  background: red;
  background-image: url(partiallytransparentimage.png);
}

body {
  background-color: red;
  background-image: url(partiallytransparentimage.png);
}

body {
  background-image: url(partiallytransparentimage.png);
  background-color: red;
}

body {
  background: red url(partiallytransparentimage.png);
}
```

Porteranno tutti al colore rosso che viene mostrato sotto l'immagine, dove le parti dell'immagine

sono trasparenti, o l'immagine non viene visualizzata (forse come risultato della `background-repeat` dello `background-repeat` ).

Si noti che quanto segue non è equivalente:

```
body {
  background-image: url(partiallytransparentimage.png);
  background: red;
}
```

Qui, il valore dello `background` sovrascrive l' `background-image` .

Per maggiori informazioni sulla proprietà dello `background` , vedere [Stenografia background sfondo](#)

## Immagine di sfondo

La proprietà `background-image` viene utilizzata per specificare un'immagine di sfondo da applicare a tutti gli elementi corrispondenti. Per impostazione predefinita, questa immagine viene affiancata per coprire l'intero elemento, escluso il margine.

```
.myClass {
  background-image: url('/path/to/image.jpg');
}
```

Per utilizzare più immagini come `background-image` , definire l' `url()` separato da virgola `url()`

```
.myClass {
  background-image: url('/path/to/image.jpg'),
                  url('/path/to/image2.jpg');
}
```

Le immagini si impilano secondo il loro ordine con la prima immagine dichiarata in cima alle altre e così via.

| Valore                                 | Risultato  |
|--|--|
| <code>url('/path/to/image.jpg')</code> | Specificare il / i percorso / i dell'immagine dell'immagine di sfondo o una risorsa immagine specificata con lo schema URI di dati (è possibile omettere gli apostrofi), separare i multipli con una virgola |
| <code>none</code>                      | Nessuna immagine di sfondo   |
| <code>initial</code>                   | Valore predefinito   |
| <code>inherit</code>                   | Eredita il valore del genitore   |

## Più CSS per l'immagine di sfondo

Questi attributi seguenti sono molto utili e quasi essenziali.

```
background-size:      xpx ypx | x% y%;
background-repeat:    no-repeat | repeat | repeat-x | repeat-y;
background-position:  left offset (px/%) right offset (px/%) | center center | left top | right
bottom;
```

## Gradienti di sfondo

I gradienti sono nuovi tipi di immagine, aggiunti in CSS3. Come immagine, le sfumature sono impostate con la proprietà `background-image`, o con la stenografia dello `background`.

Esistono due tipi di funzioni sfumate, lineari e radiali. Ogni tipo ha una variante non ripetitiva e una variante ripetitiva:

- `linear-gradient()`
- `repeating-linear-gradient()`
- `radial-gradient()`
- `repeating-radial-gradient()`

## lineare-gradiente ()

Un `linear-gradient` ha la seguente sintassi

```
background: linear-gradient( <direction>?, <color-stop-1>, <color-stop-2>, ...);
```

| Valore                               | Senso   |
|--------------------------------------|---|
| <code>&lt;direction&gt;</code>       | Potrebbe essere un argomento come <code>to top</code> , <code>to bottom</code> , <code>to right</code> o <code>to left</code> ; o un <b>angolo</b> come <code>0deg</code> , <code>90deg</code> .... L'angolo inizia dall'alto e ruota in senso orario. Può essere specificato in <b>deg</b> , <b>grad</b> , <b>rad</b> o <b>turn</b> . Se omissso, il gradiente scorre dall'alto verso il basso |
| <code>&lt;color-stop-list&gt;</code> | Elenco di colori, opzionalmente seguito ciascuno di una percentuale o <b>lunghezza</b> per visualizzarlo a. Ad esempio, <code>yellow 10%</code> , <code>rgba(0,0,0,.5) 40px</code> , <code>#fff 100%</code> ...   |

Ad esempio, questo crea un gradiente lineare che parte da destra e passa da rosso a blu

```
.linear-gradient {
  background: linear-gradient(to left, red, blue); /* you can also use 270deg */
}
```

Puoi creare un gradiente `diagonal` dichiarando una posizione di partenza sia orizzontale che verticale.

```
.diagonal-linear-gradient {
  background: linear-gradient(to left top, red, yellow 10%);
}
```

È possibile specificare qualsiasi numero di interruzioni di colore in una sfumatura separandole con virgole. I seguenti esempi creeranno una sfumatura con 8 stop di colore

```
.linear-gradient-rainbow {  
  background: linear-gradient(to left, red, orange, yellow, green, blue, indigo, violet)  
}
```

## radiale gradiente ()

```
.radial-gradient-simple {  
  background: radial-gradient(red, blue);  
}  
  
.radial-gradient {  
  background: radial-gradient(circle farthest-corner at top left, red, blue);  
}
```

| Valore          | Senso  |
|-----------------|--|
| circle          | Forma del gradiente I valori sono <code>circle</code> o <code>ellipse</code> , il valore predefinito è <code>ellipse</code> .  |
| farthest-corner | Parole chiave che descrivono quanto deve essere grande la forma finale. I valori sono il <code>closest-side farthest-side closest-corner</code> , il <code>farthest-side closest-corner farthest-corner</code> , il <code>closest-corner farthest-corner closest-corner farthest-corner</code> |
| top left        | Imposta la posizione del centro del gradiente, allo stesso modo <code>background-position</code> .   |

## Ripetendo le sfumature

La ripetizione delle funzioni del gradiente assume gli stessi argomenti degli esempi precedenti, ma affianca il gradiente sullo sfondo dell'elemento.

```
.bullseye {  
  background: repeating-radial-gradient(red, red 10%, white 10%, white 20%);  
}  
.warning {  
  background: repeating-linear-gradient(-45deg, yellow, yellow 10%, black 10%, black 20%);  
}
```

| Valore | Senso  |
|--------|--|
| -45deg | <b>Unità angolare</b> . L'angolo inizia dall'alto e ruota in senso orario. Può essere specificato in <code>deg</code> , <code>grad</code> , <code>rad</code> o <code>turn</code> . |

| Valore        | Senso  |
|---------------|--|
| to left       | Direzione del gradiente, il valore predefinito è <code>to bottom</code> . Sintassi: <code>to [y-axis(top OR bottom)] [x-axis(left OR right)]</code> cioè <code>to top right</code> |
| yellow<br>10% | Colore, opzionalmente seguito da una percentuale o lunghezza per visualizzarlo. Ripetuto due o più volte.  |

Si noti che possono essere usati i codici colore HEX, RGB, RGBA, HSL e HSLa al posto dei nomi dei colori. I nomi dei colori sono stati utilizzati per motivi illustrativi. Si noti inoltre che la sintassi del gradiente radiale è molto più complessa del gradiente lineare e qui viene mostrata una versione semplificata. Per una spiegazione completa e specifiche, vedere i [documenti MDN](#)

## Stenografia di base

La proprietà `background` può essere utilizzata per impostare una o più proprietà correlate allo sfondo:

| Valore                             | Descrizione  | CSS Ver. |
|------------------------------------|--|----------|
| <code>background-image</code>      | Immagine di sfondo da utilizzare   | 1+       |
| <code>background-color</code>      | Colore di sfondo da applicare  | 1+       |
| <code>background-position</code>   | Posizione dell'immagine di sfondo  | 1+       |
| <code>background-size</code>       | Dimensioni dell'immagine di sfondo   | 3+       |
| <code>background-repeat</code>     | Come ripetere l'immagine di sfondo   | 1+       |
| <code>background-origin</code>     | Come viene posizionato lo sfondo (ignorato quando è <code>fixed</code> <code>background-attachment</code> )  | 3+       |
| <code>background-clip</code>       | Come viene dipinto lo sfondo rispetto al <code>content-box</code> del <code>content-box</code> <code>border-box</code> o al <code>padding-box</code> | 3+       |
| <code>background-attachment</code> | Come si comporta l'immagine di sfondo, se scorre insieme al relativo blocco contenitore o ha una posizione fissa all'interno del viewport            | 1+       |
| <code>initial</code>               | Imposta la proprietà su valore come predefinito  | 3+       |
| <code>inherit</code>               | Eredita il valore della proprietà da padre   | 2+       |

L'ordine dei valori non ha importanza e ogni valore è facoltativo

# Sintassi

La sintassi della dichiarazione abbreviata di sfondo è:

```
background: [<background-image>] [<background-color>] [<background-position>]/[<background-size>] [<background-repeat>] [<background-origin>] [<background-clip>] [<background-attachment>] [<initial|inherit>];
```

## Esempi

```
background: red;
```

Semplicemente impostando un `background-color` con il valore `red`.

```
background: border-box red;
```

Impostazione di un `background-clip` di `background-color` un riquadro di `background-color` e un `background-color` di `background-color` rosso.

```
background: no-repeat center url("somepng.jpg");
```

Imposta una `background-repeat` non ripetizione, `background-origin` al centro e un'immagine `background-image` un'immagine.

```
background: url('pattern.png') green;
```

In questo esempio, il `background-color` di `background-color` dell'elemento sarebbe impostato su `green` con `pattern.png`, se disponibile, sovrapposto al colore, ripetendo tutte le volte necessarie per riempire l'elemento. Se `pattern.png` include qualsiasi trasparenza, il colore `green` sarà visibile dietro di esso.

```
background: #000000 url("picture.png") top left / 600px auto no-repeat;
```

In questo esempio abbiamo uno sfondo nero con un'immagine 'picture.png' in alto, l'immagine non si ripete in nessuno dei due assi ed è posizionata nell'angolo in alto a sinistra. Il / dopo la posizione deve essere in grado di includere la dimensione dell'immagine di sfondo che in questo caso è impostata come larghezza `600px` e auto per l'altezza. Questo esempio potrebbe funzionare bene con un'immagine caratteristica che può sfumare in un colore a tinta unita.

**NOTA:** L'uso della proprietà dello sfondo abbreviato reimposta tutti i valori delle proprietà dello sfondo precedentemente impostati, anche se non viene fornito un valore. Se si desidera solo modificare un valore della proprietà dello sfondo precedentemente impostato, utilizzare invece una proprietà longhand.

## Posizione di sfondo

La proprietà `background-position` viene utilizzata per specificare la posizione di partenza per un'immagine di sfondo o un gradiente

```
.myClass {  
  background-image: url('path/to/image.jpg');  
  background-position: 50% 50%;  
}
```

La posizione viene impostata utilizzando una coordinata **X** e **Y** e viene impostata utilizzando una qualsiasi delle unità utilizzate all'interno dei CSS.

| Unità                                | Descrizione  |
|--------------------------------------|--|
| <i>valore %</i><br><i>valore %</i>   | Una percentuale per l'offset orizzontale è relativa a ( <i>larghezza dell'area di posizionamento dello sfondo - larghezza dell'immagine di sfondo</i> ) .<br>Una percentuale per l'offset verticale è relativa a ( <i>altezza dell'area di posizionamento dello sfondo - altezza dell'immagine di sfondo</i> )<br>La dimensione dell'immagine è la dimensione data dalla dimensione dello <code>background-size</code> . |
| <i>valore px</i><br><i>valore px</i> | Offset l'immagine di sfondo di una lunghezza espressa in pixel rispetto all'angolo superiore sinistro dell'area di posizionamento dello sfondo   |

Le unità in CSS possono essere specificate con metodi diversi (vedi [qui](#) ).

## Proprietà posizione sfondo a mano lunga

Oltre alla proprietà abbreviata di cui sopra, è possibile utilizzare anche le proprietà di `background-position-x` longhand `background-position-x` e `background-position-y` . Questi ti permettono di controllare le posizioni x o y separatamente.

**NOTA:** questo è supportato in tutti i browser eccetto Firefox (versioni 31-48) <sup>2</sup> . Firefox 49, per essere rilasciato settembre 2016, *sosterrà* queste proprietà. Fino ad allora, [c'è una modifica di Firefox all'interno di questa risposta Stack Overflow](#).

### Allegato sfondo

La proprietà `background-attachment` imposta se un'immagine di sfondo è fissa o scorre con il resto della pagina.

```
body {  
  background-image: url('img.jpg');  
  background-attachment: fixed;  
}
```

| Valore    | Descrizione  |
|-----------|--|
| scorrere  | Lo sfondo scorre insieme all'elemento. Questo è il valore predefinito. |
| fisso     | Lo sfondo è fisso rispetto al viewport.                                |
| Locale    | Lo sfondo scorre insieme al contenuto dell'elemento.                   |
| iniziale  | Imposta questa proprietà sul valore predefinito.                       |
| ereditare | Eredita questa proprietà dal suo elemento padre                        |

## Esempi

### ***background-attachment: scroll***

Il comportamento predefinito, quando il corpo scorre, lo sfondo scorre con esso:

```
body {  
  background-image: url('image.jpg');  
  background-attachment: scroll;  
}
```

### ***background-attachment: fixed***

L'immagine di sfondo verrà fissata e non si sposterà quando il corpo viene scostato:

```
body {  
  background-image: url('image.jpg');  
  background-attachment: fixed;  
}
```

### ***background-attachment: local***

L'immagine di sfondo del div scorre quando il contenuto del div viene fatto scorrere.

```
div {  
  background-image: url('image.jpg');  
  background-attachment: local;  
}
```

## Ripeti lo sfondo

La proprietà background-repeat imposta se / come verrà ripetuta un'immagine di sfondo.

Per impostazione predefinita, un'immagine di sfondo viene ripetuta sia verticalmente che orizzontalmente.

```
div {
  background-image: url("img.jpg");
  background-repeat: repeat-y;
}
```

Ecco come una `background-repeat: repeat-y` assomiglia a:



## Colore di sfondo con opacità

Se imposti `opacity` su un elemento, questo avrà effetto su tutti i suoi elementi figli. Per impostare un'opacità solo sullo sfondo di un elemento, dovrai utilizzare i colori RGBA. L'esempio seguente avrà uno sfondo nero con 0.6 opacità.

```
/* Fallback for web browsers that don't support RGBA */
background-color: rgb(0, 0, 0);

/* RGBA with 0.6 opacity */
background-color: rgba(0, 0, 0, 0.6);

/* For IE 5.5 - 7*/
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr=#99000000,
endColorstr=#99000000);

/* For IE 8*/
-ms-filter: "progid:DXImageTransform.Microsoft.gradient(startColorstr=#99000000,
endColorstr=#99000000)";
```

## Immagine di sfondo multipla

Nei CSS3, possiamo impilare più sfondi nello stesso elemento.

```
#mydiv {
  background-image: url(img_1.png), /* top image */
                  url(img_2.png), /* middle image */
                  url(img_3.png); /* bottom image */
  background-position: right bottom,
                    left top,
                    right top;
  background-repeat: no-repeat,
                  repeat,
                  no-repeat;
}
```

Le immagini saranno impilate una sopra l'altra con il primo sfondo in alto e l'ultimo sfondo nella parte posteriore. `img_1` sarà in cima, `img_2` e `img_3` sono in fondo.

Possiamo anche usare la proprietà abbreviata di sfondo per questo:

```
#mydiv {
  background: url(img_1.png) right bottom no-repeat,
            url(img_2.png) left top repeat,
            url(img_3.png) right top no-repeat;
}
```

Possiamo anche impilare immagini e gradienti:

```
#mydiv {
  background: url(image.png) right bottom no-repeat,
            linear-gradient(to bottom, #fff 0%, #000 100%);
}
```

- [dimostrazione](#)

## La proprietà background-origin

La proprietà `background-origin` specifica dove è posizionata l'immagine di sfondo.

Nota: se la proprietà `background-attachment` è impostata su `fixed`, questa proprietà non ha alcun effetto.

Valore predefinito: `padding-box`

Valori possibili:

- `padding-box` - La posizione è relativa alla casella padding
- `border-box` - La posizione è relativa alla casella di confine
- `content-box` : la posizione è relativa alla casella del contenuto
- `initial`
- `inherit`

## CSS

```
.example {
  width: 300px;
  border: 20px solid black;
  padding: 50px;
  background: url(https://static.pexels.com/photos/6440/magazines-desk-work-workspace-medium.jpg);
  background-repeat: no-repeat;
}

.example1 {}

.example2 { background-origin: border-box; }

.example3 { background-origin: content-box; }
```

## HTML

```
<p>No background-origin (padding-box is default):</p>

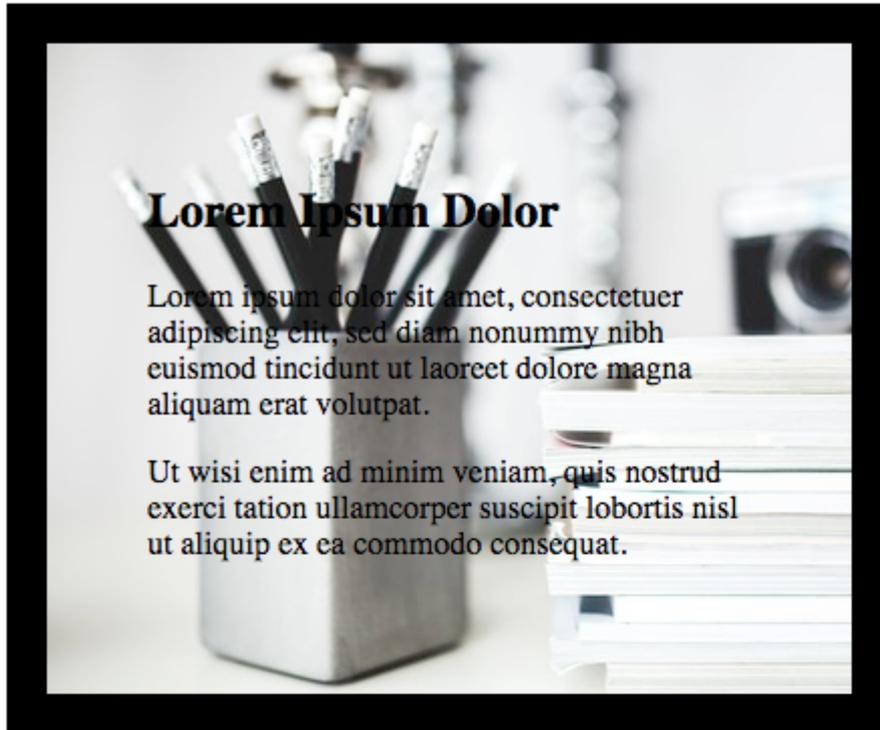
<div class="example example1">
  <h2>Lorem Ipsum Dolor</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
  <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
</div>

<p>background-origin: border-box:</p>
<div class="example example2">
  <h2>Lorem Ipsum Dolor</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
  <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
</div>

<p>background-origin: content-box:</p>
<div class="example example3">
  <h2>Lorem Ipsum Dolor</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
  <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
</div>
```

## Risultato:

No background-origin (padding-box is default):



background-origin: border-box:



background-origin: content-box:



è il valore predefinito. Ciò consente allo sfondo di estendersi fino al bordo esterno del bordo dell'elemento.

- `padding-box` ritaglia lo sfondo sul bordo esterno del padding dell'elemento e non lo lascia estendere nel bordo;
- `content-box` ritaglia lo sfondo sul bordo della casella del contenuto.
- `inherit` applica l'impostazione del genitore all'elemento selezionato.

## CSS

```
.example {
  width: 300px;
  border: 20px solid black;
  padding: 50px;
  background: url(https://static.pexels.com/photos/6440/magazines-desk-work-workspace-medium.jpg);
  background-repeat: no-repeat;
}

.example1 {}

.example2 { background-origin: border-box; }

.example3 { background-origin: content-box; }
```

## HTML

```
<p>No background-origin (padding-box is default):</p>

<div class="example example1">
  <h2>Lorem Ipsum Dolor</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
  <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
</div>

<p>background-origin: border-box:</p>
<div class="example example2">
  <h2>Lorem Ipsum Dolor</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
  <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
</div>

<p>background-origin: content-box:</p>
<div class="example example3">
  <h2>Lorem Ipsum Dolor</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
  <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
</div>
```

## Dimensione dello sfondo

---

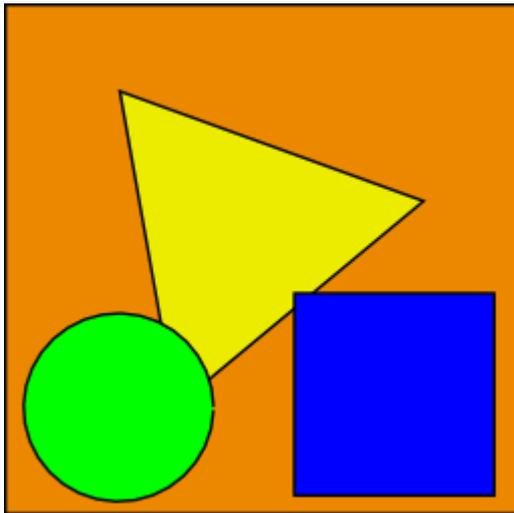
# Panoramica generale

La proprietà `background-size` consente di controllare il ridimensionamento `background-image`. Richiede fino a due valori, che determinano la scala / dimensione dell'immagine risultante in direzione verticale e orizzontale. Se la proprietà è mancante, è considerata `auto` sia in `width` che in `height`.

`auto` manterrà le proporzioni dell'immagine, se può essere determinata. L'altezza è facoltativa e può essere considerata `auto`. Pertanto, su un'immagine di 256 px x 256 px, tutte le seguenti impostazioni della `background-size` dello `background-size` produrranno un'immagine con altezza e larghezza di 50 px:

```
background-size: 50px;  
background-size: 50px auto; /* same as above */  
background-size: auto 50px;  
background-size: 50px 50px;
```

Quindi, se iniziamo con la seguente immagine (che ha le dimensioni menzionate di 256 px x 256 px),



finiremo con un 50 px x 50 px sullo schermo dell'utente, contenuto nello sfondo del nostro elemento:

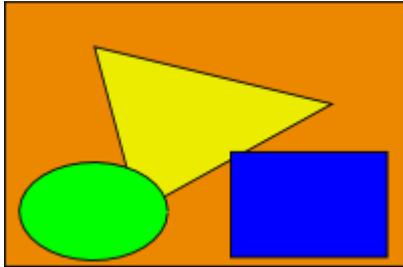


Si può anche usare valori percentuali per ridimensionare l'immagine rispetto all'elemento. L'esempio seguente produrrebbe un'immagine disegnata di 200 px x 133 px:

```
#withbackground {  
  background-image: url(to/some/background.png);  
  
  background-size: 100% 66%;  
  
  width: 200px;
```

```
height: 200px;

padding: 0;
margin: 0;
}
```



Il comportamento dipende `background-origin` dello `background-origin`.

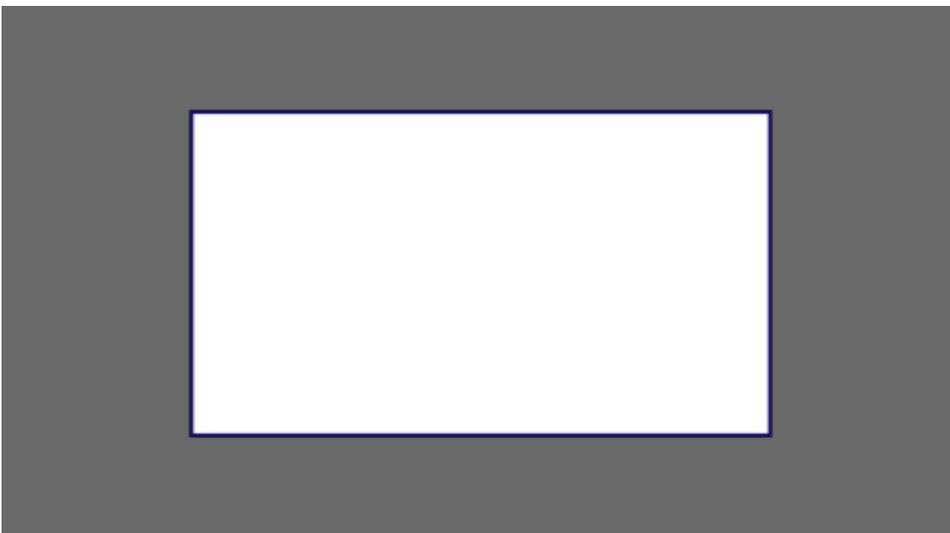
## Mantenere le proporzioni

L'ultimo esempio nella sezione precedente ha perso le proporzioni originali. Il cerchio è entrato in un'ellisse, il quadrato in un rettangolo, il triangolo in un altro triangolo.

L'approccio di lunghezza o percentuale non è abbastanza flessibile da mantenere il rapporto aspetto in ogni momento. `auto` non aiuta, dal momento che potresti non sapere quale dimensione del tuo elemento sarà più grande. Tuttavia, per coprire completamente determinate aree con un'immagine (e il rapporto di aspetto corretto) o per contenere un'immagine con proporzioni corrette completamente in un'area di sfondo, i valori, `contain` e `cover` forniscono la funzionalità aggiuntiva.

### Eggsplanation per `contain` e `cover`

Scusa per il cattivo gioco [di parole](#), ma useremo una [foto del giorno di Biswarup Ganguly](#) per la dimostrazione. Diciamo che questo è il tuo schermo, e l'area grigia è al di fuori del tuo schermo visibile. Per la dimostrazione, assumeremo un rapporto 16 x 9.



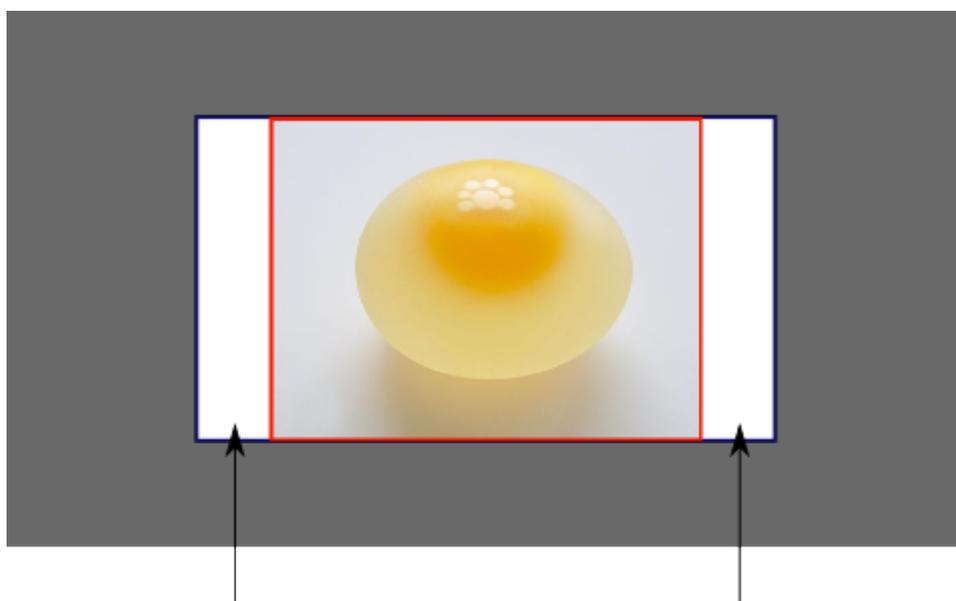
Vogliamo usare come sfondo la foto sopra riportata del giorno. Tuttavia, abbiamo ritagliato l'immagine su 4x3 per qualche motivo. Potremmo impostare la proprietà della `background-size` dello `background-size` una lunghezza fissa, ma ci concentreremo sul `contain` e sulla `cover`. Nota anche che presumo che non abbiamo manipolato la larghezza e / o l'altezza del `body`.

`contain`

```
contain
```

Ridimensiona l'immagine, pur mantenendo il suo rapporto di aspetto intrinseco (se presente), alla dimensione più grande in modo tale che sia la sua larghezza che la sua altezza possano adattarsi all'interno dell'area di posizionamento dello sfondo.

Ciò garantisce che l'immagine di sfondo sia sempre completamente contenuta nell'area di posizionamento dello sfondo, tuttavia in questo caso potrebbe esserci dello spazio vuoto riempito con il `background-color`:



`cover`

```
cover
```

Ridimensiona l'immagine, pur mantenendo il suo rapporto di aspetto intrinseco (se presente), alla dimensione più piccola in modo tale che sia la sua larghezza che la sua altezza possano coprire completamente l'area di posizionamento dello sfondo.

Questo assicura che l'immagine di sfondo copra tutto. Non ci sarà alcun `background-color` visibile, tuttavia, a seconda del rapporto dello schermo, una grande parte dell'immagine potrebbe essere tagliata:



## Dimostrazione con codice reale

```
div > div {
  background-image: url(http://i.stack.imgur.com/r5CAq.jpg);
  background-repeat: no-repeat;
  background-position: center center;
  background-color: #ccc;
  border: 1px solid;
  width: 20em;
  height: 10em;
}
div.contain {
  background-size: contain;
}
div.cover {
  background-size: cover;
}
/*****
Additional styles for the explanation boxes
*****/

div > div {
  margin: 0 1ex 1ex 0;
  float: left;
}
div + div {
  clear: both;
  border-top: 1px dashed silver;
  padding-top: 1ex;
}
div > div::after {
  background-color: #000;
  color: #fefefe;
  margin: 1ex;
  padding: 1ex;
  opacity: 0.8;
  display: block;
  width: 10ex;
  font-size: 0.7em;
  content: attr(class);
}
```

```
<div>
  <div class="contain"></div>
  <p>Note the grey background. The image does not cover the whole region, but it's fully
<em>contained</em>.
  </p>
</div>
<div>
  <div class="cover"></div>
  <p>Note the ducks/geese at the bottom of the image. Most of the water is cut, as well as a
part of the sky. You don't see the complete image anymore, but neither do you see any
background color; the image <em>covers</em> all of the <code>&lt;div&gt;</code>.</p>
</div>
```



Note the grey background. The image does not cover the whole region, but it's fully *contained*.



Note the ducks/geese at the bottom of the image. Most of the water is cut, as well as a part of the sky. You don't see the complete image anymore, but neither do you see any background color; the image *covers* all of the <code><div></code>.

## Proprietà background-blend-mode

```
.my-div {
  width: 300px;
  height: 200px;
  background-size: 100%;
  background-repeat: no-repeat;
  background-image: linear-gradient(to right, black 0%,white 100%),
url('https://static.pexels.com/photos/54624/strawberry-fruit-red-sweet-54624-medium.jpeg');
  background-blend-mode:saturation;
}
```

```
<div class="my-div">Lorem ipsum</div>
```

Vedi il risultato qui: <https://jsfiddle.net/MadalinaTn/y69d28Lb/>

Sintassi CSS: background-blend-mode: normal | moltiplicare | schermo | sovrapposizione | oscurare | alleggerire | colore-schivata | saturazione | colore | luminosità;

Leggi sfondi online: <https://riptutorial.com/it/css/topic/296/sfondi>

# Capitolo 48: straripamento

## Sintassi

- overflow: visibile | nascosto | scorrere | auto | iniziale | ereditare;

## Parametri

| Valore di Overflow | Dettagli   |
|--------------------|--|
| visible            | Mostra tutto il contenuto in eccesso all'esterno dell'elemento   |
| scroll             | Nasconde il contenuto che trabocca e aggiunge una barra di scorrimento   |
| hidden             | Nasconde il contenuto che trabocca, entrambe le barre di scorrimento scompaiono e la pagina diventa fissa              |
| auto               | Uguale a <code>scroll</code> se il contenuto fuoriesce, ma non aggiunge barra di scorrimento se il contenuto si adatta |
| inherit            | Eredita il valore dell'elemento genitore per questa proprietà  |

## Osservazioni

La proprietà di `overflow` specifica se ritagliare il contenuto, eseguire il rendering barre di scorrimento o estendere un contenitore per visualizzare il contenuto quando questo esplose nel suo contenitore a livello di blocco.

Quando un elemento è troppo piccolo per visualizzarne il contenuto, cosa succede? Per impostazione predefinita, il contenuto si sovrapporrà e verrà visualizzato all'esterno dell'elemento. Questo fa sembrare il tuo lavoro pessimo. Volete che il vostro lavoro abbia un bell'aspetto, quindi impostate la proprietà di `overflow` per gestire il contenuto in eccesso in un modo desiderabile.

I valori per la proprietà di `overflow` sono identici a quelli per le proprietà `overflow-x` e `overflow-y`, eccetto che si applicano lungo ciascun asse

La proprietà `overflow-wrap` è stata anche conosciuta come la proprietà `word-wrap`.

Nota importante: l' [utilizzo della proprietà di overflow con un valore diverso da visibile creerà un nuovo contesto di formattazione dei blocchi.](#)

## Examples

## overflow: scorrere

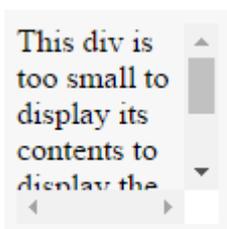
### HTML

```
<div>
  This div is too small to display its contents to display the effects of the overflow
  property.
</div>
```

### CSS

```
div {
  width:100px;
  height:100px;
  overflow:scroll;
}
```

### Risultato



Il contenuto sopra è ritagliato in una scatola da 100 px per 100 px, con lo scrolling disponibile per visualizzare il contenuto traboccante.

La maggior parte dei browser desktop visualizza barre di scorrimento orizzontali e verticali, indipendentemente dal fatto che il contenuto sia tagliato o meno. Questo può evitare problemi con le barre di scorrimento che appaiono e scompaiono in un ambiente dinamico. Le stampanti possono stampare contenuti in eccesso.

### trabocco-wrap

`overflow-wrap` dice a un browser che può spezzare una riga di testo all'interno di un elemento mirato su più linee in un posto altrimenti inviolabile. Utile nell'impedire una lunga stringa di testo che causa problemi di layout a causa del sovraccarico del suo contenitore.

### CSS

```
div {
  width:100px;
  outline: 1px dashed #bbb;
}

#div1 {
  overflow-wrap:normal;
}

#div2 {
```

```
overflow-wrap:break-word;
}
```

## HTML

```
<div id="div1">
  <strong>#div1</strong>: Small words are displayed normally, but a long word like <span
style="red;">supercalifragilisticexpialidocious</span> is too long so it will overflow past
the edge of the line-break
</div>
```

```
<div id="div2">
  <strong>#div2</strong>: Small words are displayed normally, but a long word like <span
style="red;">supercalifragilisticexpialidocious</span> will be split at the line break and
continue on the next line.
</div>
```

**#div1:** Small words are displayed normally, but a long word like **supercalifragilisticexpialidoc:** is too long so it will overflow past the edge of the line-break

**#div2:** Small words are displayed normally, but a long word like **supercalifragilisticexpialidocious** will be split at the line break and continue on the next line.

| overflow-wrap - Value | Dettagli  |
|-----------------------|---|
| normal                | Lascia una parola in overflow se è più lunga della linea      |
| break-word            | Dividerà una parola in più righe, se necessario               |
| inherit               | Eredita il valore dell'elemento genitore per questa proprietà |

### overflow: visibile

## HTML

```
<div>
  Even if this div is too small to display its contents, the content is not clipped.
</div>
```

## CSS

```
div {
  width:50px;
  height:50px;
  overflow:visible;
}
```

## Risultato



Il contenuto non viene ritagliato e verrà visualizzato all'esterno della casella del contenuto se supera la dimensione del contenitore.

## Contesto di formattazione dei blocchi creato con overflow

L'utilizzo della proprietà di `overflow` con un valore diverso da `visible` creerà un nuovo **contesto di formattazione dei blocchi** . Questo è utile per allineare un elemento di blocco accanto a un elemento flottato.

## CSS

```
img {
  float:left;
  margin-right: 10px;
}
div {
  overflow:hidden; /* creates block formatting context */
}
```

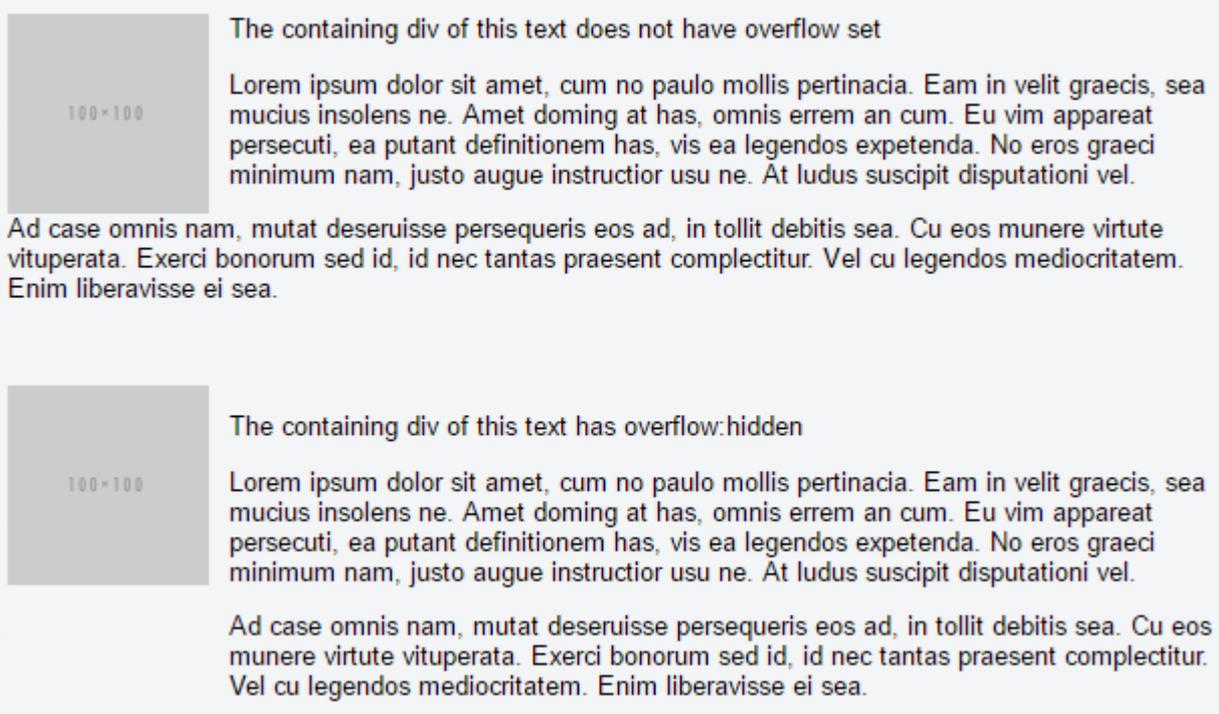
## HTML

```

```

```
<div>
  <p>Lorem ipsum dolor sit amet, cum no paulo mollis pertinacia.</p>
  <p>Ad case omnis nam, mutat deseruisse persequeris eos ad, in tollit debitis sea.</p>
</div>
```

## Risultato



The containing div of this text does not have overflow set

100x100

Lorem ipsum dolor sit amet, cum no paulo mollis pertinacia. Eam in velit gaecis, sea mucius insolens ne. Amet doming at has, omnis errem an cum. Eu vim appareat persecuti, ea putant definitionem has, vis ea legendos expetenda. No eros graeci minimum nam, justo augue instructor usu ne. At ludus suscipit disputationi vel.

Ad case omnis nam, mutat deseruisse persequeris eos ad, in tollit debitis sea. Cu eos munere virtute vituperata. Exerci bonorum sed id, id nec tantas praesent complectitur. Vel cu legendos mediocritatem. Enim liberavisse ei sea.

The containing div of this text has overflow:hidden

100x100

Lorem ipsum dolor sit amet, cum no paulo mollis pertinacia. Eam in velit gaecis, sea mucius insolens ne. Amet doming at has, omnis errem an cum. Eu vim appareat persecuti, ea putant definitionem has, vis ea legendos expetenda. No eros graeci minimum nam, justo augue instructor usu ne. At ludus suscipit disputationi vel.

Ad case omnis nam, mutat deseruisse persequeris eos ad, in tollit debitis sea. Cu eos munere virtute vituperata. Exerci bonorum sed id, id nec tantas praesent complectitur. Vel cu legendos mediocritatem. Enim liberavisse ei sea.

Questo esempio mostra come i paragrafi all'interno di un div con il set di proprietà di `overflow` interagiranno con un'immagine fluttuante.

## overflow-x e overflow-y

Queste due proprietà funzionano in modo simile alla proprietà di `overflow` e accettano gli stessi valori. Il parametro `overflow-x` funziona solo sull'asse x o da sinistra a destra. L' `overflow-y` funziona sull'asse y o dall'alto verso il basso.

## HTML

```
<div id="div-x">
  If this div is too small to display its contents,
  the content to the left and right will be clipped.
</div>

<div id="div-y">
  If this div is too small to display its contents,
  the content to the top and bottom will be clipped.
</div>
```

## CSS

```
div {
  width: 200px;
```

```
    height: 200px;
}

#div-x {
    overflow-x: hidden;
}

#div-y {
    overflow-y: hidden;
}
```

Leggi straripamento online: <https://riptutorial.com/it/css/topic/4947/straripamento>

---

# Capitolo 49: Struttura e formattazione di una regola CSS

## Osservazioni

Per facilitare la leggibilità, mantenere tutte le dichiarazioni rientrate di un livello dal selettore e la parentesi graffa di chiusura sulla propria riga. Aggiungi uno spazio singolo dopo selettori e due punti e posiziona sempre un punto e virgola dopo la dichiarazione finale.

---

## Buono

```
p {
  color: maroon;
  font-size: 16px;
}
```

---

## Male

```
p{
  color: maroon;
  font-size:16px }
```

---

## One-Liner

Se ci sono solo una o due dichiarazioni, *potresti* farla franca con questa. Non raccomandato per la maggior parte dei casi. Sii sempre coerente quando possibile.

```
p { color: maroon; font-size: 16px; }
```

---

## Examples

### Regole, selettori e blocchi di dichiarazione

Una **regola** CSS è composta da un **selettore** (ad es. `h1`) e un **blocco di dichiarazione** (`{ }`).

```
h1 { }
```

### Elenchi di proprietà

Alcune proprietà possono assumere più valori, noti collettivamente come un **elenco di proprietà**.

```
/* Two values in this property list */
span {
  text-shadow: yellow 0 0 3px, green 4px 4px 10px;
}

/* Alternate Formatting */
span {
  text-shadow:
    yellow 0 0 3px,
    green 4px 4px 10px;
}
```

## Selettori multipli

Quando si raggruppano selettori CSS, si applicano gli stessi stili a diversi elementi diversi senza ripetere gli stili nel foglio di stile. Utilizzare una virgola per separare più selettori raggruppati.

```
div, p { color: blue }
```

Quindi il colore blu si applica a tutti gli elementi `<div>` e tutti gli elementi `<p>`. Senza la virgola solo gli elementi `<p>` che sono figli di un `<div>` sarebbero rossi.

Questo vale anche per tutti i tipi di selettori.

```
p, .blue, #first, div span{ color : blue }
```

Questa regola si applica a:

- `<p>`
- elementi della classe `blue`
- elemento con l'ID `first`
- ogni `<span>` all'interno di un `<div>`

Leggi [Struttura e formattazione di una regola CSS online](https://riptutorial.com/it/css/topic/4313/struttura-e-formattazione-di-una-regola-css):

<https://riptutorial.com/it/css/topic/4313/struttura-e-formattazione-di-una-regola-css>

# Capitolo 50: Supporto browser e prefissi

## Parametri

| Prefisso       | Browser (s)   |
|----------------|---|
| -webkit-       | Google Chrome, Safari, versioni più recenti di Opera 12 e versioni successive, Android, Blackberry e UC |
| -moz-          | Mozilla Firefox   |
| -ms-           | Internet Explorer, Edge   |
| -o- , -<br>xv- | Opera fino alla versione 12   |
| -khtml-        | Konquerer   |

## Osservazioni

I prefissi dei fornitori vengono utilizzati per consentire il supporto dell'anteprima per le nuove funzionalità CSS in cui la funzionalità non è ancora raccomandata dalle specifiche.

**Si consiglia di non utilizzare i prefissi dei fornitori negli ambienti di produzione.**

Questi prefissi esistono per testare nuove funzionalità non ancora finalizzate e il comportamento è intrinsecamente inaspettato. Semplicemente usando i prefissi non si garantisce il supporto del browser per i vecchi browser, poiché non si può garantire che la funzione non sia cambiata nel tempo per avere prestazioni diverse, e *potrebbe* comunque essere interrotta in quei vecchi browser che si sostiene di supportare.

Se il supporto dei browser più vecchi è importante, dovresti invece considerare l'utilizzo di JavaScript o altre soluzioni per imitare gli effetti e garantire veramente il supporto per i vecchi browser.

I browser utilizzeranno i loro prefissi e ignoreranno le proprietà che non capiscono.

**NOTA** : i prefissi dovrebbero sempre apparire prima della sintassi ufficiale, non prefissa. Altrimenti verrebbero sovrascritte con le proprietà prefissate, che alla fine potrebbero essere un'altra implementazione.

Se un browser supporta sia una versione non prefissata sia una versione prefissata di una proprietà, la proprietà più recente da dichiarare avrà la precedenza.

## Examples

## transizioni

```
div {  
  -webkit-transition: all 4s ease;  
  -moz-transition: all 4s ease;  
  -o-transition: all 4s ease;  
  transition: all 4s ease;  
}
```

## Trasformare

```
div {  
  -webkit-transform: rotate(45deg);  
  -moz-transform: rotate(45deg);  
  -ms-transform: rotate(45deg);  
  -o-transform: rotate(45deg);  
  transform: rotate(45deg);  
}
```

Leggi **Supporto browser e prefissi online**: <https://riptutorial.com/it/css/topic/1138/supporto-browser-e-prefissi>

# Capitolo 51: tabelle

## Sintassi

- disposizione della tabella: *auto* | fisso;
- border-collapse: *separato* | crollo;
- interlinea: <lunghezza> | <lunghezza> <lunghezza>;
- celle vuote: *mostra* | nascondere;
- didascalia: in *alto* | parte inferiore;

## Osservazioni

Queste proprietà si applicano sia a `<table>` elements (\*) che a elementi HTML visualizzati come `display: table` o `display: inline-table`

(\*) Gli elementi `<table>` sono ovviamente disegnati in modo nativo da UA / browser come `display: table`

Le tabelle HTML sono semanticamente valide per i dati tabulari. Non è consigliabile utilizzare tabelle per il layout. Invece, usa CSS.

## Examples

### table-layout

La proprietà di `table-layout` modifica l'algoritmo utilizzato per il layout di una tabella.

Sotto un esempio di due tabelle entrambe impostate in `width: 150px`:

| First name | Last name | Homeworld |
|------------|-----------|-----------|
| Luke       | Skywalker | Tatooine  |
| Leia       | Organa    | Alderaan  |

| First name | Last name | Homeworld |
|------------|-----------|-----------|
| Luke       | Skywalker | Tatooine  |
| Leia       | Organa    | Alderaan  |

La tabella a sinistra ha il `table-layout: auto` mentre quella a destra ha il `table-layout: fixed`. Il primo è più largo della larghezza specificata (210px invece di 150px) ma i contenuti si adattano. Quest'ultimo prende la larghezza definita di 150 px, indipendentemente dal fatto che il contenuto esca o meno.

| Valore      | Descrizione   |
|-------------|---|
| <i>auto</i> | Questo è il valore predefinito. Definisce il layout della tabella per essere determinato dal contenuto delle sue 'celle'. |
| fisso       | Questo valore imposta il layout della tabella in base alla proprietà <code>width</code> fornita alla                      |

| Valore | Descrizione   |
|--------|---|
|        | tabella. Se il contenuto di una cella supera questa larghezza, la cella non verrà ridimensionata ma, al contrario, il contenuto si sovrapporrà. |

## border-collapse

La proprietà `border-collapse` determina se i bordi di una tabella devono essere separati o uniti.

Di seguito un esempio di due tabelle con valori diversi per la proprietà `border-collapse` :

| First name | Last name | Homeworld |
|------------|-----------|-----------|
| Luke       | Skywalker | Tatooine  |
| Leia       | Organa    | Alderaan  |

| First name | Last name | Homeworld |
|------------|-----------|-----------|
| Luke       | Skywalker | Tatooine  |
| Leia       | Organa    | Alderaan  |

La tabella a sinistra ha un `border-collapse: separate` mentre quello a destra ha `border-collapse: collapse`.

| Valore   | Descrizione   |
|----------|---|
| separato | Questo è il valore predefinito. Rende i bordi del tavolo separati gli uni dagli altri.  |
| crolla   | Questo valore imposta i bordi della tabella per unire insieme, anziché essere distinti. |

## border-spacing

La proprietà di `border-spacing` dei `border-spacing` determina la spaziatura tra le celle. Ciò non ha alcun effetto a meno che il `border-collapse` sia impostato per `separate`.

Di seguito un esempio di due tabelle con valori diversi per la proprietà `border-spacing` :

| First name | Last name | Homeworld |
|------------|-----------|-----------|
| Luke       | Skywalker | Tatooine  |
| Leia       | Organa    | Alderaan  |

| First name | Last name | Homeworld |
|------------|-----------|-----------|
| Luke       | Skywalker | Tatooine  |
| Leia       | Organa    | Alderaan  |

La tabella a sinistra ha `border-spacing: 2px` (predefinito) mentre quella a destra ha `border-spacing: 8px`.

| Valore                         | Descrizione   |
|--------------------------------|---|
| <code>&lt;length&gt;</code>    | Questo è il comportamento predefinito, anche se il valore esatto può variare tra i browser. |
| <code>&lt;lunghezza&gt;</code> | Questa sintassi consente di specificare rispettivamente valori                              |

| Valore      | Descrizione                       |
|-------------|-----------------------------------|
| <lunghezza> | orizzontali e verticali separati. |

## empty-cells

La proprietà `empty-cells` determina se le celle senza contenuto devono essere visualizzate o meno. Ciò non ha alcun effetto a meno che il `border-collapse` sia impostato per `separate`.

Sotto un esempio con due tabelle con valori diversi impostati sulla proprietà `empty-cells`:

| First name | Last name | Homeworld |
|------------|-----------|-----------|
| Luke       |           | Tatooine  |
| Leia       | Organa    |           |

| First name | Last name | Homeworld |
|------------|-----------|-----------|
| Luke       |           | Tatooine  |
| Leia       | Organa    |           |

La tabella a sinistra ha `empty-cells: show` mentre quella a destra ha `empty-cells: hide`. Il primo mostra le celle vuote mentre il secondo no.

| Valore          | Descrizione  |
|-----------------|--|
| <i>mostrare</i> | Questo è il valore predefinito. Mostra le celle anche se sono vuote.             |
| nascondere      | Questo valore nasconde una cella del tutto se non ci sono contenuti nella cella. |

Maggiori informazioni:

- <https://www.w3.org/TR/CSS21/tables.html#empty-cells>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/empty-cells>
- <http://codepen.io/SitePoint/pen/yfhtq>
- <https://css-tricks.com/almanac/properties/e/empty-cells/>

## caption-side

La proprietà del `caption-side` determina il posizionamento verticale dell'elemento `<caption>` all'interno di una tabella. Questo non ha alcun effetto se tale elemento non esiste.

Sotto un esempio con due tabelle con valori diversi impostati sulla proprietà del `caption-side`:

| Star Wars figures |           |           |
|-------------------|-----------|-----------|
| First name        | Last name | Homeworld |
| Luke              | Skywalker | Tatooine  |
| Leia              | Organa    | Alderaan  |

| First name | Last name | Homeworld |
|------------|-----------|-----------|
| Luke       | Skywalker | Tatooine  |
| Leia       | Organa    | Alderaan  |

Star Wars figures

La tabella a sinistra ha `caption-side: top` mentre quella a destra ha `caption-side: bottom`.

| Valore           | Descrizione  |
|------------------|--|
| <i>superiore</i> | Questo è il valore predefinito. Posiziona la didascalia sopra il tavolo. |
| parte inferiore  | Questo valore posiziona la didascalia sotto la tabella.                  |

Leggi tabelle online: <https://riptutorial.com/it/css/topic/1074/tabelle>

# Capitolo 52: Tipografia

## Sintassi

- font: [ *font-style* ] [*font-variant*] [*font-weight*] *font-size* [/ *line-height*] *font-family* ;
- stile *font* : stile *font*
- font-variant: *font-variant*
- font-weight: *font-weight* ;
- font-size: *font-size* ;
- line-height: *line-height* ;
- font-family: *font-family* ;
- colore: *colore* ;
- citazioni: *nessuno* | *stringa* | *iniziale* | *eredita* ;
- font-stretch: *font-stretch* ;
- allineamento del testo : allineamento del testo ;
- text-indent: *length* | *initial* | *inherit* ;
- overflow del testo: *clip* | *ellipsis* | *string* | *initial* | *inherit* ;
- text-transform: *none* | *capitalize* | *uppercase* | *minuscolo* | *initial* | *inherit* ;
- text-shadow: *h-shadow v-shadow blur-radius color* | *none* | *initial* | *inherit* ;
- font-size-adjust: *numero* | *none* | *initial* | *inherit* ;
- font stretch: *ultra-condensato* | *extra-condensato* | *condensato* | *semi-condensato* | *normale* | *semi-espanso* | *espanso* | *extra-espanso* | *ultra-espanso* | *iniziale* | *ereditato* ;
- trattini: *nessuno* | *manuale* | *auto* ;
- tab-size: *numero* | *lunghezza* | *iniziale* | *eredita* ;
- letter-spacing: *normal* | *length* | *initial* | *inherit* ;
- spaziatura delle parole: *normale* | *lunghezza* | *iniziale* | *eredita* ;

## Parametri

| Parametro                  | Dettagli  |
|----------------------------|---|
| <i>stile del font</i>      | <i>italics</i> <b>O</b> <i>oblique</i>  |
| <i>font-variant</i>        | <i>normal</i> <b>O</b> <i>small-caps</i>  |
| <i>font-weight</i>         | <i>normal</i> , <b>bold</b> <b>O</b> numerico da 100 a 900.                             |
| <i>dimensione del font</i> | La dimensione del carattere indicata in % , px , em o qualsiasi altra misura CSS valida |
| <i>altezza della linea</i> | L'altezza della linea indicata in % , px , em o qualsiasi altra misura CSS valida       |
| <i>famiglia di font</i>    | Questo è per la definizione del nome della famiglia.                                    |

| Parametro              | Dettagli   |
|------------------------|--|
| <i>colore</i>          | Qualsiasi <a href="#">rappresentazione del colore CSS</a> valida, come <code>red</code> , <code>#00FF00</code> , <code>hsl(240, 100%, 50%)</code> ecc.   |
| <i>font-stretch</i>    | Se utilizzare o meno una faccia confusa o espansa dal carattere. I valori validi sono <code>normal</code> , <code>ultra-condensed</code> , <code>extra-condensed</code> , <code>condensed</code> , <code>semi-condensed</code> , <code>semi-expanded</code> , <code>expanded</code> , <code>extra-expanded</code> <b>O</b> <code>ultra-expanded</code> |
| <i>text-align</i>      | <code>start</code> , <code>end</code> , <code>left</code> , <code>right</code> , <code>center</code> , <code>justify</code> , <code>match-parent</code>  |
| <i>text-decoration</i> | <code>none</code> , <code>underline</code> , <code>overline</code> , <code>line-through</code> , <code>initial</code> , <code>inherit</code> ;   |

## Osservazioni

- La proprietà `text-shadow` non è supportata dalle versioni di Internet Explorer meno di 10.

## Examples

### Dimensione del font

#### HTML:

```
<div id="element-one">Hello I am some text.</div>
<div id="element-two">Hello I am some smaller text.</div>
```

#### CSS:

```
#element-one {
  font-size: 30px;
}

#element-two {
  font-size: 10px;
}
```

Il testo all'interno di `#element-one` avrà `#element-one` dimensione di `30px` , mentre il testo in `#element-two` avrà una dimensione di `10px` .

### La stenografia del font

Con la sintassi:

```
element {
  font: [font-style] [font-variant] [font-weight] [font-size/line-height] [font-family];
}
```

Puoi avere tutti gli stili relativi ai font in una dichiarazione con la stenografia del `font` . Basta usare la proprietà del `font` e inserire i valori nell'ordine corretto.

Ad esempio, per rendere tutti gli elementi `p` grassetto con una dimensione del font di 20px e usando Arial come tipografia, tipicamente lo si codificherebbe come segue:

```
p {
  font-weight: bold;
  font-size: 20px;
  font-family: Arial, sans-serif;
}
```

Tuttavia con la stenografia dei font può essere sintetizzata come segue:

```
p {
  font: bold 20px Arial, sans-serif;
}
```

**Nota** : dal momento che `font-style` , `font-variant` , `font-weight` e `line-height` sono opzionali, in questo esempio vengono saltati tre di questi. È importante notare che l'utilizzo della scorciatoia **ripristina** gli altri attributi non dati. Un altro punto importante è che i due attributi necessari per il collegamento ai font da utilizzare sono `font-size` e `font-family` . Se non sono entrambi inclusi, la scorciatoia viene ignorata.

Valore iniziale per ciascuna proprietà:

- `font-style: normal;`
- `font-variant: normal;`
- `font-weight: normal;`
- `font-stretch: normal;`
- `font-size: medium;`
- `line-height: normal;`
- `font-family` - dipende dal programma utente

## Stack di font

```
font-family: 'Segoe UI', Tahoma, sans-serif;
```

Il browser tenterà di applicare il carattere di carattere "Segoe UI" ai caratteri all'interno degli elementi scelti come target dalla proprietà precedente. Se questo font non è disponibile, o il font non contiene un glifo per il carattere richiesto, il browser tornerà a Tahoma e, se necessario, qualsiasi font sans-serif sul computer dell'utente. Nota che i nomi dei font con più di una parola come "Segoe UI" devono avere virgolette singole o doppie intorno a loro.

```
font-family: Consolas, 'Courier New', monospace;
```

Il browser tenterà di applicare il carattere "Consolas" ai caratteri all'interno degli elementi scelti come target dalla proprietà precedente. Se questo font non è disponibile, o il font non contiene un glifo per il carattere richiesto, il browser tornerà a "Courier New" e, se necessario, qualsiasi font

monospace sul computer dell'utente.

## Spaziatura del carattere

```
h2 {
  /* adds a 1px space horizontally between each letter;
  also known as tracking */
  letter-spacing: 1px;
}
```

La proprietà `letter-spacing` viene utilizzata per specificare lo spazio tra i caratteri in un testo.

! la spaziatura delle lettere supporta anche valori negativi:

```
p {
  letter-spacing: -1px;
}
```

Risorse: <https://developer.mozilla.org/en-US/docs/Web/CSS/letter-spacing>

## Trasformazione del testo

La proprietà `text-transform` consente di modificare la maiuscola del testo. I valori validi sono:

`uppercase` , `capitalize` , `lowercase` , `initial` , `inherit` e `none`

CSS:

```
.example1 {
  text-transform: uppercase;
}
.example2 {
  text-transform: capitalize;
}
.example3 {
  text-transform: lowercase;
}
```

## HTML

```
<p class="example1">
  all letters in uppercase <!-- "ALL LETTERS IN UPPERCASE" -->
</p>
<p class="example2">
  all letters in capitalize <!-- "All Letters In Capitalize (Sentence Case)" -->
</p>
<p class="example3">
  all letters in lowercase <!-- "all letters in lowercase" -->
</p>
```

## Indentatura del testo

```
p {
  text-indent: 50px;
}
```

La proprietà `text-indent` specifica quanto testo dello spazio orizzontale deve essere spostato prima dell'inizio della prima riga del contenuto testuale di un elemento.

risorse:

- [Indentazione solo della prima riga di testo in un paragrafo?](#)
- <https://www.w3.org/TR/CSS21/text.html#propdef-text-indent>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/text-indent>

## Decorazione del testo

La proprietà `text-decoration` viene utilizzata per impostare o rimuovere le decorazioni dal testo.

```
h1 { text-decoration: none; }
h2 { text-decoration: overline; }
h3 { text-decoration: line-through; }
h4 { text-decoration: underline; }
```

la decorazione del testo può essere utilizzata in combinazione con lo stile di decorazione del testo e il colore della decorazione del testo come proprietà di stenografia:

```
.title { text-decoration: underline dotted blue; }
```

Questa è una versione abbreviata di

```
.title {
  text-decoration-style: dotted;
  text-decoration-line: underline;
  text-decoration-color: blue;
}
```

Va notato che le seguenti proprietà sono supportate solo in Firefox

- `text-decoration-color`
- `text-decoration-line`
- `text-decoration-style`
- `testo-decorazione-salto`

## Overflow del testo

La proprietà `text-overflow` riguarda il modo in cui il contenuto in overflow deve essere segnalato agli utenti. In questo esempio, i `ellipsis` rappresentano il testo ritagliato.

```
.text {
  overflow: hidden;
  text-overflow: ellipsis;
}
```

```
}
```

Sfortunatamente, `text-overflow: ellipsis` funzionano solo su un'unica riga di testo. Non è possibile supportare i puntini di sospensione nell'ultima riga del CSS standard, ma è possibile ottenerli con l'implementazione di Flexbox non standard.

```
.giveMeEllipsis {
  overflow: hidden;
  text-overflow: ellipsis;
  display: -webkit-box;
  -webkit-box-orient: vertical;
  -webkit-line-clamp: N; /* number of lines to show */
  line-height: X;      /* fallback */
  max-height: X*N;    /* fallback */
}
```

Esempio (aperto in Chrome o Safari):

<http://jsfiddle.net/csYjC/1131/>

risorse:

<https://www.w3.org/TR/2012/WD-css3-ui-20120117/#text-overflow0>

## Word Spacing

La proprietà `word-spacing` specifica il comportamento di spaziatura tra tag e parole.

### Valori possibili

- una *lunghezza* positiva o negativa (usando `em` `px` `vh` `cm` ecc.) o *percentuale* (usando `%` )
- la parola chiave `normal` utilizza la spaziatura delle parole predefinita del font
- la parola chiave `inherit` prende il valore dall'elemento genitore

## CSS

```
.normal { word-spacing: normal; }
.narrow { word-spacing: -3px; }
.extensive { word-spacing: 10px; }
```

## HTML

```
<p>
  <span class="normal">This is an example, showing the effect of "word-spacing".</span><br>
  <span class="narrow">This is an example, showing the effect of "word-spacing".</span><br>
  <span class="extensive">This is an example, showing the effect of "word-spacing".</span><br>
</p>
```

## Online-Demo

[Prova tu stesso](#)

## Ulteriori letture:

- [word-spacing - MDN](#)
- [word-spacing - w3.org](#)

## Direzione del testo

```
div {
  direction: ltr; /* Default, text read from left-to-right */
}
.ex {
  direction: rtl; /* text read from right-to-left */
}
.horizontal-tb {
  writing-mode: horizontal-tb; /* Default, text read from left-to-right and top-to-bottom.
*/
}
.vertical-rtl {
  writing-mode: vertical-rl; /* text read from right-to-left and top-to-bottom */
}
.vertical-ltr {
  writing-mode: vertical-rl; /* text read from left-to-right and top to bottom */
}
```

La proprietà `direction` viene utilizzata per modificare la direzione orizzontale del testo di un elemento.

Sintassi: `direction: ltr | rtl | initial | inherit;`

---

La proprietà in `writing-mode` modifica l'allineamento del testo in modo che possa essere letto dall'alto verso il basso o da sinistra a destra, a seconda della lingua.

Sintassi: `direction: horizontal-tb | vertical-rl | vertical-lr;`

## Carattere Variante

attributi:

### ***normale***

Attributo predefinito dei caratteri.

### ***maiuscoletto***

Imposta ogni lettera in maiuscolo, **ma** rende le lettere minuscole (dal testo originale) di dimensioni inferiori rispetto alle lettere che in origine sono maiuscole.

**CSS:**

```
.smallcaps{
  font-variant: small-caps;
}
```

## HTML:

```
<p class="smallcaps">
  Documentation about CSS Fonts
  <br>
  aNd ExAmPlE
</p>
```

## USCITA:

DOCUMENTATION ABOUT CSS FONTS  
AND EXAMPLE

Nota: la proprietà `font-variant` è una scorciatoia per le proprietà: `font-variant-caps`, `font-variant-numeric`, `font-variant-alternates`, `font-variant-ligatures` e `font-variant-east-asian`.

## Citazioni

La proprietà `quotes` viene utilizzata per personalizzare le virgolette di apertura e chiusura del tag `<q>`.

```
q {
  quotes: "«" "»";
}
```

## Ombra di testo

Per aggiungere ombre al testo, usa la proprietà `text-shadow`. La sintassi è la seguente:

```
text-shadow: horizontal-offset vertical-offset blur color;
```

## Ombra senza raggio di sfocatura

```
h1 {
  text-shadow: 2px 2px #0000FF;
}
```

Questo crea un effetto ombra blu attorno a un'intestazione

## Ombra con il raggio di sfocatura

Per aggiungere un effetto sfocatura, aggiungi un'opzione argomento `blur radius`

```
h1 {
  text-shadow: 2px 2px 10px #0000FF;
}
```

## Ombre multiple

Per dare a un elemento più ombre, separale con virgole

```
h1 {  
  text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;  
}
```

Leggi Tipografia online: <https://riptutorial.com/it/css/topic/427/tipografia>

# Capitolo 53: transizioni

## Sintassi

- transizione: [transizione-proprietà] [transizione-durata] [transizione-temporizzazione-funzione] [transizione-ritardo];

## Parametri

| Parametro                   | Dettagli   |
|-----------------------------|--|
| transizione immobili        | La specifica proprietà CSS la cui variazione di valore deve essere modificata (o) <code>all</code> , se tutte le <a href="#">proprietà di transizione</a> devono essere trasferite.  |
| durata della transizione    | La durata (o il periodo) in secondi ( <code>s</code> ) o millisecondi ( <code>ms</code> ) su cui deve avvenire la transizione.   |
| transizione-timing-function | Una funzione che descrive come vengono calcolati i valori intermedi durante la transizione. I valori comunemente usati sono <code>ease</code> , <code>ease-in</code> , <code>ease-out</code> , <code>ease-in-out</code> , <code>linear</code> , <code>cubic-bezier()</code> , <code>steps()</code> . Maggiori informazioni sulle varie funzioni di temporizzazione possono essere trovate nelle <a href="#">specifiche del W3C</a> . |
| transizione ritardo         | La quantità di tempo che deve trascorrere prima che la transizione possa iniziare. Può essere specificato in secondi ( <code>s</code> ) o millisecondi ( <code>ms</code> )   |

## Osservazioni

Alcuni browser meno recenti supportano solo proprietà di `transition` [prefisso del fornitore](#) :

- `-webkit` : Chrome 25-, Safari 6, Safari e Chrome per iOS 6.1-, Android 4.3- Browser, Blackberry Browser 7-, UC Browser 9.9- per Android.
- `-moz` : Firefox 15-.
- `-o` : Opera 11.5-, Opera Mobile 12-.

Esempio:

```
-webkit-transition: all 1s;  
-moz-transition: all 1s;  
-o-transition: all 1s;  
transition: all 1s;
```

## Examples

## Stenografia di transizione

### CSS

```
div{
  width: 150px;
  height:150px;
  background-color: red;
  transition: background-color 1s;
}
div:hover{
  background-color: green;
}
```

### HTML

```
<div></div>
```

Questo esempio cambierà il colore di sfondo quando il div è sospeso sul colore di sfondo, il cambiamento durerà 1 secondo.

### Transizione (longhand)

### CSS

```
div {
  height: 100px;
  width: 100px;
  border: 1px solid;
  transition-property: height, width;
  transition-duration: 1s, 500ms;
  transition-timing-function: linear;
  transition-delay: 0s, 1s;
}
div:hover {
  height: 200px;
  width: 200px;
}
```

### HTML

```
<div></div>
```

- **transition-property** : specifica le proprietà CSS per cui l'effetto di transizione è valido. In questo caso, il div si espanderà sia orizzontalmente che verticalmente quando si libererà.
- **transition-duration** : specifica il tempo che una transizione impiega per completare. Nell'esempio sopra, le transizioni di altezza e larghezza impiegheranno rispettivamente 1 secondo e 500 millisecondi.

- **transition-timing-function** : specifica la curva di velocità dell'effetto di transizione. Un valore *lineare* indica che la transizione avrà la stessa velocità dall'inizio alla fine.
- **transizione-ritardo** : specifica la quantità di tempo necessario per attendere prima che inizi l'effetto di transizione. In questo caso, l'altezza inizierà immediatamente la transizione, mentre la larghezza attenderà 1 secondo.

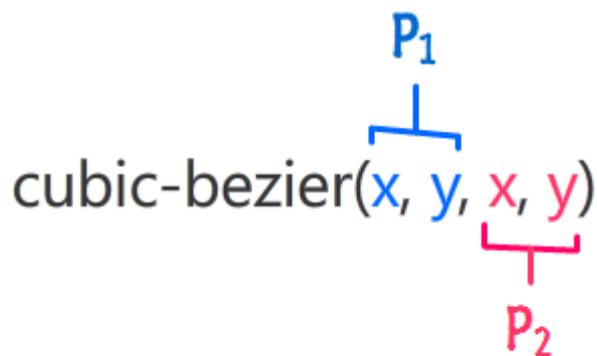
## cubico-Bezier

La funzione `cubic-bezier` è una funzione di temporizzazione della transizione che viene spesso utilizzata per transizioni personalizzate e fluide.

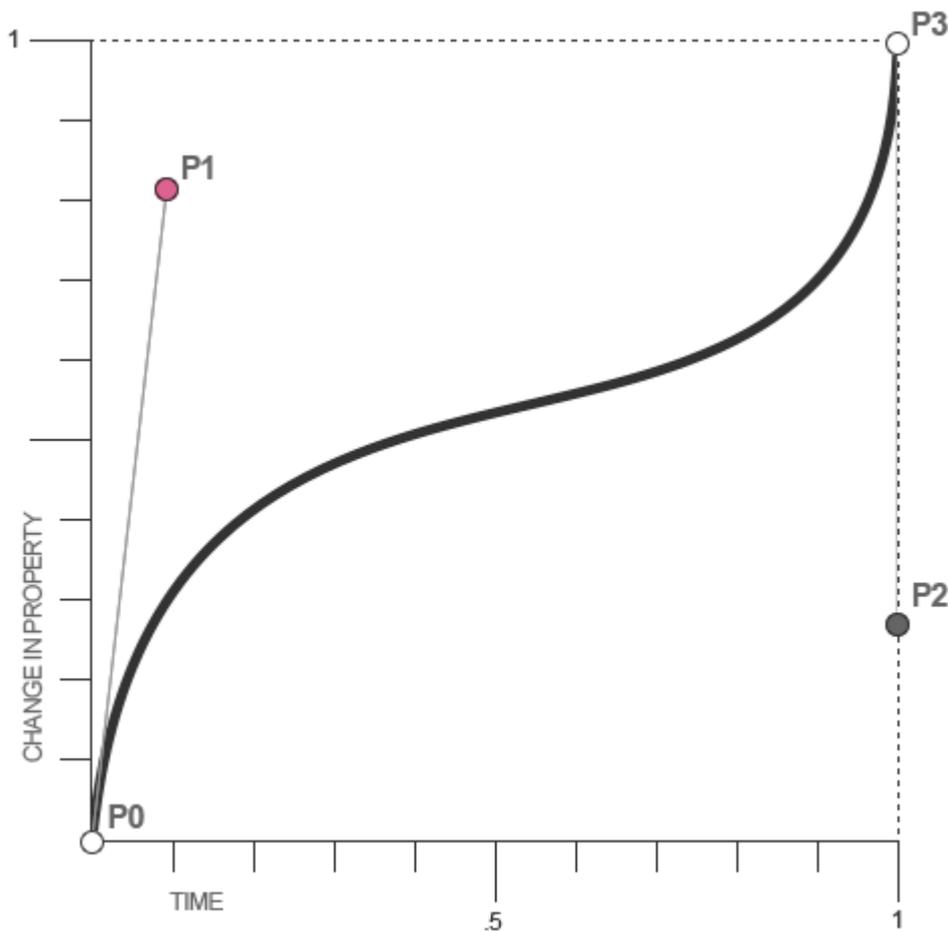
```
transition-timing-function: cubic-bezier(0.1, 0.7, 1.0, 0.1);
```

La funzione richiede quattro parametri:

```
cubic-bezier(P1_x, P1_y, P2_x, P2_y)
```



Questi parametri saranno mappati ai punti che fanno parte di una [curva di Bézier](#) :



Per CSS Bézier Curves, P0 e P3 sono sempre nello stesso punto. P0 è a (0,0) e P3 è a (1,1), il che significa che i parametri passati alla funzione cubico-bezier possono essere solo tra 0 e 1.

Se si passano parametri che non sono in questo intervallo, la funzione passerà automaticamente a una transizione `linear`.

Poiché cubico-bezier è la transizione più flessibile nei CSS, è possibile tradurre tutte le altre funzioni di temporizzazione della transizione in funzioni cubico-bezier:

```
linear : cubic-bezier(0,0,1,1)
```

```
ease-in : cubic-bezier(0.42, 0.0, 1.0, 1.0)
```

```
ease-out : cubic-bezier(0.0, 0.0, 0.58, 1.0)
```

```
ease-in-out : cubic-bezier(0.42, 0.0, 0.58, 1.0)
```

Leggi transizioni online: <https://riptutorial.com/it/css/topic/751/transizioni>

# Capitolo 54: Trasformazioni 2D

## Sintassi

- **Ruota Trasforma**
  - trasforma: ruota (<angolo>)
- **Traduci Trasforma**
  - transform: translate (<lunghezza o percentuale> [, lunghezza o percentuale]?)
  - transform: translateX (<lunghezza o percentuale>)
  - transform: translateY (<lunghezza o percentuale>)
- **Traslazione obliqua**
  - transform: skew (<angle> [, <angle>]?)
  - trasforma: skewX (<angle>)
  - trasformare: skewY (<angle>)
- **Trasforma la scala**
  - transform: scale (<fattore-scala> [, <fattore-scala>]?)
  - transform: scaleX (<fattore di scala>)
  - transform: scaleY (<fattore di scala>)
- **Matrix Transform**
  - transform: matrix (<numero> [, <numero>] {5,5})

## Parametri

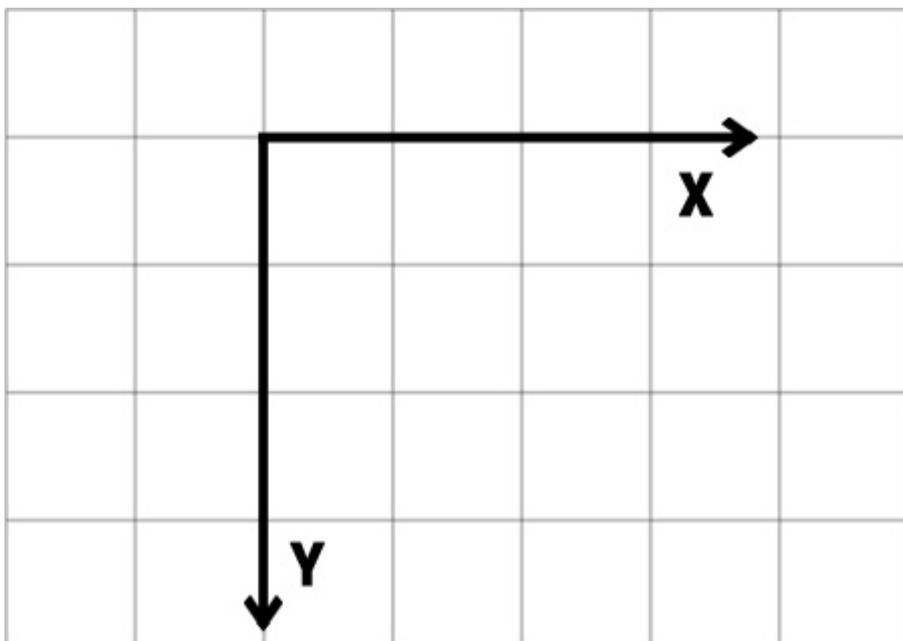
| Funzione / Parametro        | Dettagli  |
|-----------------------------|---|
| <code>rotate(x)</code>      | Definisce una trasformazione che sposta l'elemento attorno a un punto fisso sull'asse Z.                                |
| <code>translate(x,y)</code> | Sposta la posizione dell'elemento sull'asse X e Y.  |
| <code>translateX(x)</code>  | Sposta la posizione dell'elemento sull'asse X.  |
| <code>translateY(y)</code>  | Sposta la posizione dell'elemento sull'asse Y.  |
| <code>scale(x,y)</code>     | Modifica la dimensione dell'elemento sull'asse X e Y.   |
| <code>scaleX(x)</code>      | Modifica la dimensione dell'elemento sull'asse X.   |
| <code>scaleY(y)</code>      | Modifica la dimensione dell'elemento sull'asse Y.   |
| <code>skew(x,y)</code>      | Mappatura a shear, o transvection, distorcendo ogni punto di un elemento di un certo angolo in ogni direzione           |
| <code>skewX(x)</code>       | Mappatura di shear orizzontale che distorce ciascun punto di un elemento di un certo angolo nella direzione orizzontale |

| Funzione / Parametro    | Dettagli   |
|-------------------------|--|
| <code>skewY(y)</code>   | Mappatura di taglio verticale che distorce ciascun punto di un elemento di un certo angolo nella direzione verticale   |
| <code>matrix()</code>   | Definisce una trasformazione 2D sotto forma di una matrice di trasformazione.  |
| angolo                  | L'angolo con cui l'elemento deve essere ruotato o inclinato (a seconda della funzione con cui viene utilizzato). L'angolo può essere fornito in gradi ( <code>deg</code> ), gradienti ( <code>grad</code> ), radianti ( <code>rad</code> ) o giri ( <code>turn</code> ). Nella funzione <code>skew()</code> , il secondo angolo è opzionale. Se non fornito, non ci saranno (0) disallineamenti nell'asse Y. |
| lunghezza-o-percentuale | La distanza espressa come lunghezza o percentuale in base alla quale l'elemento deve essere tradotto. Nella funzione <code>translate()</code> , la seconda lunghezza o percentuale è facoltativa. Se non fornito, non ci sarebbe nessuna (0) traduzione nell'asse Y.   |
| fattore di scala        | Un numero che definisce quante volte l'elemento deve essere ridimensionato nell'asse specificato. Nella funzione <code>scale()</code> , il secondo fattore di scala è facoltativo. Se non viene fornito, verrà applicato anche il primo fattore di scala per l'asse Y.   |

## Osservazioni

### Sistema 2D Coordinante

Le trasformazioni sono realizzate secondo un sistema coordinante 2D X / Y. L'asse X va da destra a sinistra e l'asse Y va verso il basso come mostrato nell'immagine seguente:



Quindi un `translateY()` positivo `translateY()` va verso il basso e un `translateX()` positivo `translateX()` va bene.

## Supporto per browser e prefissi

- IE supporta questa proprietà da IE9 con il prefisso `-ms-`. Versioni precedenti e Edge non hanno bisogno del prefisso
- Firefox lo supporta dalla versione 3.5 e ha bisogno del prefisso `-moz-` fino alla versione 15
- Chrome dalla versione 4 e fino alla versione 34 richiede il prefisso `-webkit-`
- Safari ha bisogno del prefisso `-webkit-` fino alla versione 8
- Opera ha bisogno del prefisso `-o-` per la versione 11.5 e del prefisso `-webkit-` dalla versione 15 alla 22
- Android ha bisogno del prefisso `-webkit-` dalla versione 2.1 alla 4.4.4

## Esempio di trasformazione prefissata:

```
-webkit-transform: rotate(45deg);  
-ms-transform: rotate(45deg);  
transform: rotate(45deg);
```

## Examples

### Ruotare

### HTML

```
<div class="rotate"></div>
```

### CSS

```
.rotate {
  width: 100px;
  height: 100px;
  background: teal;
  transform: rotate(45deg);
}
```

Questo esempio ruoterà il div di 45 gradi in senso orario. Il centro di rotazione si trova al centro del div, 50% da sinistra e 50% dall'alto. Puoi cambiare il centro di rotazione impostando la proprietà `transform-origin`.

```
transform-origin: 100% 50%;
```

L'esempio sopra imposterà il centro di rotazione al centro del lato destro.

## Scala

### HTML

```
<div class="scale"></div>
```

### CSS

```
.scale {
  width: 100px;
  height: 100px;
  background: teal;
  transform: scale(0.5, 1.3);
}
```

Questo esempio ridimensiona il div a  $100\text{px} * 0.5 = 50\text{px}$  sull'asse X ea  $100\text{px} * 1.3 = 130\text{px}$  sull'asse Y.

Il centro della trasformazione si trova al centro del div, 50% da sinistra e 50% dall'alto.

## Tradurre

### HTML

```
<div class="translate"></div>
```

### CSS

```
.translate {
  width: 100px;
  height: 100px;
  background: teal;
  transform: translate(200px, 50%);
}
```

Questo esempio sposterà il div di 200px sull'asse X e di  $100\text{px} * 50\% = 50\text{px}$  sull'asse Y.

Puoi anche specificare le traduzioni su un singolo asse.

Sull'asse X:

```
.translate {
  transform: translateX(200px);
}
```

Sull'asse Y:

```
.translate {
  transform: translateY(50%);
}
```

## Storto

### HTML

```
<div class="skew"></div>
```

### CSS

```
.skew {
  width: 100px;
  height: 100px;
  background: teal;
  transform: skew(20deg, -30deg);
}
```

Questo esempio inclina il div di 20 gradi sull'asse X e di -30 gradi sull'asse Y.

Il centro della trasformazione si trova al centro del div, 50% da sinistra e 50% dall'alto.

Vedi il risultato [qui](#) .

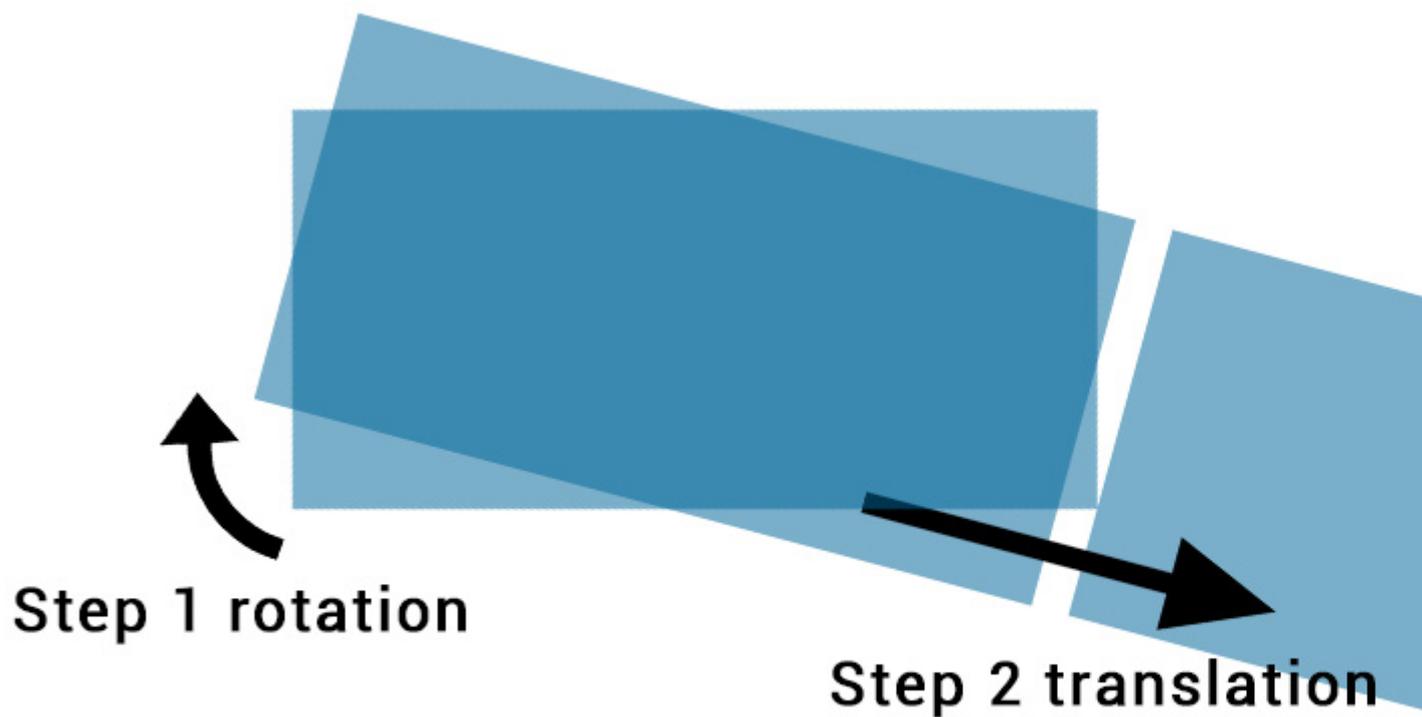
## Trasformazioni multiple

Più trasformazioni possono essere applicate a un elemento in una proprietà come questa:

```
transform: rotate(15deg) translateX(200px);
```

Questo ruoterà l'elemento di 15 gradi in senso orario e poi lo tradurrà a 200px a destra.

Nelle trasformazioni concatenate, **il sistema di coordinate si sposta con l'elemento** . Ciò significa che la traduzione non sarà orizzontale ma su un asse ruoterà di 15 gradi in senso orario come mostrato nell'immagine seguente:



Cambiando l'ordine delle trasformazioni cambierà l'output. Il primo esempio sarà diverso da

```
transform: translateX(200px) rotate(15deg);
```

```
<div class="transform"></div>
```

```
.transform {  
  transform: rotate(15deg) translateX(200px);  
}
```

Come mostrato in questa immagine:



## Step 1 translation

### Trasforma l'origine

Le trasformazioni vengono eseguite rispetto a un punto definito dalla proprietà `transform-origin`.

La proprietà accetta 2 valori: `transform-origin: XY;`

Nell'esempio seguente la prima div ( `.t1` ) ruota intorno all'angolo in alto a sinistra con `transform-origin: 0 0;` e il secondo ( `.tr` ) viene trasformato attorno al suo angolo in alto a destra con `transform-origin: 100% 0`. La rotazione viene applicata **al passaggio del mouse** :

*HTML:*

```
<div class="transform origin1"></div>
<div class="transform origin2"></div>
```

*CSS:*

```
.transform {
  display: inline-block;
  width: 200px;
  height: 100px;
  background: teal;
  transition: transform 1s;
}

.origin1 {
```

```
    transform-origin: 0 0;
}

.origin2 {
    transform-origin: 100% 0;
}

.transform:hover {
    transform: rotate(30deg);
}
```

Il valore predefinito per la proprietà `transform-origin` è `50% 50%` che rappresenta il centro dell'elemento.

Leggi **Trasformazioni 2D** online: <https://riptutorial.com/it/css/topic/938/trasformazioni-2d>

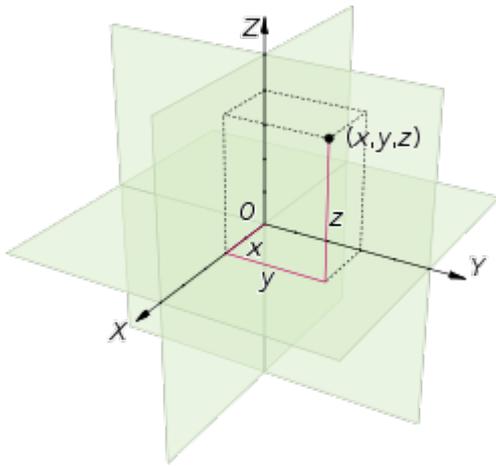
# Capitolo 55: Trasformazioni 3D

## Osservazioni

### Sistema di coordinate

Le trasformazioni 3D sono fatte secondo un sistema di coordinate vettoriali  $(x, y, z)$  nello spazio euclideo .

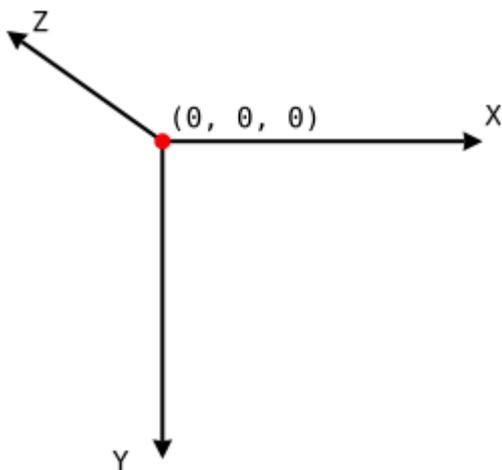
L'immagine seguente mostra un esempio di coordinate nello spazio euclideo:



In CSS,

- L'asse  $x$  rappresenta l'orizzontale (sinistra e destra)
- L'asse  $y$  rappresenta la verticale (su e giù)
- L'asse  $z$  rappresenta la profondità (avanti e indietro / vicino e oltre)

L'immagine seguente mostra come queste coordinate sono tradotte in CSS:



## Examples

## Cubo 3D

Le trasformazioni 3D possono essere utilizzate per creare molte forme 3D. Ecco un semplice esempio di cubo CSS 3D:

### HTML:

```
<div class="cube">
  <div class="cubeFace"></div>
  <div class="cubeFace face2"></div>
</div>
```

### CSS:

```
body {
  perspective-origin: 50% 100%;
  perspective: 1500px;
  overflow: hidden;
}
.cube {
  position: relative;
  padding-bottom: 20%;
  transform-style: preserve-3d;
  transform-origin: 50% 100%;
  transform: rotateY(45deg) rotateX(0);
}
.cubeFace {
  position: absolute;
  top: 0;
  left: 40%;
  width: 20%;
  height: 100%;
  margin: 0 auto;
  transform-style: inherit;
  background: #C52329;
  box-shadow: inset 0 0 0 5px #333;
  transform-origin: 50% 50%;
  transform: rotateX(90deg);
  backface-visibility: hidden;
}
.face2 {
  transform-origin: 50% 50%;
  transform: rotateZ(90deg) translateX(100%) rotateY(90deg);
}
.cubeFace:before, .cubeFace:after {
  content: '';
  position: absolute;
  width: 100%;
  height: 100%;
  transform-origin: 0 0;
  background: inherit;
  box-shadow: inherit;
  backface-visibility: inherit;
}
.cubeFace:before {
  top: 100%;
  left: 0;
  transform: rotateX(-90deg);
```

```
}
.cubeFace:after {
  top: 0;
  left: 100%;
  transform: rotateY(90deg);
}
```

### Guarda questo esempio

Nella demo è stato aggiunto uno stile aggiuntivo e una trasformazione viene applicata al passaggio del mouse per visualizzare le 6 facce del cubo.

Va notato che:

- 4 facce sono fatte con pseudo elementi
- vengono applicate [trasformazioni concatenate](#)

## controfaccia visibilità

La proprietà di `backface-visibility` riferisce a trasformazioni 3D.

Con le trasformazioni 3D e la proprietà di `backface-visibility` del `backface-visibility`, puoi ruotare un elemento in modo tale che il fronte originale di un elemento non sia più rivolto verso lo schermo.

Ad esempio, questo capovolgerebbe un elemento dallo schermo:

### JSFIDDLE

```
<div class="flip">Loren ipsum</div>
<div class="flip back">Lorem ipsum</div>
```

```
.flip {
  -webkit-transform: rotateY(180deg);
  -moz-transform: rotateY(180deg);
  -ms-transform: rotateY(180deg);
  -webkit-backface-visibility: visible;
  -moz-backface-visibility: visible;
  -ms-backface-visibility: visible;
}

.flip.back {
  -webkit-backface-visibility: hidden;
  -moz-backface-visibility: hidden;
  -ms-backface-visibility: hidden;
}
```

Firefox 10+ e IE 10+ supportano la `backface-visibility` senza prefisso. Opera, Chrome, Safari, iOS e Android richiedono tutti `-webkit-backface-visibility`.

Ha 4 valori:

1. **visibile** (predefinito): l'elemento sarà sempre visibile anche quando non è rivolto verso lo schermo.

2. **nascosto** - l'elemento non è visibile quando non è rivolto verso lo schermo.
3. **inherit** : la proprietà otterrà il suo valore dal relativo elemento padre
4. **iniziale** : imposta la proprietà sul valore predefinito, che è visibile

## Puntatore della bussola o forma dell'ago mediante trasformazioni 3D

### CSS

```
div.needle {
  margin: 100px;
  height: 150px;
  width: 150px;
  transform: rotateY(85deg) rotateZ(45deg);
  /* presentational */
  background-image: linear-gradient(to top left, #555 0%, #555 40%, #444 50%, #333 97%);
  box-shadow: inset 6px 6px 22px 8px #272727;
}
```

### HTML

```
<div class='needle'></div>
```

Nell'esempio sopra, una forma di ago o bussola viene creata utilizzando trasformazioni 3D. Generalmente quando applichiamo la trasformazione di `rotate` su un elemento, la rotazione avviene solo sull'asse Z e nella migliore delle ipotesi finiremo con le sole figure diamantate. Ma quando viene aggiunta una trasformazione `rotateY` su di essa, l'elemento viene schiacciato nell'asse Y e finisce quindi con l'aspetto di un ago. Maggiore è la rotazione dell'asse Y, più spremuta l'aspetto dell'elemento.

L'output dell'esempio precedente sarebbe un ago appoggiato sulla punta. Per creare un ago che poggia sulla sua base, la rotazione dovrebbe essere lungo l'asse X anziché lungo l'asse Y. Quindi il valore della proprietà di `transform` dovrebbe essere qualcosa come `rotateX(85deg)`

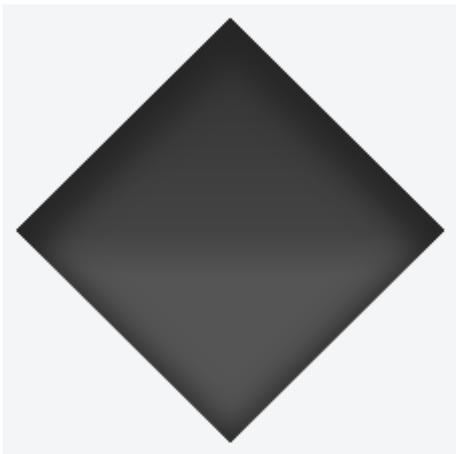
```
rotateZ(45deg); .
```

[Questa penna](#) utilizza un approccio simile per creare qualcosa che assomigli al logo Safari o al quadrante di una bussola.

#### Screenshot dell'elemento senza trasformazione:



## Schermata dell'elemento con solo trasformazione 2D:



## Screenshot dell'elemento con trasformazione 3D:



## Effetto testo 3D con ombra

### HTML:

```
<div id="title">
  <h1 data-content="HOVER">HOVER</h1>
</div>
```

### CSS:

```
*{margin:0;padding:0;}
html,body{height:100%;width:100%;overflow:hidden;background:#0099CC;}
#title{
  position:absolute;
  top:50%; left:50%;
  transform:translate(-50%,-50%);
  perspective-origin:50% 50%;
  perspective:300px;
}
h1{
  text-align:center;
  font-size:12vmin;
  font-family:'Open Sans', sans-serif;
```

```

color:rgba(0,0,0,0.8);
line-height:1em;
transform:rotateY(50deg);
perspective:150px;
perspective-origin:0% 50%;
}
h1:after{
content:attr(data-content);
position:absolute;
left:0;top:0;
transform-origin:50% 100%;
transform:rotateX(-90deg);
color:#0099CC;
}
#title:before{
content:'';
position:absolute;
top:-150%; left:-25%;
width:180%; height:328%;
background:rgba(255,255,255,0.7);
transform-origin: 0 100%;
transform: translatez(-200px) rotate(40deg) skewX(35deg);
border-radius:0 0 100% 0;
}

```

[Visualizza l'esempio con effetto hover aggiuntivo](#)



In questo esempio, il testo viene trasformato per far sembrare che stia andando sullo schermo lontano dall'utente.

L'ombra viene trasformata di conseguenza in modo che segua il testo. Poiché è realizzato con

uno pseudo elemento e l'attributo `data` , eredita le trasformazioni dal genitore (il tag H1).

La "luce" bianca è creata con uno pseudo elemento sull'elemento `#title` . È inclinato e usa il raggio del bordo per l'angolo arrotondato.

Leggi Trasformazioni 3D online: <https://riptutorial.com/it/css/topic/2446/trasformazioni-3d>

# Capitolo 56: Unità di lunghezza

## introduzione

Una misurazione della distanza CSS è un numero immediatamente seguito da un'unità di lunghezza (px, em, pc, in, ...)

CSS supporta un numero di unità di misura della lunghezza. Sono assoluti o relativi.

## Sintassi

- **unità di valore**
- 1em

## Parametri

| Unità      | Descrizione   |
|------------|---|
| %          | Definire le dimensioni in termini di oggetti padre o oggetto corrente dipendente dalla proprietà                  |
| em         | Relativo alla dimensione del carattere dell'elemento (2em significa 2 volte la dimensione del carattere corrente) |
| rem        | Relativo alla dimensione del carattere dell'elemento radice   |
| vw         | Relativo all'1% della larghezza del viewport *  |
| vh         | Relativo all'1% dell'altezza del viewport *   |
| Vmin       | Relativo all'1% della dimensione più piccola * della finestra   |
| vmax       | Relativo all'1% della dimensione più grande * della finestra  |
| centimetro | centimetri  |
| mm         | millimetri  |
| nel        | pollici (1in = 96px = 2,54 cm)  |
| px         | pixel (1px = 1 / 96th di 1in)   |
| pt         | punti (1pt = 1/72 di 1 in)  |
| pc         | picas (1pc = 12 pt)   |
| S          | secondi (usato per animazioni e transizioni)  |

| Unità     | Descrizione   |
|-----------|---|
| Signorina | millisecondi (utilizzato per animazioni e transizioni)  |
| ex        | Relativo all'altezza x del carattere corrente           |
| ch        | Basato sulla larghezza del carattere zero (0)           |
| fr        | unità frazionale (utilizzata per il layout griglia CSS) |

## Osservazioni

- Uno spazio bianco non può apparire tra il numero e l'unità. Tuttavia, se il valore è 0, l'unità può essere omessa.
- Per alcune proprietà CSS, sono consentite lunghezze negative.

## Examples

### Dimensione del carattere con rem

CSS3 introduce alcune nuove unità, inclusa l'unità `rem`, che sta per "root em". Diamo un'occhiata a come funziona `rem`.

Per prima cosa, diamo un'occhiata alle differenze tra `em` e `rem`.

- **em**: relativo alla dimensione del carattere del genitore. Ciò causa il problema di composizione
- **rem**: relativo alla dimensione del carattere dell'elemento root o `<html>`. Ciò significa che è possibile dichiarare una singola dimensione del carattere per l'elemento html e definire tutte le unità `rem` come una percentuale di quello.

Il problema principale con l'uso di `rem` per il dimensionamento dei font è che i valori sono alquanto difficili da usare. Ecco un esempio di alcune dimensioni di carattere comuni espresse in unità `rem`, assumendo che la dimensione di base sia 16px:

- 10px = 0.625rem
- 12px = 0,75rem
- 14px = 0.875rem
- 16px = 1rem (base)
- 18px = 1.125rem
- 20px = 1,25rem
- 24px = 1.5rem
- 30px = 1.875rem
- 32px = 2rem

### CODICE:

3

```

html {
  font-size: 16px;
}

h1 {
  font-size: 2rem;          /* 32px */
}

p {
  font-size: 1rem;         /* 16px */
}

li {
  font-size: 1.5em;       /* 24px */
}

```

## Creazione di elementi scalabili usando rems ed ems

### 3

Puoi usare `rem` definito dalla `font-size` del `font-size` del tuo tag `html` per gli elementi di stile, impostando la `font-size` un valore di `rem` e usando `em` all'interno dell'elemento per creare elementi in scala con la `font-size` globale.

#### HTML:

```





```

#### CSS pertinente:

```

html {
  font-size: 16px;
}

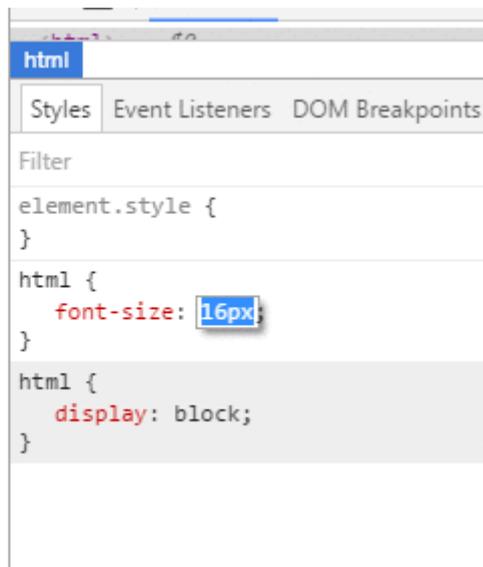
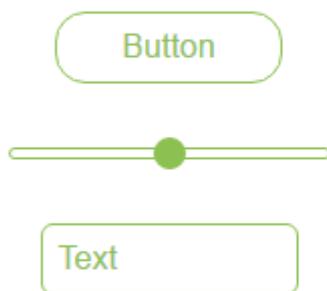
input[type="button"] {
  font-size: 1rem;
  padding: 0.5em 2em;
}

input[type="range"] {
  font-size: 1rem;
  width: 10em;
}

input[type="text"] {
  font-size: 1rem;
  padding: 0.5em;
}

```

#### Possibile risultato:



## vh e vw

CSS3 ha introdotto due unità per rappresentare le dimensioni.

- **vh**, che sta per `viewport height` è relativo all'1% dell'altezza del viewport
- **vw**, che sta per `viewport width` è relativo all'1% della larghezza della finestra

3

```
div {
  width: 20vw;
  height: 20vh;
}
```

Sopra, la dimensione per il div occupa il 20% della larghezza e dell'altezza del viewport

## vmin e vmax

- **vmin** : relativo all'1 percento della dimensione più piccola della finestra
- **vmax** : relativo all'1 percento della dimensione maggiore della finestra

In altre parole,  $1 \text{ vmin}$  è uguale al più piccolo di  $1 \text{ vh}$  e  $1 \text{ vw}$

$1 \text{ vmax}$  è uguale al maggiore di  $1 \text{ vh}$  e  $1 \text{ vw}$

**Nota** :  $\text{vmax}$  non è supportato in:

- qualsiasi versione di Internet Explorer
- Safari prima della versione 6.1

## usando percentuale%

Una delle unità utili durante la creazione di un'applicazione reattiva.

Le sue dimensioni dipendono dal suo contenitore principale.

## Equazione:

(Larghezza contenitore principale) \* (Percentuale (%)) = Output

## Per esempio:

Il genitore ha una larghezza di **100 px** mentre il *bambino* ha il **50%** .

Sull'**output** , la larghezza del *Child* sarà la metà (50%) del *Parent* 's, che è **50px** .

## HTML

```
<div class="parent">
  PARENT
  <div class="child">
    CHILD
  </div>
</div>
```

## CSS

```
<style>

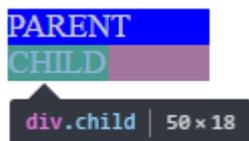
*{
  color: #CCC;
}

.parent{
  background-color: blue;
  width: 100px;
}

.child{
  background-color: green;
  width: 50%;
}

</style>
```

## PRODUZIONE



Leggi Unità di lunghezza online: <https://riptutorial.com/it/css/topic/864/unita-di-lunghezza>

# Titoli di coda

| S. No | Capitoli                                  | Contributors   |
|-------|---|--|
| 1     | Iniziare con i CSS                        | <a href="#">adamboro</a> , <a href="#">animuson</a> , <a href="#">Ashwin Ramaswami</a> , <a href="#">awe</a> , <a href="#">Boysenb3rry</a> , <a href="#">Chris</a> , <a href="#">Community</a> , <a href="#">csx.cc</a> , <a href="#">darrylyeo</a> , <a href="#">FelipeAls</a> , <a href="#">Gabriel R.</a> , <a href="#">Garconis</a> , <a href="#">Gerardas</a> , <a href="#">GoatsWearHats</a> , <a href="#">G-Wiz</a> , <a href="#">Harish Gyanani</a> , <a href="#">Heri Hehe Setiawan</a> , <a href="#">J Atkin</a> , <a href="#">Jmh2013</a> , <a href="#">joe_young</a> , <a href="#">Jose Gomez</a> , <a href="#">Just a student</a> , <a href="#">Lambda Ninja</a> , <a href="#">Marjorie Pickard</a> , <a href="#">Nathan Arthur</a> , <a href="#">patelarpan</a> , <a href="#">RamenChef</a> , <a href="#">Rocket Risa</a> , <a href="#">Saroj Sasmal</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">selvassn</a> , <a href="#">Sverri M. Olsen</a> , <a href="#">Teo Dragovic</a> , <a href="#">Todd</a> , <a href="#">TylerH</a> , <a href="#">Vivek Ghaisas</a> , <a href="#">Xinyang Li</a> , <a href="#">ZaneDickens</a>  |
| 2     | Adattamento e posizionamento dell'oggetto | <a href="#">4444</a> , <a href="#">Miles</a> , <a href="#">RamenChef</a>   |
| 3     | animazioni                                | <a href="#">Aeolingamenfel</a> , <a href="#">apaul</a> , <a href="#">Dex Star</a> , <a href="#">Jasmin Solanki</a> , <a href="#">Nathan Arthur</a> , <a href="#">RamenChef</a> , <a href="#">Richard Hamilton</a> , <a href="#">TylerH</a>   |
| 4     | Blocca i contesti di formattazione        | <a href="#">Madalina Taina</a> , <a href="#">Milan Laslop</a> , <a href="#">RamenChef</a>  |
| 5     | box-ombra                                 | <a href="#">Hristo</a> , <a href="#">Madalina Taina</a> , <a href="#">RamenChef</a>  |
| 6     | Cascading e Specificità                   | <a href="#">amflare</a> , <a href="#">Arjan Einbu</a> , <a href="#">brandaemon</a> , <a href="#">DarkAjax</a> , <a href="#">dippas</a> , <a href="#">dmkerr</a> , <a href="#">geeksal</a> , <a href="#">Grant Palin</a> , <a href="#">G-Wiz</a> , <a href="#">James Donnelly</a> , <a href="#">jehna1</a> , <a href="#">John Slegers</a> , <a href="#">kingcobra1986</a> , <a href="#">Matas Vaitkevicius</a> , <a href="#">Nathan Arthur</a> , <a href="#">Nhan</a> , <a href="#">Ortomala Lokni</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Sunnyok</a> , <a href="#">Ze Rubeus</a>  |
| 7     | Centraggio                                | <a href="#">abaracedo</a> , <a href="#">Alex Morales</a> , <a href="#">Alohci</a> , <a href="#">andreas</a> , <a href="#">animuson</a> , <a href="#">apaul</a> , <a href="#">Christiaan Maks</a> , <a href="#">Daniel Käfer</a> , <a href="#">Devid Farinelli</a> , <a href="#">Diego V</a> , <a href="#">dippas</a> , <a href="#">Eliran Malka</a> , <a href="#">Emanuele Parisio</a> , <a href="#">Euan Williams</a> , <a href="#">F. Müller</a> , <a href="#">Farzad YZ</a> , <a href="#">Felix A J</a> , <a href="#">geek1011</a> , <a href="#">insertusernamehere</a> , <a href="#">JedaiCoder</a> , <a href="#">jehna1</a> , <a href="#">JHS</a> , <a href="#">John Slegers</a> , <a href="#">Jonathan Argentiero</a> , <a href="#">Kilian Stinson</a> , <a href="#">Kyle Ratliff</a> , <a href="#">Lambda Ninja</a> , <a href="#">Lucia Bentivoglio</a> , <a href="#">Luke Taylor</a> , <a href="#">Madalina Taina</a> , <a href="#">maho</a> , <a href="#">Maxouhell</a> , <a href="#">Michael_B</a> , <a href="#">Mifeet</a> , <a href="#">Mod Proxy</a> , <a href="#">Mohammad Usman</a> , <a href="#">Nathan Arthur</a> , <a href="#">Nhan</a> , <a href="#">o.v.</a> , <a href="#">Ortomala Lokni</a> , <a href="#">paaacman</a> , <a href="#">Paul Kozlovitch</a> , <a href="#">Praveen Kumar</a> , <a href="#">Sandeep Tuniki</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Sergej</a> , <a href="#">Siavas</a> , <a href="#">smonff</a> , <a href="#">Someone</a> , <a href="#">Stewartside</a> , <a href="#">Sunnyok</a> , <a href="#">Taylor</a> , <a href="#">TylerH</a> , <a href="#">web-tiki</a> , <a href="#">Ze Rubeus</a> , <a href="#">zeel</a> |
| 8     | Centratura verticale                      | <a href="#">animuson</a> , <a href="#">AVAVT</a> , <a href="#">bocanegra</a> , <a href="#">Chiller</a> , <a href="#">Chris</a> , <a href="#">jaredsk</a> , <a href="#">leo_ap</a> ,  |

|    |                           |  |
|----|---------------------------|--|
|    |                           | <a href="#">mmativ</a> , <a href="#">patelarpan</a> , <a href="#">Phil</a> , <a href="#">Praveen Kumar</a> , <a href="#">RamenChef</a>   |
| 9  | colonne                   | <a href="#">Brett DeWoody</a> , <a href="#">Madalina Taina</a> , <a href="#">RamenChef</a>   |
| 10 | Colonne multiple          | <a href="#">bipon</a> , <a href="#">Sebastian Zartner</a>  |
| 11 | Colori                    | <a href="#">andreas</a> , <a href="#">animuson</a> , <a href="#">Arjan Einbu</a> , <a href="#">Brett DeWoody</a> , <a href="#">Community</a> , <a href="#">cuervoo</a> , <a href="#">darrylyeo</a> , <a href="#">designcise</a> , <a href="#">H. Pauwelyn</a> , <a href="#">Jasmin Solanki</a> , <a href="#">John Slegers</a> , <a href="#">Kuhan</a> , <a href="#">Marc</a> , <a href="#">Michael Moriarty</a> , <a href="#">Miro</a> , <a href="#">Nathan Arthur</a> , <a href="#">niyasc</a> , <a href="#">RamenChef</a> , <a href="#">Richard Hamilton</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">SeinopSys</a> , <a href="#">Stewartside</a> , <a href="#">user007</a> , <a href="#">Wolfgang</a> , <a href="#">X-27</a> |
| 12 | Commenti                  | <a href="#">animuson</a> , <a href="#">bdkopen</a> , <a href="#">coderfin</a> , <a href="#">Madalina Taina</a> , <a href="#">Nick</a>  |
| 13 | Confine                   | <a href="#">andreas</a> , <a href="#">Cassidy Williams</a> , <a href="#">doctorsherlock</a> , <a href="#">FelipeAls</a> , <a href="#">Gnietschow</a> , <a href="#">Harry</a> , <a href="#">jaredsk</a> , <a href="#">Madalina Taina</a> , <a href="#">Nobal Mohan</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Trevor Clarke</a>  |
| 14 | contatori                 | <a href="#">Harry</a> , <a href="#">RamenChef</a>  |
| 15 | Contesto impilabile       | <a href="#">Nemanja Trifunovic</a> , <a href="#">RamenChef</a>   |
| 16 | Controllo del layout      | <a href="#">Arjun Iv</a> , <a href="#">Chathuranga Jayanath</a> , <a href="#">Chris</a> , <a href="#">Jmh2013</a> , <a href="#">Kevin Katzke</a> , <a href="#">Kurtis Beavers</a> , <a href="#">Madalina Taina</a> , <a href="#">mnoronha</a> , <a href="#">Niek Brouwer</a> , <a href="#">RamenChef</a> , <a href="#">Sander Koedood</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">SeinopSys</a>  |
| 17 | CSS Image Sprites         | <a href="#">Elegant.Scripting</a> , <a href="#">Jmh2013</a> , <a href="#">RamenChef</a> , <a href="#">Ted Goas</a> , <a href="#">Ulrich Schwarz</a>  |
| 18 | CSS Object Model (CSSOM)  | <a href="#">Alohci</a> , <a href="#">animuson</a> , <a href="#">feeela</a> , <a href="#">Paul Sweatte</a> , <a href="#">RamenChef</a> , <a href="#">rishabh dev</a>  |
| 19 | Cursore Styling           | <a href="#">cone56</a> , <a href="#">Madalina Taina</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Squazz</a>   |
| 20 | Elenca stili              | <a href="#">animuson</a> , <a href="#">Madalina Taina</a> , <a href="#">Marten Koetsier</a> , <a href="#">RamenChef</a> , <a href="#">Ted Goas</a>   |
| 21 | Eredità                   | <a href="#">Chris</a> , <a href="#">mnoronha</a> , <a href="#">RamenChef</a>   |
| 22 | Forme ad elementi singoli | <a href="#">andreas</a> , <a href="#">animuson</a> , <a href="#">Brett DeWoody</a> , <a href="#">Chiller</a> , <a href="#">J F</a> , <a href="#">Nick</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">TylerH</a>   |
| 23 | Forme per galleggianti    | <a href="#">animuson</a> , <a href="#">Harry</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a>   |
| 24 | funzioni                  | <a href="#">animuson</a> , <a href="#">Brett DeWoody</a> , <a href="#">cone56</a> , <a href="#">dodopok</a> , <a href="#">H. Pauwelyn</a> , <a href="#">haim770</a> , <a href="#">jaredsk</a> , <a href="#">khawarPK</a> , <a href="#">Kobi</a> , <a href="#">RamenChef</a> , <a href="#">SeinopSys</a> , <a href="#">TheGenie OfTruth</a> , <a href="#">TylerH</a>  |
| 25 | Galleggianti              | <a href="#">demonofthemist</a> , <a href="#">FelipeAls</a> , <a href="#">Madalina Taina</a> , <a href="#">Nathan Arthur</a> ,  |

|    |   |  |
|----|---|--|
|    |   | <a href="#">RamenChef</a> , <a href="#">vishak</a>   |
| 26 | Griglia                                   | <a href="#">Chris Spittles</a> , <a href="#">FelipeAls</a> , <a href="#">Jonathan Zúñiga</a> , <a href="#">Mike McCaughan</a> , <a href="#">Nhan</a> , <a href="#">Praveen Kumar</a> , <a href="#">RamenChef</a> , <a href="#">Sebastian Zartner</a>   |
| 27 | Hack di Internet Explorer                 | <a href="#">Aeolingamenfel</a> , <a href="#">animuson</a> , <a href="#">Elizaveta Revyakina</a> , <a href="#">feeela</a> , <a href="#">John Slegers</a> , <a href="#">LiLacTac</a> , <a href="#">Praveen Kumar</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Timothy Miller</a> , <a href="#">TylerH</a>   |
| 28 | I selettori                               | <a href="#">75th Trombone</a> , <a href="#">A.J. Aaron</a> , <a href="#">abaracedo</a> , <a href="#">Ahmad Alfy</a> , <a href="#">Alex Filatov</a> , <a href="#">amflare</a> , <a href="#">Anil</a> , <a href="#">animuson</a> , <a href="#">Araknid</a> , <a href="#">Arjan Einbu</a> , <a href="#">Ashwin Ramaswami</a> , <a href="#">BoltClock</a> , <a href="#">Cerbrus</a> , <a href="#">Charlie H</a> , <a href="#">Chris</a> , <a href="#">Chris Nager</a> , <a href="#">Clinton Yeboah</a> , <a href="#">Community</a> , <a href="#">CPHPython</a> , <a href="#">darrylyeo</a> , <a href="#">Dave Everitt</a> , <a href="#">David Fullerton</a> , <a href="#">Demeter Dimitri</a> , <a href="#">designcise</a> , <a href="#">Devid Farinelli</a> , <a href="#">Devon Bernard</a> , <a href="#">Dinidu</a> , <a href="#">dippas</a> , <a href="#">Erenor Paz</a> , <a href="#">Felix Edelmann</a> , <a href="#">Felix Schütz</a> , <a href="#">flyingfisch</a> , <a href="#">Forty</a> , <a href="#">fracz</a> , <a href="#">Frits</a> , <a href="#">gandreadis</a> , <a href="#">geeksal</a> , <a href="#">George Bailey</a> , <a href="#">George Grigorita</a> , <a href="#">H. Pauwelyn</a> , <a href="#">HansCz</a> , <a href="#">henry</a> , <a href="#">Hugo Buff</a> , <a href="#">Hynes</a> , <a href="#">J Atkin</a> , <a href="#">J F</a> , <a href="#">Jacob Gray</a> , <a href="#">James Donnelly</a> , <a href="#">James Taylor</a> , <a href="#">Jasha</a> , <a href="#">jehna1</a> , <a href="#">Jmh2013</a> , <a href="#">joejoe31b</a> , <a href="#">Joël Bonet Rodríguez</a> , <a href="#">John Slegers</a> , <a href="#">Kurtis Beavers</a> , <a href="#">Madalina Taina</a> , <a href="#">Marc</a> , <a href="#">Mark Perera</a> , <a href="#">Matas Vaitkevicius</a> , <a href="#">Matsemann</a> , <a href="#">Michael_B</a> , <a href="#">Milan Laslop</a> , <a href="#">Naeem Shaikh</a> , <a href="#">Nathan Arthur</a> , <a href="#">Nick</a> , <a href="#">Ortomala Lokni</a> , <a href="#">Persijn</a> , <a href="#">Praveen Kumar</a> , <a href="#">RamenChef</a> , <a href="#">rdans</a> , <a href="#">Richard Hamilton</a> , <a href="#">Rion Williams</a> , <a href="#">Robert Koritnik</a> , <a href="#">RockPaperLizard</a> , <a href="#">RoToRa</a> , <a href="#">Sbats</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Shaggy</a> , <a href="#">Siavas</a> , <a href="#">Stewartside</a> , <a href="#">sudo bangbang</a> , <a href="#">Sumner Evans</a> , <a href="#">Sunnyok</a> , <a href="#">ThatWeirdo</a> , <a href="#">theB</a> , <a href="#">Thomas Gerot</a> , <a href="#">TylerH</a> , <a href="#">xpy</a> , <a href="#">Yury Fedorov</a> , <a href="#">Zac</a> , <a href="#">Zaffy</a> , <a href="#">Zaz</a> , <a href="#">Ze Rubeus</a> , <a href="#">Zze</a> |
| 29 | Il modello di scatola                     | <a href="#">Arjan Einbu</a> , <a href="#">Ben Rhys-Lewis</a> , <a href="#">Benolot</a> , <a href="#">BiscuitBaker</a> , <a href="#">FelipeAls</a> , <a href="#">James Donnelly</a> , <a href="#">Lambda Ninja</a> , <a href="#">Nathan Arthur</a> , <a href="#">Ortomala Lokni</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Sergej</a> , <a href="#">V-Kopio</a>  |
| 30 | Imbottitura                               | <a href="#">Andy G</a> , <a href="#">CalvT</a> , <a href="#">Felix A J</a> , <a href="#">Madalina Taina</a> , <a href="#">Mehdi Dehghani</a> , <a href="#">Nathan</a> , <a href="#">Paul Sweatte</a> , <a href="#">pixelbandito</a> , <a href="#">RamenChef</a> , <a href="#">StefanBob</a> , <a href="#">Will DiFruscio</a>   |
| 31 | La frammentazione                         | <a href="#">animuson</a> , <a href="#">dodopok</a> , <a href="#">Madalina Taina</a> , <a href="#">Milan Laslop</a> , <a href="#">RamenChef</a>   |
| 32 | Layout di blocco in linea                 | <a href="#">Marten Koetsier</a> , <a href="#">RamenChef</a>  |
| 33 | Layout flessibile della scatola (Flexbox) | <a href="#">Ahmad Alfy</a> , <a href="#">Asim K T</a> , <a href="#">FelipeAls</a> , <a href="#">fzzylogic</a> , <a href="#">James Donnelly</a> , <a href="#">Jef</a> , <a href="#">Lambda Ninja</a> , <a href="#">Marc-Antoine Leclerc</a> , <a href="#">Nathan Arthur</a> , <a href="#">Nhan</a> , <a href="#">Ori Marash</a> , <a href="#">RamenChef</a> , <a href="#">Randy</a> , <a href="#">Squazz</a> , <a href="#">takeradi</a> , <a href="#">Ted Goas</a> , <a href="#">Timothy Morris</a> , <a href="#">TylerH</a>  |
| 34 | Lineamenti                                | <a href="#">Arif</a> , <a href="#">Casey</a> , <a href="#">Chathuranga Jayanath</a> , <a href="#">Daniel Nugent</a> , <a href="#">FelipeAls</a> , <a href="#">Madalina Taina</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a>   |

|    |   |  |
|----|---|--|
| 35 | margin                                  | <a href="#">Arjan Einbu</a> , <a href="#">cdm</a> , <a href="#">Chris Spittles</a> , <a href="#">Community</a> , <a href="#">J F</a> , <a href="#">Madalina Taina</a> , <a href="#">Mr_Green</a> , <a href="#">Nathan Arthur</a> , <a href="#">RamenChef</a> , <a href="#">rejnev</a> , <a href="#">Sun Qingyao</a> , <a href="#">Sunnyok</a> , <a href="#">Tot Zam</a> , <a href="#">Trevor Clarke</a>  |
| 36 | Media Queries                           | <a href="#">amflare</a> , <a href="#">Chathuranga Jayanath</a> , <a href="#">darrylyeo</a> , <a href="#">Demeter Dimitri</a> , <a href="#">dodopok</a> , <a href="#">James Donnelly</a> , <a href="#">Jmh2013</a> , <a href="#">joe_young</a> , <a href="#">joejoe31b</a> , <a href="#">John Slegers</a> , <a href="#">Matas Vaitkevicius</a> , <a href="#">Mattia Astorino</a> , <a href="#">Maximillian Laumeister</a> , <a href="#">Nathan Arthur</a> , <a href="#">Praveen Kumar</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">srikarg</a> , <a href="#">Teo Dragovic</a> , <a href="#">Viktor</a>   |
| 37 | Modelli di progettazione CSS            | <a href="#">John Slegers</a>   |
| 38 | Normalizzazione degli stili del browser | <a href="#">andre mcgruder</a> , <a href="#">Confused One</a> , <a href="#">Grant Palin</a> , <a href="#">MMachinegun</a> , <a href="#">mnoronha</a> , <a href="#">RamenChef</a> , <a href="#">SeinopSys</a>   |
| 39 | Opacità                                 | <a href="#">Andrew</a> , <a href="#">animuson</a> , <a href="#">Hillel Tech</a> , <a href="#">Madalina Taina</a> , <a href="#">RamenChef</a>   |
| 40 | Posizionamento                          | <a href="#">Abhishek Singh</a> , <a href="#">Alohci</a> , <a href="#">animuson</a> , <a href="#">Blunderfest</a> , <a href="#">CalvT</a> , <a href="#">Cassidy Williams</a> , <a href="#">Chetan Joshi</a> , <a href="#">GingerPlusPlus</a> , <a href="#">Jacob Gray</a> , <a href="#">Lambda Ninja</a> , <a href="#">Madalina Taina</a> , <a href="#">Matthew Beckman</a> , <a href="#">Mattia Astorino</a> , <a href="#">Rahul Nanwani</a> , <a href="#">RamenChef</a> , <a href="#">Sourav Ghosh</a> , <a href="#">Stephen Leppik</a> , <a href="#">Theodore K.</a> , <a href="#">TylerH</a>  |
| 41 | Prestazione                             | <a href="#">mnoronha</a> , <a href="#">RamenChef</a> , <a href="#">TrungDQ</a>   |
| 42 | Proprietà filtro                        | <a href="#">Jeffery Tang</a> , <a href="#">Nathan</a> , <a href="#">RamenChef</a>  |
| 43 | Proprietà personalizzate (variabili)    | <a href="#">animuson</a> , <a href="#">Brett DeWoody</a> , <a href="#">Community</a> , <a href="#">Daniel Käfer</a> , <a href="#">Muthu Kumaran</a> , <a href="#">Obsidian</a> , <a href="#">RamenChef</a> , <a href="#">RedRiderX</a> , <a href="#">TylerH</a>  |
| 44 | Pseudo-Elementi                         | <a href="#">4dgaurav</a> , <a href="#">ankit</a> , <a href="#">Chris</a> , <a href="#">Community</a> , <a href="#">dippas</a> , <a href="#">dmnsgn</a> , <a href="#">geek1011</a> , <a href="#">geeksal</a> , <a href="#">Gofilord</a> , <a href="#">kevin vd</a> , <a href="#">Marcatectura</a> , <a href="#">Milche Patern</a> , <a href="#">Nathan</a> , <a href="#">Pat</a> , <a href="#">Praveen Kumar</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Shaggy</a> , <a href="#">Stewartside</a> , <a href="#">Sunnyok</a>   |
| 45 | Query di caratteristiche                | <a href="#">Andrew Myers</a> , <a href="#">RamenChef</a>   |
| 46 | Ritaglio e mascheratura                 | <a href="#">Andre Lopes</a> , <a href="#">Harry</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">web-tiki</a>   |
| 47 | sfondi                                  | <a href="#">4444</a> , <a href="#">Ahmad Alfy</a> , <a href="#">animuson</a> , <a href="#">Asim K T</a> , <a href="#">Ben Rhys-Lewis</a> , <a href="#">Boris</a> , <a href="#">CalvT</a> , <a href="#">cdm</a> , <a href="#">Charlie H</a> , <a href="#">CocoaBean</a> , <a href="#">Dan Devine</a> , <a href="#">Dan Eastwell</a> , <a href="#">Daniel G. Blázquez</a> , <a href="#">Daniel Stradowski</a> , <a href="#">Darthstroke</a> , <a href="#">designcise</a> , <a href="#">Devid Farinelli</a> , <a href="#">dippas</a> , <a href="#">fcalderan</a> , <a href="#">FelipeAls</a> , <a href="#">Goose</a> , <a href="#">Horst Jahns</a> , <a href="#">Hynes</a> , <a href="#">Jack</a> , <a href="#">Jacob Gray</a> , <a href="#">James Taylor</a> , <a href="#">John Slegers</a> , <a href="#">Jon Chan</a> , <a href="#">Jonathan Zúñiga</a> , <a href="#">Kevin Montrose</a> , <a href="#">Louis St-Amour</a> , |

|    |   |   |
|----|---|---|
|    |   | <a href="#">Madalina Taina</a> , <a href="#">Maximillian Laumeister</a> , <a href="#">Michael Moriarty</a> , <a href="#">Mr. Alien</a> , <a href="#">mtb</a> , <a href="#">Nate</a> , <a href="#">Nathan Arthur</a> , <a href="#">Nhan</a> , <a href="#">Persijn</a> , <a href="#">Praveen Kumar</a> , <a href="#">RamenChef</a> , <a href="#">Richard Hamilton</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Sergey Denisov</a> , <a href="#">Shaggy</a> , <a href="#">Sourav Ghosh</a> , <a href="#">Stewartside</a> , <a href="#">Stratboy</a> , <a href="#">think123</a> , <a href="#">Timothy</a> , <a href="#">Trevor Clarke</a> , <a href="#">TylerH</a> , <a href="#">Zac</a> , <a href="#">Zeta</a> , <a href="#">Zze</a>  |
| 48 | straripamento                               | <a href="#">Andrew</a> , <a href="#">bdkopen</a> , <a href="#">jgh</a> , <a href="#">Madalina Taina</a> , <a href="#">Miles</a> , <a href="#">Qaz</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Ted Goas</a> , <a href="#">Zac</a>  |
| 49 | Struttura e formattazione di una regola CSS | <a href="#">Alon Eitan</a> , <a href="#">darrylyeo</a> , <a href="#">Marjorie Pickard</a>   |
| 50 | Supporto browser e prefissi                 | <a href="#">Andrew</a> , <a href="#">animuson</a> , <a href="#">Braiam</a> , <a href="#">Nhan</a> , <a href="#">Obsidian</a> , <a href="#">RamenChef</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Shaggy</a> , <a href="#">TylerH</a>  |
| 51 | tabelle                                     | <a href="#">Casper Spruit</a> , <a href="#">Chris</a> , <a href="#">FelipeAls</a> , <a href="#">JHS</a> , <a href="#">Madalina Taina</a>  |
| 52 | Tipografia                                  | <a href="#">Alex Morales</a> , <a href="#">Alohci</a> , <a href="#">andreas</a> , <a href="#">Arjan Einbu</a> , <a href="#">ChaoticTwist</a> , <a href="#">Evgeny</a> , <a href="#">Felix A J</a> , <a href="#">Goulven</a> , <a href="#">Hynes</a> , <a href="#">insertusernamehere</a> , <a href="#">James Donnelly</a> , <a href="#">joe_young</a> , <a href="#">Jon Chan</a> , <a href="#">Madalina Taina</a> , <a href="#">Michael Moriarty</a> , <a href="#">Nathan Arthur</a> , <a href="#">Nhan</a> , <a href="#">Praveen Kumar</a> , <a href="#">RamenChef</a> , <a href="#">Richard Hamilton</a> , <a href="#">rmondessilva</a> , <a href="#">Ryan</a> , <a href="#">Sourav Ghosh</a> , <a href="#">Suhaib Janjua</a> , <a href="#">Ted Goas</a> , <a href="#">Toby</a> , <a href="#">ToniB</a> , <a href="#">Trevor Clarke</a> , <a href="#">user2622348</a> , <a href="#">Vlusion</a> , <a href="#">Volker E.</a> |
| 53 | transizioni                                 | <a href="#">Christiaan Maks</a> , <a href="#">dippas</a> , <a href="#">Harry</a> , <a href="#">Praveen Kumar</a> , <a href="#">RamenChef</a> , <a href="#">Richard Hamilton</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">Sergey Denisov</a> , <a href="#">web-tiki</a>   |
| 54 | Trasformazioni 2D                           | <a href="#">Charlie H</a> , <a href="#">Christiaan Maks</a> , <a href="#">Harry</a> , <a href="#">Hors Sujet</a> , <a href="#">John Slegers</a> , <a href="#">Luke Taylor</a> , <a href="#">Madalina Taina</a> , <a href="#">mnoronha</a> , <a href="#">PaMaDo</a> , <a href="#">Praveen Kumar</a> , <a href="#">RamenChef</a> , <a href="#">web-tiki</a> , <a href="#">zer00ne</a>   |
| 55 | Trasformazioni 3D                           | <a href="#">Harry</a> , <a href="#">Luka Kerr</a> , <a href="#">Madalina Taina</a> , <a href="#">mnoronha</a> , <a href="#">Mr. Meeseeks</a> , <a href="#">Nhan</a> , <a href="#">RamenChef</a> , <a href="#">web-tiki</a>  |
| 56 | Unità di lunghezza                          | <a href="#">4dgaurav</a> , <a href="#">A B</a> , <a href="#">animuson</a> , <a href="#">Epodax</a> , <a href="#">geeksal</a> , <a href="#">J F</a> , <a href="#">Marc</a> , <a href="#">Milche Patern</a> , <a href="#">Ortomala Lokni</a> , <a href="#">RamenChef</a> , <a href="#">Richard Hamilton</a> , <a href="#">rmondessilva</a> , <a href="#">Robert Koritnik</a> , <a href="#">Robotnicka</a> , <a href="#">ScientiaEtVeritas</a> , <a href="#">StefanBob</a> , <a href="#">Stewartside</a> , <a href="#">Thomas Altmann</a> , <a href="#">Toby</a> , <a href="#">user2622348</a> , <a href="#">vladdobra</a> , <a href="#">Zakaria Acharki</a> , <a href="#">zer00ne</a>   |