



EBook Gratuito

APPENDIMENTO HTML

Free unaffiliated eBook created from
Stack Overflow contributors.

#html

Sommario

Di.....	1
Capitolo 1: Iniziare con HTML.....	2
Osservazioni.....	2
Versioni.....	2
Examples.....	2
Ciao mondo.....	2
introduzione.....	2
Intuizione dell'elemento.....	3
Creare una semplice pagina.....	3
Semplice pagina abbattere.....	4
Capitolo 2: Ancore e collegamenti ipertestuali.....	6
introduzione.....	6
Sintassi.....	6
Parametri.....	6
Examples.....	7
Link a un altro sito.....	7
Apri il link in una nuova scheda / finestra.....	7
Collegamento a un'ancora.....	8
Link che esegue JavaScript.....	8
Link a una pagina sullo stesso sito.....	9
Link che esegue il client di posta elettronica.....	9
Link che compone un numero.....	10
Capitolo 3: ARIA.....	11
Sintassi.....	11
Osservazioni.....	11
Examples.....	12
role = "alert".....	12
role = "alertdialog".....	12
role = "applicazione".....	12
role = "articolo".....	12

role = "banner"	13
role = "button"	13
role = "cella"	13
role = "checkbox"	13
role = "columnheader"	14
role = "Casella combinata"	14
role = "complementare"	14
role = "ContentInfo"	14
role = "definizione"	15
role = "finestra"	15
role = "directory"	15
role = "documento"	15
role = "forma"	15
role = "griglia"	16
role = "gridcell"	16
role = "gruppo"	17
role = "heading"	17
role = "img"	17
role = "link"	17
role = "lista"	17
role = "casella di riepilogo"	18
role = "listitem"	18
role = "log"	18
role = "main"	18
role = "tendone"	18
role = "matematica"	19
role = "menu"	19
role = "barra dei menu"	19
role = "menuitem"	19
role = "menuitemcheckbox"	19
role = "menuitemradio"	20
role = "navigazione"	20
role = "note"	20

role = "opzione"	20
role = "presentazione"	20
role = "progressbar"	21
role = "radio"	21
role = "regione"	21
role = "RadioGroup"	21
role = "riga"	21
role = "rowgroup"	22
role = "rowheader"	22
role = "barra di scorrimento"	22
role = "search"	23
role = "searchbox"	23
role = "separatore"	23
role = "slider"	23
role = "SpinButton"	23
role = "Stato"	24
role = "interruttore"	24
role = "tab"	24
role = "tavolo"	24
role = "tablist"	24
role = "tabpanel"	25
role = "testo"	25
role = "timer"	25
role = "barra degli strumenti"	25
role = "tooltip"	25
role = "albero"	26
role = "treegrid"	26
role = "treeitem"	26

Capitolo 4: Attributi dei dati	28
Sintassi	28
Parametri	28
Examples	28
Uso degli attributi dei dati	28

Supporto per i browser meno recenti.....	28
Capitolo 5: Attributi evento HTML.....	30
Examples.....	30
Eventi modulo HTML.....	30
Eventi di tastiera.....	30
Capitolo 6: Attributi globali.....	31
Parametri.....	31
Osservazioni.....	31
Examples.....	31
Attributo Contenteditable.....	31
Capitolo 7: Barre di navigazione.....	33
Examples.....	33
Barra di navigazione di base.....	33
Barra di navigazione HTML5.....	33
Capitolo 8: Classi e ID.....	34
introduzione.....	34
Sintassi.....	34
Parametri.....	34
Osservazioni.....	34
Examples.....	34
Dando un elemento a una classe.....	34
Usando le classi in CSS.....	35
Dando un elemento un ID.....	36
Problemi relativi agli ID duplicati.....	36
Valori accettabili.....	37
Per un ID.....	37
Per una classe.....	38
Nota importante: il modo in cui i valori ID e Classe vengono trattati al di fuori dell'HTM.....	38
Riferimenti W3C.....	39
Capitolo 9: Collegamento di risorse.....	40
introduzione.....	40
Sintassi.....	40

Parametri.....	40
Examples.....	40
Foglio di stile CSS esterno.....	41
JavaScript.....	41
Sincrono.....	41
asincrono.....	41
differite.....	42
<Noscript>.....	42
favicon.....	42
CSS alternativo.....	42
Feed Web.....	43
Collega l'attributo 'media'.....	43
Prev e Next.....	43
Suggerimento sulla risorsa: dns-prefetch, prefetch, prerender.....	43
preconnect.....	43
DNS-Prefetch.....	43
prefetch.....	44
Prerender.....	44
Capitolo 10: Commenti.....	45
introduzione.....	45
Sintassi.....	45
Osservazioni.....	45
Examples.....	45
Creazione di commenti.....	45
Commenti condizionali per Internet Explorer.....	46
Di livello inferiore-nascosto.....	46
Di livello inferiore-rivelato.....	47
Commentando spazi bianchi tra elementi in linea.....	47
Capitolo 11: Controlli del menu di selezione.....	49
Sintassi.....	49
Examples.....	49

Seleziona Menu.....	49
Modifica della dimensione.....	49
Menu di selezione multi-opzione.....	49
Gruppi di opzioni.....	50
Opzioni.....	50
Selezione di un'opzione per impostazione predefinita.....	51
Datalist.....	51
Supporto del browser.....	51
Capitolo 12: doctype.....	53
introduzione.....	53
Sintassi.....	53
Osservazioni.....	53
Examples.....	53
Aggiungere il Doctype.....	53
Documento HTML 4.01.....	54
HTML 4.01 Strict.....	54
HTML 4.01 Transitional.....	54
HTML 4.01 Frameset.....	54
HTML 5 Doctype.....	54
Case Insensitivity.....	55
Vecchi doctypes.....	55
HTML 3.2.....	55
HTML 2.0.....	55
Capitolo 13: Elementi di sezionamento.....	56
Osservazioni.....	56
Examples.....	56
Articolo Element.....	56
Evitare l'uso non necessario.....	56
Elemento principale.....	57
Elemento Nav.....	58
Elementi in linea.....	58

Utilizzare gli elementi dell'elenco quando necessario	59
Evitare l'uso non necessario	59
Elemento di sezione	60
Elemento di intestazione	61
Esempi:	61
Elemento di piè di pagina	61
Capitolo 14: Elementi multimediali	62
Parametri	62
Osservazioni	62
Supporto nei browser	62
Examples	63
Usando ``e` elemento per visualizzare contenuti audio / video	63
Audio	64
video	64
Intestazione o sfondo del video	64
Capitolo 15: Elementi vuoti	66
introduzione	66
Osservazioni	66
Examples	66
Elementi vuoti	66
Capitolo 16: Elemento di progresso	68
Parametri	68
Osservazioni	68
Examples	68
Progresso	68
Modifica del colore di una barra di avanzamento	68
Chrome / Safari / Opera	69
Firefox	69
Internet Explorer	69
Fallback HTML	69
Capitolo 17: Elemento di uscita	71

Parametri.....	71
Examples.....	71
Elemento di output che utilizza gli attributi For e Form.....	71
Elemento di output con attributi.....	72
Capitolo 18: Elemento div.....	73
introduzione.....	73
Sintassi.....	73
Examples.....	73
annidamento.....	73
Utilizzo di base.....	74
Capitolo 19: Elemento etichetta.....	75
Sintassi.....	75
Parametri.....	75
Examples.....	75
Uso di base.....	75
A proposito di etichetta.....	76
Capitolo 20: elenchi.....	77
introduzione.....	77
Sintassi.....	77
Osservazioni.....	77
Examples.....	77
Lista non ordinata.....	77
Lista ordinata.....	78
Modifica manuale dei numeri.....	78
Modifica del tipo di numero.....	79
Elenco di descrizione.....	80
Liste annidate.....	81
Capitolo 21: Entità personaggio.....	82
Examples.....	82
Caratteri speciali comuni.....	82
Entità dei caratteri in HTML.....	83
Capitolo 22: Formattazione del testo.....	84

introduzione	84
Sintassi	84
Examples	84
Grassetto, Corsivo e Sottolineato	84
Testo grassetto	84
Testo in corsivo	85
Testo sottolineato	85
evidenziando	85
Inserito, eliminato o colpito	86
Apice e pedice	86
Abbreviazione	86
Capitolo 23: HTML 5 Cache	87
Osservazioni	87
Examples	87
Esempio di base della cache Html 5	87
Capitolo 24: I paragrafi	88
introduzione	88
Parametri	88
Examples	88
Paragrafi HTML	88
Capitolo 25: IFrame	89
Parametri	89
Osservazioni	89
Politica della stessa origine	89
attributo sandbox	90
Examples	90
Nozioni di base su una cornice in linea	90
Impostazione della dimensione del fotogramma	90
Usando Ancore con IFrames	91
Utilizzando l'attributo "srcdoc"	91
sandboxing	91
Capitolo 26: immagini	93

Sintassi.....	93
Parametri.....	93
Examples.....	93
Creare un'immagine.....	93
Larghezza e altezza dell'immagine.....	94
Scegliere il testo alternativo.....	95
Le note.....	96
Immagine reattiva usando l'attributo srcset.....	96
Usando srcset con le dimensioni.....	96
Utilizzo di srcset senza dimensioni.....	97
Immagine reattiva usando l'elemento dell'immagine.....	97
Capitolo 27: Includi codice JavaScript in HTML.....	98
Sintassi.....	98
Parametri.....	98
Osservazioni.....	98
Examples.....	98
Collegamento a un file JavaScript esterno.....	98
Include direttamente il codice JavaScript.....	99
Incluso un file JavaScript in esecuzione in modo asincrono.....	99
Gestione Javascript disabilitato.....	99
Capitolo 28: Incorporare.....	100
Parametri.....	100
Examples.....	100
Utilizzo di base.....	100
Definire il tipo MIME.....	100
Capitolo 29: Input Control Elements.....	101
introduzione.....	101
Sintassi.....	101
Parametri.....	101
Osservazioni.....	102
Examples.....	103

Casella di controllo e pulsanti di opzione.....	103
Panoramica.....	103
attributi.....	104
value.....	104
checked.....	104
Accessibilità.....	104
etichette.....	104
Gruppi di pulsanti.....	105
Nascosto.....	105
Parola d'ordine.....	106
Sottoscrivi.....	106
File.....	106
Input Validation.....	107
necessario.....	107
Lunghezza minima / massima.....	107
Specifica di un intervallo.....	108
Abbina un modello.....	108
Accetta il tipo di file.....	108
Reset.....	109
Numero.....	109
Tel.....	110
E-mail.....	110
Pulsante.....	110
attributi.....	110
[name].....	110
[type].....	111
[value].....	111
Attributi aggiuntivi per i pulsanti di invio.....	111
Colore.....	111
url.....	112
Data.....	112
DateTime-locale.....	113

Immagine	113
Gamma	113
Mese	113
Tempo	113
Settimana	114
Testo	114
Ricerca	115
DateTime (globale)	115
Capitolo 30: Le forme	116
introduzione	116
Sintassi	116
Parametri	116
Osservazioni	116
Examples	116
Invio	117
L'attributo di azione	117
L'attributo del metodo	117
Più attributi	117
Attributo di destinazione nel tag modulo	117
Caricamento di file	118
Raggruppamento di alcuni campi di input	118
Capitolo 31: Lingue di contenuto	120
Sintassi	120
Osservazioni	120
Accessibilità	120
Examples	120
Element Language	120
Elementi con più lingue	121
Gestione degli attributi con lingue diverse	121
Base Document Language	121
URL regionali	121
Capitolo 32: Mappe di immagini	122

Sintassi.....	122
Parametri.....	122
Osservazioni.....	123
Examples.....	123
Introduzione alle mappe di immagini.....	123
Descrizione	123
Esempio di base.....	123
Capitolo 33: Marcatura del codice del computer.....	125
Sintassi.....	125
Osservazioni.....	125
Elementi correlati.....	125
Examples.....	125
In linea con.....	125
Blocca con e.....	125
Capitolo 34: Meta informazioni.....	127
introduzione.....	127
Sintassi.....	127
Osservazioni.....	127
Examples.....	127
Codifica dei caratteri.....	127
Aggiornamento automatico.....	128
Controllo del layout mobile.....	128
Informazioni sulla pagina.....	129
application-name.....	129
author.....	129
description.....	129
generator.....	130
keywords.....	130
robot.....	130
Riconoscimento numero telefonico.....	131
Social media.....	131
Facebook / Apri grafico.....	131

Facebook / Articoli istantanei.....	132
cinguettio.....	132
Google+ / Schema.org.....	132
Reindirizzamento automatico.....	133
App Web.....	133
Capitolo 35: Quotazioni di marcatura.....	134
Osservazioni.....	134
Examples.....	134
In linea con.....	134
Virgolette.....	134
URL di origine (attributo di cite).....	134
Blocca con.....	134
URL di origine (attributo di cite).....	135
Citation / Attribuzione.....	135
Capitolo 36: SVG.....	136
introduzione.....	136
Osservazioni.....	136
Examples.....	136
Incorporamento di file SVG esterni in HTML.....	136
Utilizzando l'elemento immagine.....	136
Utilizzando l'elemento dell'oggetto.....	136
Inline SVG.....	136
Incorporare SVG usando CSS.....	137
Capitolo 37: tabelle.....	139
introduzione.....	139
Sintassi.....	139
Osservazioni.....	139
Examples.....	140
Tavolo semplice.....	140
Spanning colonne o righe.....	140
Tavolo con thead, tbody, tfoot e didascalia.....	141
Gruppi di colonne.....	143

Ambito di titolo.....	143
Capitolo 38: tabindex.....	146
Parametri.....	146
Osservazioni.....	146
Examples.....	146
Aggiungi un elemento all'ordine di tabulazione.....	146
Rimuovi un elemento dall'ordine di tabulazione.....	146
Definire un ordine di tabulazione personalizzato (non consigliato).....	146
Capitolo 39: Tela.....	148
Parametri.....	148
Osservazioni.....	148
Examples.....	148
Esempio di base.....	148
Disegnare due rettangoli su a.....	148
Capitolo 40: Titoli.....	150
introduzione.....	150
Sintassi.....	150
Osservazioni.....	150
Examples.....	150
Usando le intestazioni.....	150
La struttura corretta conta.....	151
Capitolo 41: Usare l'HTML con i CSS.....	152
introduzione.....	152
Sintassi.....	152
Examples.....	152
Uso del foglio di stile esterno.....	152
Foglio di stile interno.....	152
Stile in linea.....	153
Fogli di stile multipli.....	153
Titoli di coda.....	155

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

It is an unofficial and free HTML 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 HTML.

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

Capitolo 1: Iniziare con HTML

Osservazioni

HTML (**H** yper **t** ext **M** arkup **L** anguage) è un sistema compatibile con **XML** di annotazione di documenti con 'tag'. Viene utilizzato specificamente per creare contenuti per pagine Web e applicazioni Web, che possono quindi essere condivisi su una rete.

Oltre al testo, l'attuale versione di HTML supporta molti **tipi** diversi **di media** , incluse immagini e video.

Versioni

Versione	specificazione	Data di rilascio
1.0	N / A	1994-01-01
2.0	RFC 1866	1995/11/24
3.2	W3C: specifica HTML 3.2	1997/01/14
4.0	W3C: specifica HTML 4.0	1998/04/24
4.01	W3C: specifica HTML 4.01	1999/12/24
5	WHATWG: HTML Living Standard	2014/10/28
5.1	W3C: specifica HTML 5.1	2016/11/01

Examples

Ciao mondo

introduzione

HTML (**H** yper **t** ext **M** arkup **L** anguage) utilizza un sistema di markup composto da elementi che rappresentano contenuti specifici. *Markup* significa che con HTML si dichiara *ciò che* viene presentato a un visualizzatore, non *come* viene presentato. Le rappresentazioni visive sono definite da **Cascading Style Sheets (CSS)** e realizzate dai browser. **Gli elementi ancora esistenti che consentono tale** , come ad esempio il `font` , "sono completamente obsoleti e non devono essere usati dagli autori" ^[1] .

L'HTML è talvolta chiamato un linguaggio di programmazione ma non ha logica, quindi è un

linguaggio di markup . I tag HTML forniscono significato semantico e leggibilità automatica al contenuto della pagina.

Un elemento di solito consiste in un tag di apertura (`<element_name>`), un tag di chiusura (`</element_name>`), che contiene il nome dell'elemento circondato da parentesi angolari e il contenuto tra: `<element_name>...content...</element_name>`

Ci sono alcuni elementi HTML che non hanno un tag di chiusura o qualsiasi contenuto. Questi sono chiamati **elementi void** . Gli elementi vuoti includono `<img>` , `<meta>` , `<link>` e `<input>` .

I nomi degli elementi possono essere considerati parole chiave descrittive per il contenuto che contengono, ad esempio `video` , `audio` , `table` , `footer` .

Una pagina HTML può contenere potenzialmente centinaia di elementi che vengono poi letti da un browser Web, interpretati e resi in contenuti leggibili o udibili sullo schermo.

Per questo documento è importante notare la differenza tra elementi e tag:

Elementi: `video` , `audio` , `table` , `footer`

Tag: `<video>` , `<audio>` , `<table>` , `<footer>` , `</html>` , `</body>`

Intuizione dell'elemento

Analizziamo un tag ...

Il tag `<p>` rappresenta un paragrafo comune.

Gli elementi hanno in genere un tag di apertura e un tag di chiusura. Il tag di apertura contiene il nome dell'elemento tra parentesi angolari (`<p>`). Il tag di chiusura è identico al tag di apertura con l'aggiunta di una barra (`/`) tra la parentesi aperta e il nome dell'elemento (`</p>`).

Il contenuto può quindi passare tra questi due tag: `<p>This is a simple paragraph.</p>` .

Creare una semplice pagina

Il seguente esempio HTML crea una semplice pagina Web "Hello World" .

I file HTML possono essere creati utilizzando qualsiasi **editor di testo** . I file devono essere salvati con un'estensione `.html` o `.htm` ^[2] per essere riconosciuti come file HTML.

Una volta creato, questo file può essere aperto in qualsiasi browser web.

```
<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="UTF-8">
    <title>Hello!</title>
  </head>

  <body>
    <h1>Hello World!</h1>
    <p>This is a simple paragraph.</p>
  </body>

</html>
```

Semplice pagina abbattere

Questi sono i tag utilizzati nell'esempio:

Etichetta	Senso
<!DOCTYPE>	Definisce la versione HTML utilizzata nel documento. In questo caso è HTML5. Vedere l' argomento doctypes per ulteriori informazioni.
<html>	Apri la pagina. Nessun markup dovrebbe venire dopo il tag di chiusura (</html>). L'attributo <code>lang</code> dichiara la lingua principale della pagina utilizzando i codici lingua ISO (<code>en</code> per inglese). Vedere l' argomento della lingua del contenuto per ulteriori informazioni.
<head>	Apri la sezione head, che non appare nella finestra principale del browser ma contiene principalmente informazioni <i>sul</i> documento HTML, chiamate <i>metadati</i> . Può anche contenere importazioni da fogli di stile e script esterni. Il tag di chiusura è </head> .
<meta>	Fornisce al browser alcuni metadati relativi al documento. L'attributo <code>charset</code> dichiara la codifica dei caratteri . I documenti HTML moderni dovrebbero sempre utilizzare UTF-8 , anche se non è un requisito. In HTML, il tag <meta> non richiede un tag di chiusura. Vedi l' argomento Meta per maggiori informazioni.
<title>	Il titolo della pagina. Il testo scritto tra questa apertura e il tag di chiusura (</title>) verrà visualizzato nella scheda della pagina o nella barra del titolo del browser.
<body>	Apri la parte del documento visualizzata agli utenti, ovvero tutto il contenuto visibile o udibile di una pagina. Nessun contenuto dovrebbe essere aggiunto

Etichetta	Senso
	dopo il tag di chiusura <code></body></code> .
<code><h1></code>	Un'intestazione di livello 1 per la pagina. Vedi i titoli per maggiori informazioni.
<code><p></code>	Rappresenta un paragrafo di testo comune.

1. ↑ [HTML5, 11.2 Funzioni non conformi](#)
2. ↑ `.htm` è ereditato dal limite di estensione del file di tre caratteri legacy [DOS](#) .

Leggi Iniziare con HTML online: <https://riptutorial.com/it/html/topic/217/iniziare-con-html>

Capitolo 2: Ancore e collegamenti ipertestuali

introduzione

I tag di ancoraggio sono comunemente usati per collegare pagine web separate, ma possono anche essere usati per collegare tra luoghi diversi in un singolo documento, spesso all'interno del sommario o persino avviare applicazioni esterne. Questo argomento spiega l'implementazione e l'applicazione dei tag di ancoraggio HTML in vari ruoli.

Sintassi

- `<a href="URL or anchor">Link Text</a>`

Parametri

Parametro	Dettagli
<code>href</code>	Specifica l'indirizzo di destinazione. Può essere un URL assoluto o relativo o il <code>name</code> di un'ancora. Un URL assoluto è l'URL completo di un sito web come http://example.com/ . Un URL relativo punta a un'altra directory e / o documento all'interno dello stesso sito web, ad esempio <code>/about-us/</code> punta alla directory "about-us" all'interno della directory root (<code>/</code>). Quando si punta a un'altra directory senza specificare esplicitamente il documento, i server Web in genere restituiscono il documento "index.html" all'interno di tale directory.
<code>hreflang</code>	Specifica la lingua della risorsa collegata dall'attributo <code>href</code> (che deve essere presente con questo attributo). Utilizza i valori della lingua da BCP 47 per HTML5 e RFC 1766 per HTML 4.
<code>rel</code>	Specifica la relazione tra il documento corrente e il documento collegato. Per HTML5, i valori devono essere definiti nella specifica o registrati nel wiki Microformats .
<code>target</code>	Specifica dove aprire il collegamento, ad esempio in una nuova scheda o finestra. I valori possibili sono <code>_blank</code> , <code>_self</code> , <code>_parent</code> , <code>_top</code> e <code>iframe</code> (deprecato). Forzare tale comportamento non è raccomandato in quanto viola il controllo dell'utente su un sito web.
<code>title</code>	Specifica informazioni aggiuntive su un collegamento. Le informazioni vengono spesso visualizzate come testo di descrizione quando il cursore si sposta sul collegamento. Questo attributo non è limitato ai link, può essere utilizzato su quasi tutti i tag HTML.
<code>download</code>	Specifica che il target verrà scaricato quando un utente fa clic sul collegamento ipertestuale. Il valore dell'attributo sarà il nome del file scaricato. Non ci sono

Parametro	Dettagli
	restrizioni sui valori consentiti e il browser rileva automaticamente l'estensione corretta del file e la aggiunge al file (.img, .pdf, ecc.). Se il valore è omissso, viene utilizzato il nome file originale.

Examples

Link a un altro sito

Questo è l'uso di base dell'elemento `<a>` ([un elemento ncur](#)) :

```
<a href="http://example.com/">Link to example.com</a>
```

Crea un collegamento ipertestuale, all'URL `http://example.com/` come specificato dall'attributo `href` (riferimento ipertestuale), con il testo di ancoraggio "Link a example.com". Sarebbe simile al seguente:

[Link a example.com](http://example.com/)

Per indicare che questo collegamento porta a un sito Web esterno, è possibile utilizzare il tipo di collegamento `external` :

```
<a href="http://example.com/" rel="external">example site</a>
```

È possibile collegare a un sito che utilizza un protocollo diverso da HTTP. Ad esempio, per collegare a un sito FTP, puoi farlo,

```
<a href="ftp://example.com/">This could be a link to a FTP site</a>
```

In questo caso, la differenza è che questo tag di ancoraggio richiede che il browser dell'utente si colleghi a `example.com` utilizzando il protocollo FTP (File Transfer Protocol) anziché HTTP (Hypertext Transfer Protocol).

[Questo potrebbe essere un collegamento a un sito FTP](#)

Apri il link in una nuova scheda / finestra

```
<a href="example.com" target="_blank">Text Here</a>
```

L'attributo `target` specifica dove aprire il link. Impostandolo su `_blank` , dici al browser di aprirlo in una nuova scheda o finestra (per preferenza dell'utente).

VULNERABILITÀ DI SICUREZZA ATTENZIONE!

L'utilizzo di `target="_blank"` fornisce al sito di apertura l'accesso parziale all'oggetto

`window.opener` tramite JavaScript, che consente a tale pagina di accedere e modificare la proprietà `window.opener.location` della *propria* pagina e potenzialmente di reindirizzare gli utenti a malware o siti di phishing.

Ogni volta che usi questo per pagine che non controlli, aggiungi `rel="noopener"` al tuo link per impedire che l'oggetto `window.opener` venga inviato con la richiesta.

Attualmente, Firefox non supporta il `noopener`, quindi è necessario utilizzare `rel="noopener noreferrer"` per ottenere il massimo effetto.

Collegamento a un'ancora

Le ancore possono essere utilizzate per passare a tag specifici su una pagina HTML. Il tag `<a>` può puntare a qualsiasi elemento che abbia un attributo `id`. Per ulteriori informazioni sugli ID, consulta la [documentazione su Classi e ID](#). Le ancore sono principalmente utilizzate per passare a una sottosezione di una pagina e vengono utilizzate insieme ai tag di intestazione.

Supponi di aver creato una pagina (`page1.html`) su molti argomenti:

```
<h2>First topic</h2>
<p>Content about the first topic</p>
<h2>Second topic</h2>
<p>Content about the second topic</p>
```

Una volta che hai diverse sezioni, potresti voler creare un sommario nella parte superiore della pagina con collegamenti rapidi (o segnalibri) a sezioni specifiche.

Se hai dato un attributo `id` ai tuoi argomenti, potresti quindi collegarti a loro

```
<h2 id="Topic1">First topic</h2>
<p>Content about the first topic</p>
<h2 id="Topic2">Second topic</h2>
<p>Content about the second topic</p>
```

Ora puoi usare l'ancora nel tuo sommario:

```
<h1>Table of Contents</h1>
<a href="#Topic1">Click to jump to the First Topic</a>
<a href="#Topic2">Click to jump to the Second Topic</a>
```

Questi ancoraggi sono anche collegati alla pagina Web in cui si trovano (`page1.html`). Quindi puoi collegare il sito da una pagina all'altra facendo riferimento alla pagina e al nome dell'ancora.

```
Remember, you can always <a href="page1.html#Topic1">look back in the First Topic</a> for
supporting information.
```

Link che esegue JavaScript

Basta usare il `javascript:` protocollo per eseguire il testo come JavaScript invece di aprirlo come

un normale link:

```
<a href="javascript:myFunction();">Run Code</a>
```

Puoi anche ottenere la stessa cosa usando l'attributo `onclick` :

```
<a href="#" onclick="myFunction(); return false;">Run Code</a>
```

Il `return false;` è necessario per impedire alla pagina di scorrere verso l'alto quando si fa clic sul collegamento a `#` . Assicurati di includere tutto il codice che desideri eseguire prima di esso, poiché il ritorno interromperà l'esecuzione di ulteriori codici.

Inoltre, è possibile includere un punto esclamativo `!` dopo l'hashtag per impedire alla pagina di scorrere verso l'alto. Funziona perché qualsiasi slug non valido farà sì che il link non scorra in alcun *punto* della pagina, perché non è in grado di individuare l'elemento a cui fa riferimento (un elemento con `id="!"`). Puoi anche utilizzare qualsiasi slug non valido (come `#scrollsNowhere`) per ottenere lo stesso effetto. In questo caso, `return false;` non è richiesto:

```
<a href="#!" onclick="myFunction();">Run Code</a>
```

Dovresti usare qualcosa di questo?

La risposta è quasi certamente **no** . L'esecuzione di JavaScript in linea con l'elemento come questo è una pratica abbastanza negativa. Prendi in considerazione l'utilizzo di soluzioni JavaScript pure che cercano l'elemento nella pagina e associano invece una funzione ad esso. [Ascoltando un evento](#)

Considera anche se questo elemento è in realtà un *pulsante* anziché un *link* . Se è così, dovresti usare `<button>` .

Link a una pagina sullo stesso sito

È possibile utilizzare un [percorso relativo](#) per collegarsi a pagine sullo stesso sito Web.

```
<a href="/example">Text Here</a>
```

L'esempio precedente passerebbe al file `example` nella directory root (`/`) del server.

Se questo link era su <http://example.com> , i seguenti due link porterebbero l'utente nella stessa posizione

```
<a href="/page">Text Here</a>
<a href="http://example.com/page">Text Here</a>
```

Entrambi di quanto sopra sarebbe andato alla `page` file alla directory principale di `example.com` .

Link che esegue il client di posta elettronica

Utilizzo di base

Se il valore del `href`-attribute inizia con `mailto:` si tenta di aprire un client di posta elettronica al clic:

```
<a href="mailto:example@example.com">Send email</a>
```

Ciò inserirà l'indirizzo email `example@example.com` come destinatario per l'email appena creata.

Cc e Ccn

È inoltre possibile aggiungere indirizzi per i destinatari cc o bcc utilizzando la seguente sintassi:

```
<a href="mailto:example@example.com?cc=john@example.com&bcc=jane@example.com">Send email</a>
```

Oggetto e testo del corpo

Puoi popolare l'oggetto e il corpo anche per la nuova email:

```
<a href="mailto:example@example.com?subject=Example+subject&body=Message+text">Send email</a>
```

Questi valori devono essere [codificati URL](#).

Cliccando su un link con `mailto:` tenterà di aprire il client email predefinito specificato dal tuo sistema operativo o ti chiederà di scegliere quale client desideri utilizzare. Non tutte le opzioni specificate dopo l'indirizzo del destinatario sono supportate in tutti i client di posta elettronica.

Link che compone un numero

Se il valore del `href`-attribute inizia con `tel:`, il dispositivo comporrà il numero, quando si fa clic su di esso. Funziona su dispositivi mobili o su computer / tablet con software in esecuzione, come Skype o FaceTime, che possono effettuare chiamate telefoniche.

```
<a href="tel:11234567890">Call us</a>
```

La maggior parte dei dispositivi e dei programmi richiederà all'utente in qualche modo di confermare il numero che stanno per comporre.

Leggi Ancora e collegamenti ipertestuali online: <https://riptutorial.com/it/html/topic/254/ancore-e-collegamenti-ipertestuali>

Capitolo 3: ARIA

Sintassi

- Aria-live
- Aria rilevanti
- Aria-completamento automatico
- Aria-controllato
- Aria-disabled
- Aria-espanso
- Aria-haspopup
- Aria-nascosto
- aria-non valida
- Aria-label
- Aria-level
- Aria-multilinea
- Aria-multiselectable
- aria-orientamento
- Aria-pressato
- Aria-in sola lettura
- Aria-required
- Aria-selezionati
- Aria-Sort
- Aria-valuemax
- Aria-valuemin
- Aria-valuenow
- Aria-valuetext
- Aria-atomica
- Aria-occupato
- Aria-dropEffect
- aria-trascinato
- aria-activedescendant
- Aria-controlli
- Aria-describedby
- Aria-flowto
- Aria-labelledby
- Aria-proprietaria
- Aria-posinset
- Aria-setSize

Osservazioni

ARIA è una specifica per descrivere semanticamente applicazioni web ricche. I seguenti standard ARIA possono aumentare l'accessibilità per coloro che utilizzano tecnologie assistive (come uno

screen reader) per accedere ai tuoi contenuti.

Examples

role = "alert"

Un messaggio con informazioni importanti e solitamente sensibili al tempo.

```
<div role="alert" aria-live="assertive">Your session will expire in 60 seconds.</div>
```

Nota che ho incluso sia `role="alert"` che `aria-live="assertive"` allo stesso tempo. Questi sono sinonimi, ma alcuni screen reader supportano solo l'uno o l'altro. Usando entrambi contemporaneamente, massimizziamo quindi le possibilità che la regione live funzioni come previsto.

Fonte - Heydon Pickering ["Alcuni esempi pratici di ARIA"](#)

role = "alertdialog"

Un tipo di finestra di dialogo che contiene un messaggio di avviso, in cui l'attenzione iniziale viene spostata su un elemento all'interno della finestra di dialogo.

```
<div role="alertdialog">
  <h1>Warning</h1>
  <div role="alert">Your session will expire in 60 seconds.</div>
</div>
```

role = "applicazione"

Una regione dichiarata come applicazione web, al contrario di un documento Web. In questo esempio, l'applicazione è una semplice calcolatrice che potrebbe aggiungere due numeri insieme.

```
<div role="application">
  <h1>Calculator</h1>
  <input id="num1" type="text"> + <input id="num2" type="text"> =
  <span id="result"></span>
</div>
```

role = "articolo"

Una sezione di una pagina composta da una composizione che costituisce una parte indipendente di un documento, una pagina o un sito.

L'impostazione di un attributo ARIA e / o `aria-*` che corrisponde alla semantica implicita ARIA predefinita non è necessaria e non è consigliata in quanto queste proprietà sono già impostate dal browser.

```
<article>
  <h1>My first article</h1>
  <p>Lorem ipsum...</p>
</article>
```

Dovresti usare `role=article` su elementi non semantici (non raccomandato, non valido)

```
<div role="article">
  <h1>My first article</h1>
  <p>Lorem ipsum...</p>
</div>
```

Voce W3C per [role=article](#)

role = "banner"

Una regione che contiene principalmente contenuti orientati al sito, piuttosto che contenuti specifici della pagina.

```
<div role="banner">
  <h1>My Site</h1>

  <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/about">About</a></li>
    <li><a href="/contact">Contact</a></li>
  </ul>
</div>
```

role = "button"

Un input che consente azioni attivate dall'utente quando si fa clic o si preme.

```
<button role="button">Add</button>
```

role = "cella"

Una cella in un contenitore tabulare.

```
<table>
  <thead>
    <!-- etc -->
  </thead>
  <tbody>
    <td role="cell">95</td>
    <td role="cell">14</td>
    <td role="cell">25</td>
  </tbody>
</table>
```

role = "checkbox"

Un input controllabile che ha tre possibili valori: true, false o mixed.

```
<p>
  <input type="checkbox" role="checkbox" aria-checked="false">
  I agree to the terms
</p>
```

role = "columnheader"

Una cella contenente le informazioni di intestazione per una colonna.

```
<table role="grid">
  <thead>
    <tr>
      <th role="columnheader">Day 1</th>
      <th role="columnheader">Day 2</th>
      <th role="columnheader">Day 3</th>
    </tr>
  </thead>
  <tbody>
    <!-- etc -->
  </tbody>
</table>
```

role = "Casella combinata"

Una presentazione di una selezione; di solito simile a una casella di testo in cui gli utenti possono digitare avanti per selezionare un'opzione, o digitare per inserire testo arbitrario come nuovo elemento nell'elenco.

```
<input type="text" role="combobox" aria-expanded="false">
```

In genere, si utilizza JavaScript per creare il resto della funzionalità typeahead o list select.

role = "complementare"

Una sezione di supporto del documento, progettata per essere complementare al contenuto principale a un livello simile nella gerarchia DOM, ma rimane significativa se separata dal contenuto principale.

```
<div role="complementary">
  <h2>More Articles</h2>

  <ul>
    <!-- etc -->
  </ul>
</div>
```

role = "ContentInfo"

Un'ampia regione percepibile che contiene informazioni sul documento principale.

```
<p role="contentinfo">
  Author: Albert Einstein<br>
  Published: August 15, 1940
</p>
```

role = "definizione"

Una definizione di un termine o concetto.

```
<span role="term" aria-labelledby="def1">Love</span>
<span id="def1" role="definition">an intense feeling of deep affection.</span>
```

role = "finestra"

Una finestra di dialogo è una finestra dell'applicazione progettata per interrompere l'elaborazione corrente di un'applicazione al fine di richiedere all'utente di immettere informazioni o richiedere una risposta.

```
<div role="dialog">
  <p>Are you sure?</p>
  <button role="button">Yes</button>
  <button role="button">No</button>
</div>
```

role = "directory"

Un elenco di riferimenti ai membri di un gruppo, ad esempio un indice statico.

```
<ul role="directory">
  <li><a href="/chapter-1">Chapter 1</a></li>
  <li><a href="/chapter-2">Chapter 2</a></li>
  <li><a href="/chapter-3">Chapter 3</a></li>
</ul>
```

role = "documento"

Una regione contenente informazioni correlate dichiarate come contenuto del documento, a differenza di un'applicazione web.

```
<div role="document">
  <h1>The Life of Albert Einstein</h1>
  <p>Lorem ipsum...</p>
</div>
```

role = "forma"

Una regione di riferimento che contiene una raccolta di oggetti e oggetti che, nel complesso, si combinano per creare un modulo.

L'utilizzo dell'elemento HTML semanticamente corretto `<form>` implica la semantica ARIA predefinita, ovvero `role=form` non è richiesto in quanto non si dovrebbe applicare un ruolo di contrasto a un elemento che è già semantico, poiché l'aggiunta di un ruolo sovrascrive la semantica nativa di un elemento.

L'impostazione di un attributo ARIA e / o `aria-*` che corrisponde alla semantica implicita ARIA predefinita non è necessaria e non è consigliata in quanto queste proprietà sono già impostate dal browser.

```
<form action="">
  <fieldset>
    <legend>Login form</legend>
    <div>
      <label for="username">Your username</label>
      <input type="text" id="username" aria-describedby="username-tip" required />
      <div role="tooltip" id="username-tip">Your username is your email address</div>
    </div>
    <div>
      <label for="password">Your password</label>
      <input type="text" id="password" aria-describedby="password-tip" required />
      <div role="tooltip" id="password-tip">Was emailed to you when you signed up</div>
    </div>
  </fieldset>
</form>
```

Dovresti usare `role=form` su elementi non semantici (non raccomandato, non valido)

```
<div role=form>
  <input type="email" placeholder="Your email address">
  <button>Sign up</button>
</div>
```

role = "griglia"

Una griglia è un controllo interattivo che contiene celle di dati tabulari disposti in righe e colonne, come una tabella.

```
<table role="grid">
  <thead>
    <!-- etc -->
  </thead>
  <tbody>
    <!-- etc -->
  </tbody>
</table>
```

role = "gridcell"

Una cella in una griglia o treegrida.

```
<table role="grid">
  <thead>
    <!-- etc -->
```



```
</thead>
<tbody>
  <tr>
    <td role="gridcell">17</td>
    <td role="gridcell">64</td>
    <td role="gridcell">18</td>
  </tr>
</tbody>
<table>
```

role = "gruppo"

Un insieme di oggetti dell'interfaccia utente che non devono essere inclusi in un riepilogo o in un sommario della pagina mediante tecnologie assistive.

```
<div role="group">
  <button role="button">Previous</button>
  <button role="button">Next</button>
</div>
```

role = "heading"

Un'intestazione per una sezione della pagina.

```
<h1 role="heading">Introduction</h1>
<p>Lorem ipsum...</p>
```

role = "img"

Un contenitore per una raccolta di elementi che formano un'immagine.

```
<figure role="img">
  
  <figcaption>This is my cat, Albert.</figcaption>
</figure>
```

role = "link"

Un riferimento interattivo a una risorsa interna o esterna che, quando attivata, fa sì che l'interprete utilizzi tale risorsa.

Nella maggior parte dei casi l'impostazione di un attributo ARIA e / o di un attributo aria- * corrispondente alla [semantica implicita di ARIA](#) non è necessaria e non è consigliata in quanto tali proprietà sono già impostate dal browser.

Fonte: <https://www.w3.org/TR/html5/dom.html#aria-usage-note>

role = "lista"

Un gruppo di voci di elenco non interattive.

```
<ul role="list">
  <li role="listitem">One</li>
  <li role="listitem">Two</li>
  <li role="listitem">Three</li>
</ul>
```

role = "casella di riepilogo"

Un widget che consente all'utente di selezionare uno o più elementi da un elenco di scelte.

```
<ul role="listbox">
  <li>One</li>
  <li>Two</li>
  <li>Three</li>
</ul>
```

In genere, si utilizza JavaScript per creare la funzionalità di selezione multipla.

role = "listitem"

Un singolo elemento in un elenco o directory.

```
<ul role="list">
  <li role="listitem">One</li>
  <li role="listitem">Two</li>
  <li role="listitem">Three</li>
</ul>
```

role = "log"

Un tipo di area live in cui vengono aggiunte nuove informazioni in ordine significativo e le vecchie informazioni potrebbero scomparire.

```
<ul role="log">
  <li>User 1 logged in.</li>
  <li>User 2 logged in.</li>
  <li>User 1 logged out.</li>
</ul>
```

role = "main"

Il contenuto principale di un documento.

```
<!-- header & nav here -->
<div role="main">
  <p>Lorem ipsum...</p>
</div>
<!-- footer here -->
```

role = "tendone"

Un tipo di regione in cui le informazioni non essenziali cambiano frequentemente.

```
<ul role="marquee">
  <li>Dow +0.26%</li>
  <li>Nasdaq +0.54%</li>
  <li>S&P +0.44%</li>
</ul>
```

role = "matematica"

Contenuto che rappresenta un'espressione matematica.

```

```

role = "menu"

Un tipo di widget che offre un elenco di scelte all'utente.

```
<ul role="menu">
  <li role="menuitem">New</li>
  <li role="menuitem">Open</li>
  <li role="menuitem">Save</li>
  <li role="menuitem">Close</li>
</ul>
```

role = "barra dei menu"

Una presentazione di menu che di solito rimane visibile e di solito viene presentata orizzontalmente.

```
<ul role="menubar">
  <li role="menuitem">File</li>
  <li role="menuitem">Edit</li>
  <li role="menuitem">View</li>
  <li role="menuitem">Help</li>
</ul>
```

role = "menuitem"

Un'opzione in un gruppo di scelte contenute in un menu o in una barra dei menu.

```
<ul role="menubar">
  <li role="menuitem">File</li>
  <li role="menuitem">Edit</li>
  <li role="menuitem">View</li>
  <li role="menuitem">Help</li>
</ul>
```

role = "menuitemcheckbox"

Un menu di menu controllabile che ha tre possibili valori: vero, falso o misto.

```
<ul role="menu">
  <li role="menuitem">Console</li>
  <li role="menuitem">Layout</li>
  <li role="menuitemcheckbox" aria-checked="true">Word wrap</li>
</ul>
```

role = "menuitemradio"

Un menu di menu controllabile in un gruppo di ruoli menuitemradio, solo uno dei quali può essere controllato alla volta.

```
<ul role="menu">
  <li role="menuitemradio" aria-checked="true">Left</li>
  <li role="menuitemradio" aria-checked="false">Center</li>
  <li role="menuitemradio" aria-checked="false">Right</li>
</ul>
```

role = "navigazione"

Una raccolta di elementi di navigazione (in genere collegamenti) per la navigazione del documento o dei documenti correlati.

```
<ul role="navigation">
  <li><a href="/">Home</a></li>
  <li><a href="/about">About</a></li>
  <li><a href="/contact">Contact</a></li>
</ul>
```

role = "note"

Una sezione il cui contenuto è parentetico o accessorio al contenuto principale della risorsa.

```
<p>Lorem ipsum...</p>
<p>Lorem ipsum...</p>
<p role="note">Lorem ipsum...</p>
```

role = "opzione"

Un elemento selezionabile in un elenco selezionato.

```
<ul role="listbox">
  <li role="option">Option 1</li>
  <li role="option">Option 2</li>
  <li role="option">Option 3</li>
</ul>
```

role = "presentazione"

Un elemento la cui semantica implicita del ruolo nativo non verrà mappata all'API di accessibilità.

```
<div style="float:left;">Some content on the left.</div>
<div style="float:right;">Some content on the right</div>
<div role="presentation" style="clear:both;"></div> <!-- Only used to clear floats -->
```

role = "progressbar"

Un elemento che visualizza lo stato di avanzamento delle attività che richiedono molto tempo.

```
<progress role="progressbar" value="25" max="100">25%</progress>
```

role = "radio"

Un input controllabile in un gruppo di ruoli radio, di cui solo uno può essere controllato alla volta.

```
<div role="radiogroup">
  <input role="radio" type="radio" aria-checked="true"> One<br>
  <input role="radio" type="radio" aria-checked="false"> Two<br>
  <input role="radio" type="radio" aria-checked="false"> Three
</div>
```

role = "regione"

Una grande sezione percepibile di una pagina web o di un documento, che l'autore ritiene sia abbastanza importante da essere inclusa in un riepilogo della pagina o in un sommario, ad esempio un'area della pagina contenente statistiche di eventi sportivi in diretta.

```
<div role="region">
  Home team: 4<br>
  Away team: 2
</div>
```

role = "RadioGroup"

Un gruppo di pulsanti radio.

```
<div role="radiogroup">
  <input role="radio" type="radio" aria-checked="true"> One<br>
  <input role="radio" type="radio" aria-checked="false"> Two<br>
  <input role="radio" type="radio" aria-checked="false"> Three
</div>
```

role = "riga"

Una fila di celle in un contenitore tabulare.

```
<table>
  <thead>
```

```

    <!-- etc -->
</thead>
<tbody>
  <tr role="row">
    <!-- etc -->
  </tr>
</tbody>
</table>

```

role = "rowgroup"

Un gruppo contenente uno o più elementi riga in una griglia.

```

<table>
  <thead role="rowgroup">
    <!-- etc -->
  </thead>
  <tbody role="rowgroup">
    <!-- etc -->
  </tbody>
</table>

```

role = "rowheader"

Una cella contenente informazioni di intestazione per una riga in una griglia.

```

<table role="grid">
  <thead>
    <!-- etc -->
  </thead>
  <tbody>
    <tr>
      <th role="rowheader">Day 1</th>
      <td>65</td>
    </tr>
    <tr>
      <th role="rowheader">Day 2</th>
      <td>74</td>
    </tr>
  </tbody>
</table>

```

role = "barra di scorrimento"

Un oggetto grafico che controlla lo scorrimento del contenuto all'interno di un'area di visualizzazione, indipendentemente dal fatto che il contenuto sia completamente visualizzato nell'area di visualizzazione.

```

<div id="content1">Lorem ipsum...</div>
<div
  role="scrollbar"
  aria-controls="content1"
  aria-orientation="vertical"
  aria-valuemax="100"

```

```
aria-valuemin="0"  
aria-valuenow="25">  
  <div class="scrollhandle"></div>  
</div>
```

role = "search"

Una regione di riferimento che contiene una raccolta di oggetti e oggetti che, nel complesso, si combinano per creare una funzione di ricerca.

```
<div role="search">  
  <input role="searchbox" type="text">  
  <button role="button">Search</button>  
</div>
```

role = "searchbox"

Un tipo di casella di testo destinato a specificare i criteri di ricerca.

```
<div role="search">  
  <input role="searchbox" type="text">  
  <button role="button">Search</button>  
</div>
```

role = "separatore"

Un divisore che separa e distingue sezioni di contenuto o gruppi di menu.

```
<p>Lorem ipsum...</p>  
<hr role="separator">  
<p>Lorem ipsum...</p>
```

role = "slider"

Un input dell'utente in cui l'utente seleziona un valore all'interno di un determinato intervallo.

```
<div  
  role="slider"  
  aria-valuemax="100"  
  aria-valuemin="0"  
  aria-valuenow="25">  
  <div class="sliderhandle"></div>  
</div>
```

role = "SpinButton"

Una forma di intervallo che si aspetta che l'utente scelga tra le scelte discrete.

```
<input  
  role="spinbutton">
```

```
aria-valuemax="100"
aria-valuemin="0"
aria-valuenow="25"
type="number"
value="25">
```

role = "Stato"

Un contenitore il cui contenuto è un'informazione di consulenza per l'utente ma non è abbastanza importante da giustificare un avviso, spesso ma non necessariamente presentato come barra di stato.

```
<div role="status">Online</div>
```

role = "interruttore"

Un tipo di checkbox che rappresenta valori on / off, a differenza dei valori selezionati / non selezionati.

```
<select role="switch" aria-checked="false">
  <option>On</option>
  <option selected>Off</option>
</select>
```

role = "tab"

Un'etichetta di raggruppamento che fornisce un meccanismo per selezionare il contenuto della scheda che deve essere reso all'utente.

```
<ul role="tablist">
  <li role="tab">Introduction</li>
  <li role="tab">Chapter 1</li>
  <li role="tab">Chapter 2</li>
</ul>
```

role = "tavolo"

Una sezione contenente dati disposti in righe e colonne. Il ruolo della tabella è inteso per i contenitori tabulari che non sono interattivi.

```
<table role="table">
  <thead>
    <!-- etc -->
  </thead>
  <tbody>
    <!-- etc -->
  </tbody>
</table>
```

role = "tablist"

Un elenco di elementi di tabulazione, che sono riferimenti a elementi di tabpanel.

```
<ul role="tablist">
  <li role="tab">Introduction</li>
  <li role="tab">Chapter 1</li>
  <li role="tab">Chapter 2</li>
</ul>
```

role = "tabpanel"

Un contenitore per le risorse associate a una scheda, in cui ogni scheda è contenuta in un elenco di schede.

```
<ul role="tablist">
  <li role="tab">Introduction</li>
  <li role="tab">Chapter 1</li>
  <li role="tab">Chapter 2</li>
</ul>
<div role="tabpanel">
  <!-- etc -->
</div>
```

role = "testo"

Input che consente al testo in forma libera il suo valore.

```
<textarea role="textbox"></textarea>
```

role = "timer"

Un tipo di regione attiva contenente un contatore numerico che indica una quantità di tempo trascorso da un punto iniziale o il tempo rimanente fino a un punto finale.

```
<p>
  <span role="timer">60</span> seconds remaining.
</p>
```

role = "barra degli strumenti"

Una raccolta di pulsanti di funzione comunemente utilizzati rappresentati in forma visiva compatta.

```
<ul role="toolbar">
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

role = "tooltip"

Un popup contestuale che visualizza una descrizione per un elemento.

```
<span aria-describedby="slopedesc">Slope</span>
<div role="tooltip" id="slopedesc">y=mx+b</div>
```

In genere, il suggerimento sarebbe nascosto. Usando JavaScript, il suggerimento sarebbe visualizzato dopo un ritardo quando l'utente passa sopra l'elemento che descrive.

role = "albero"

Un tipo di elenco che può contenere gruppi nidificati di livello inferiore che possono essere compressi ed espansi.

```
<ul role="tree">
  <li role="treeitem">
    Part 1
    <ul>
      <li role="treeitem">Chapter 1</li>
      <li role="treeitem">Chapter 2</li>
      <li role="treeitem">Chapter 3</li>
    </ul>
  </li>
  <li role="treeitem">
    Part 2
    <ul>
      <li role="treeitem">Chapter 4</li>
      <li role="treeitem">Chapter 5</li>
      <li role="treeitem">Chapter 6</li>
    </ul>
  </li>
  <li role="treeitem">
    Part 3
    <ul>
      <li role="treeitem">Chapter 7</li>
      <li role="treeitem">Chapter 8</li>
      <li role="treeitem">Chapter 9</li>
    </ul>
  </li>
</ul>
```

role = "treegrid"

Una griglia le cui righe possono essere espanse e compresse nello stesso modo di un albero.

role = "treeitem"

Una voce di opzione di un albero. Questo è un elemento all'interno di un albero che può essere espanso o compresso se contiene un gruppo sottolivello di alberi.

```
<ul role="tree">
  <li role="treeitem">
    Part 1
    <ul>
      <li role="treeitem">Chapter 1</li>
```

```
    <li role="treeitem">Chapter 2</li>
    <li role="treeitem">Chapter 3</li>
  </ul>
</li>
<li role="treeitem">
  Part 2
  <ul>
    <li role="treeitem">Chapter 4</li>
    <li role="treeitem">Chapter 5</li>
    <li role="treeitem">Chapter 6</li>
  </ul>
</li>
<li role="treeitem">
  Part 3
  <ul>
    <li role="treeitem">Chapter 7</li>
    <li role="treeitem">Chapter 8</li>
    <li role="treeitem">Chapter 9</li>
  </ul>
</li>
</ul>
```

Leggi ARIA online: <https://riptutorial.com/it/html/topic/2734/aria>

Capitolo 4: Attributi dei dati

Sintassi

- `<element data-custom-name="somevalue">`

Parametri

Valore	Descrizione
somevalue	Specifica il valore dell'attributo (come stringa)

Examples

Uso degli attributi dei dati

Gli attributi HTML5 `data-*` forniscono un modo conveniente per archiviare dati in elementi HTML. I dati memorizzati possono essere letti o modificati utilizzando JavaScript

```
<div data-submitted="yes" class="user_profile">
  ... some content ...
</div>
```

- La struttura degli attributi dei dati è `data-*`, ovvero il nome dell'attributo dati viene fornito dopo la parte `data-`. Usando questo nome, è possibile accedere all'attributo.
- I dati in formato stringa (incluso `json`) possono essere memorizzati utilizzando l'attributo `data-*`.

Supporto per i browser meno recenti

Gli attributi dei dati sono stati introdotti in HTML5, supportato da tutti i browser moderni, ma i browser meno recenti prima che HTML5 non riconosca gli attributi dei dati.

Tuttavia, nelle specifiche HTML, gli attributi che non sono riconosciuti dal browser devono essere lasciati da soli e il browser li ignorerà semplicemente durante il rendering della pagina.

Gli sviluppatori Web hanno utilizzato questo fatto per creare attributi non standard, che sono tutti gli attributi che non fanno parte delle specifiche HTML. Ad esempio, l'attributo `value` nella riga sottostante è considerato un attributo non standard poiché le specifiche per il tag `<img>` non hanno un attributo `value` e non è un attributo globale:

```

```

Ciò significa che, sebbene gli attributi dei dati non siano supportati nei browser più vecchi,

funzionano ancora ed è possibile impostarli e recuperarli utilizzando gli stessi metodi JavaScript generici `setAttribute` e `getAttribute` , ma non è possibile utilizzare la nuova proprietà del `dataset` che è supportata solo nei browser moderni.

Leggi Attributi dei dati online: <https://riptutorial.com/it/html/topic/1182/attributi-dei-dati>

Capitolo 5: Attributi evento HTML

Examples

Eventi modulo HTML

Eventi attivati da azioni all'interno di un modulo HTML (si applica a quasi tutti gli elementi HTML, ma è maggiormente utilizzato negli elementi del modulo):

Attributo	Descrizione
sfocato	Spara nel momento in cui l'elemento perde la messa a fuoco
onchange	Spara il momento in cui il valore dell'elemento viene modificato
oncontextmenu	Script da eseguire quando viene attivato un menu di scelta rapida
onfocus	Spara il momento in cui l'elemento diventa focalizzato
oninput	Script da eseguire quando un elemento riceve l'input dell'utente
oninvalid	Script da eseguire quando un elemento non è valido
onreset	Viene attivato quando viene fatto clic sul pulsante Ripristina in un modulo
onsearch	Si attiva quando l'utente scrive qualcosa in un campo di ricerca (per <code><input = "ricerca"></code>)
onselect	Incendi dopo che un testo è stato selezionato in un elemento
onSubmit	Viene generato quando viene inviato un modulo

Eventi di tastiera

Attributo	Descrizione
onKeyDown	Si attiva quando un utente preme un tasto
onkeypress	Si attiva quando un utente preme un tasto
onKeyUp	Si attiva quando un utente rilascia una chiave

Leggi Attributi evento HTML online: <https://riptutorial.com/it/html/topic/10924/attributi-evento-html>

Capitolo 6: Attributi globali

Parametri

Attributo	Descrizione
<code>class</code>	Definisce uno o più nomi di classi per un elemento. Vedi Classi e ID .
<code>contenteditable</code>	Imposta se il contenuto di un elemento può essere modificato.
<code>contextmenu</code>	Definisce un menu contestuale mostrato quando un utente fa clic con il pulsante destro del mouse su un elemento.
<code>dir</code>	Imposta la direzione del testo per il testo all'interno di un elemento.
<code>draggable</code>	Imposta se un elemento può essere trascinato.
<code>hidden</code>	Nasconde un elemento non attualmente in uso nella pagina.
<code>id</code>	Definisce un identificativo univoco per un elemento. Vedi Classi e ID .
<code>lang</code>	Definisce la lingua del contenuto di un elemento e i suoi valori di attributo del testo. Vedi le lingue del contenuto .
<code>spellcheck</code>	Imposta se eseguire il controllo ortografico / grammaticale sul contenuto di un elemento.
<code>style</code>	Definisce un insieme di stili CSS in linea per un elemento.
<code>tabindex</code>	Imposta l'ordine in cui gli elementi di una pagina vengono spostati dalla scorciatoia da tastiera della scheda.
<code>title</code>	Definisce ulteriori informazioni su un elemento, generalmente sotto forma di testo del tooltip al passaggio del mouse.
<code>translate</code>	Definisce se tradurre il contenuto di un elemento.

Osservazioni

Gli attributi globali vengono semplicemente attribuiti e possono essere applicati a qualsiasi elemento dell'intero documento.

Examples

Attributo Contenteditable

```
<p contenteditable>This is an editable paragraph.</p>
```

Facendo clic sul paragrafo, il contenuto di esso può essere modificato in modo simile a un campo di testo di input.

Quando l'attributo `contenteditable` non è impostato su un elemento, l'elemento lo eredita dal suo genitore. Quindi, tutto il testo figlio di un elemento modificabile contenuto sarà anche modificabile, ma è *possibile* disattivarla per il testo specifico, in questo modo:

```
<p contenteditable>
  This is an editable paragraph.
  <span contenteditable="false">But not this.</span>
</p>
```

Si noti che un elemento di testo non modificabile all'interno di un elemento modificabile avrà comunque un cursore di testo ereditato anche dal suo genitore.

Leggi Attributi globali online: <https://riptutorial.com/it/html/topic/2811/attributi-globali>

Capitolo 7: Barre di navigazione

Examples

Barra di navigazione di base

Le barre di navigazione sono essenzialmente un elenco di collegamenti, quindi gli elementi `ul` e `li` vengono utilizzati per racchiudere i collegamenti di navigazione.

```
<ul>
  <li><a href="#">Home</a></li>
  <li><a href="#">About</a></li>
  <li><a href="#">Contact</a></li>
</ul>
```

Barra di navigazione HTML5

Per creare una barra di navigazione utilizzando l'elemento `nav` HTML5, racchiudi i collegamenti all'interno del tag `nav`.

```
<nav>
  <a href="#">Home</a>
  <a href="#">About</a>
  <a href="#">Contact</a>
</nav>
```

Leggi Barre di navigazione online: <https://riptutorial.com/it/html/topic/10725/barre-di-navigazione>

Capitolo 8: Classi e ID

introduzione

Classi e ID rendono più semplice la referenziazione di elementi HTML da script e fogli di stile. L'attributo `class` può essere utilizzato su uno o più tag ed è usato dai CSS per lo styling. Tuttavia, gli ID si riferiscono a un singolo elemento, il che significa che lo stesso ID non dovrebbe mai essere usato due volte. Gli ID sono generalmente usati con JavaScript e link ai documenti interni e sono scoraggiati nei CSS. Questo argomento contiene spiegazioni ed esempi utili sull'uso corretto degli attributi di classe e ID in HTML.

Sintassi

- `class = "class1 class2 class3"`
- `id = "uniqueid"`

Parametri

Parametro	Dettagli
classe	Indica la Classe dell'elemento (non univoco)
id	Indica l'ID dell'elemento (unico nello stesso contesto)

Osservazioni

- Sia la `class` che l' `id` sono attributi globali e possono quindi essere assegnati a qualsiasi elemento HTML.
- I nomi delle classi devono iniziare con una lettera (AZ o az) e possono essere seguiti da lettere, cifre, trattini e underscore.
- In `HTML5`, gli attributi di classe e id possono essere utilizzati su qualsiasi elemento. In `HTML 4.0.1`, erano off-limits per i `<base>`, `<head>`, `<html>`, `<meta>`, `<param>`, `<script>`, `<style>` e `<title>`.
- Un elemento può avere una o più classi. Le classi sono separate da spazi e non possono contenere gli spazi stessi.
- Un elemento può avere un solo ID e deve essere unico nel suo contesto (cioè una pagina web). Anche gli ID non possono contenere gli spazi stessi.

Examples

Dando un elemento a una classe

Le classi sono identificatori per gli elementi a cui sono assegnati. Usa l'attributo `class` per

assegnare una classe a un elemento.

```
<div class="example-class"></div>
```

Per assegnare più classi a un elemento, separa i nomi di classe con spazi.

```
<div class="class1 class2"></div>
```

Usando le classi in CSS

Le classi possono essere utilizzate per lo styling di determinati elementi senza modificare tutti gli elementi di quel tipo. Ad esempio, questi due elementi `span` possono avere uno stile completamente diverso:

```
<span></span>
<span class="special"></span>
```

Le classi con lo stesso nome possono essere assegnate a qualsiasi numero di elementi in una pagina e riceveranno tutti lo stile associato a quella classe. Questo sarà sempre vero a meno che non si specifichi l'elemento all'interno del CSS.

Ad esempio, abbiamo due elementi, entrambi con l' `highlight` della classe:

```
<div class="highlight">Lorem ipsum</div>
<span class="highlight">Lorem ipsum</span>
```

Se il nostro CSS è come sotto, allora il colore verde verrà applicato al testo all'interno di entrambi gli elementi:

```
.highlight { color: green; }
```

Tuttavia, se vogliamo solo targetizzare `div` con la classe `highlight`, possiamo aggiungere specificità come di seguito:

```
div.highlight { color: green; }
```

Tuttavia, quando si `.highlight` styling con CSS, è generalmente consigliabile utilizzare solo classi (ad es. `.highlight`) piuttosto che elementi con classi (es. `div.highlight`).

Come con qualsiasi altro selettore, le classi possono essere annidate:

```
.main .highlight { color: red; } /* Descendant combinator */
.footer > .highlight { color: blue; } /* Child combinator */
```

Puoi anche concatenare il selettore di classe per selezionare solo elementi che hanno una combinazione di più classi. Ad esempio, se questo è il nostro HTML:

```
<div class="special left menu">This text will be pink</div>
```

E vogliamo colorare questo specifico pezzo di testo rosa, possiamo fare quanto segue nel nostro CSS:

```
.special.left.menu { color: pink; }
```

Dando un elemento un ID

L'attributo ID di un elemento è un identificatore che deve essere univoco nell'intero documento. Il suo scopo è quello di identificare univocamente l'elemento durante il collegamento (usando un'ancora), lo scripting o lo styling (con CSS).

```
<div id="example-id"></div>
```

Non dovresti avere due elementi con lo stesso ID nello stesso documento, anche se gli attributi sono collegati a due diversi tipi di elementi. Ad esempio, il seguente codice non è corretto:

```
<div id="example-id"></div>
<span id="example-id"></span>
```

I browser faranno del loro meglio per rendere questo codice, ma potrebbe verificarsi un comportamento imprevisto durante lo styling con CSS o l'aggiunta di funzionalità con JavaScript.

Per fare riferimento agli elementi in base al loro ID in CSS, anteporre l'ID a # .

```
#example-id { color: green; }
```

Per passare a un elemento con un ID su una determinata pagina, aggiungere # con il nome dell'elemento nell'URL.

```
http://example.com/about#example-id
```

Questa funzione è supportata nella maggior parte dei browser e non richiede JavaScript o CSS aggiuntivi per funzionare.

Problemi relativi agli ID duplicati

Avere più di un elemento con lo stesso ID è un problema difficile da risolvere. Il parser HTML di solito proverà a renderizzare la pagina in ogni caso. Di solito non si verifica alcun errore. Ma il ritmo potrebbe finire in una pagina web scorretta.

In questo esempio:

```
<div id="aDiv">a</div>
<div id="aDiv">b</div>
```

I selettori CSS funzionano ancora

```
#aDiv {  
    color: red;  
}
```

Ma JavaScript non riesce a gestire entrambi gli elementi:

```
var html = document.getElementById("aDiv").innerHTML;
```

In questo caso, la variabile `html` porta solo il primo contenuto `div ("a")`.

Valori accettabili

Per un ID

5

Le uniche restrizioni sul valore di un `id` sono:

1. deve essere unico nel documento
2. non deve contenere caratteri spaziali
3. deve contenere almeno un carattere

Quindi il valore può essere tutte cifre, solo una cifra, solo caratteri di punteggiatura, includere caratteri speciali, qualunque sia. Solo nessuno spazio bianco.

Quindi questi sono validi:

```
<div id="container"> ... </div>  
<div id="999"> ... </div>  
<div id="#%LV-||"> ... </div>  
<div id="____V"> ... </div>  
<div id="[] "> ... </div>  
<div id="♥"> ... </div>  
<div id="{ }"> ... </div>  
<div id="@ "> ... </div>  
<div id="☺☹☈☉☊☋☌☍☎☏☐☑☒☓☔☕☖☗☘☙☚☛☜☝☞☟☠☡☢☣☤☥☦☧☨☩☪☫☬☭☮☯☰☱☲☳☴☵☶☷☸☹☺☻☼☽☾☿☽☿☽☿"> ... </div>
```

Questo non è valido:

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

Anche questo non è valido, se incluso nello stesso documento:

```
<div id="results"> ... </div>  
<div id="results"> ... </div>
```

4.01

Un valore `id` deve iniziare con una lettera, che può essere seguita solo da:

- lettere (AZ / az)
- cifre (0-9)
- trattini ("-")
- caratteri di sottolineatura ("_")
- due punti (":")
- periodi (".")

Facendo riferimento al primo gruppo di esempi nella precedente sezione HTML5, solo uno è valido:

```
<div id="container"> ... </div>
```

Anche questi sono validi:

```
<div id="sampletext"> ... </div>
<div id="sample-text"> ... </div>
<div id="sample_text"> ... </div>
<div id="sample:text"> ... </div>
<div id="sample.text"> ... </div>
```

Di nuovo, se non inizia con una lettera (maiuscola o minuscola), non è valida.

Per una classe

Le regole per le classi sono essenzialmente le stesse di un `id`. La differenza è che `class` valori delle `class` *non* devono essere univoci nel documento.

Riferendosi agli esempi sopra, anche se questo non è valido nello stesso documento:

```
<div id="results"> ... </div>
<div id="results"> ... </div>
```

Questo è perfettamente a posto:

```
<div class="results"> ... </div>
<div class="results"> ... </div>
```

Nota importante: il modo in cui i valori ID e Classe vengono trattati al di fuori dell'HTML

Tieni presente che le regole e gli esempi sopra riportati si applicano nel contesto dell'HTML.

L'utilizzo di numeri, punteggiatura o caratteri speciali nel valore di un `id` o di una `class` può causare problemi in altri contesti, come CSS, JavaScript ed espressioni regolari.

Ad esempio, sebbene il seguente `id` sia valido in HTML5:

```
<div id="9lions"> ... </div>
```

... non è valido nei CSS:

4.1.3 Personaggi e caso

Nei CSS, gli *identificatori* (compresi i nomi degli elementi, le classi e gli ID nei selettori) possono contenere solo i caratteri [a-zA-Z0-9] e ISO 10646 caratteri U + 00A0 e superiori, più il trattino (-) e il carattere di sottolineatura (_); **non possono iniziare con una cifra, due trattini o un trattino seguito da una cifra** . (enfasi aggiunta)

Nella maggior parte dei casi potresti essere in grado di sfuggire ai personaggi in contesti in cui hanno restrizioni o significato speciale.

Riferimenti W3C

- [3.2.5.1 L'attributo `id`](#)
- [3.2.5.7 L'attributo della `class`](#)
- [6.2 Tipi di base SGML](#)

Leggi Classi e ID online: <https://riptutorial.com/it/html/topic/586/classi-e-id>

Capitolo 9: Collegamento di risorse

introduzione

Mentre molti script, icone e fogli di stile possono essere scritti direttamente nel codice HTML, è consigliabile e più efficiente includere queste risorse nel proprio file e collegarle al documento. Questo argomento riguarda il collegamento di risorse esterne come fogli di stile e script in un documento HTML.

Sintassi

- `<link rel="link-relation" type="mime-type" href="url">`
- `<script src="path-to-script"></script>`

Parametri

Attributo	Dettagli
charset	Specifica la codifica dei caratteri del documento collegato
crossorigin	Specifica come l'elemento gestisce le richieste di origine incrociata
href	Specifica la posizione del documento collegato
hreflang	Specifica la lingua del testo nel documento collegato
media	Specifica su quale dispositivo verrà visualizzato il documento collegato, spesso utilizzato con la selezione di fogli di stile in base al dispositivo in questione
rel	Richiesto Specifica la relazione tra il documento corrente e il documento collegato
rev	Specifica la relazione tra il documento collegato e il documento corrente
sizes	Specifica la dimensione della risorsa collegata. Solo quando <code>rel="icon"</code>
target	Specifica dove deve essere caricato il documento collegato
type	Specifica il tipo di supporto del documento collegato
integrity	Specifica un hash codificato base64 (sha256, sha384 o sha512) della risorsa collegata che consente al browser di verificarne la legittimità.

Examples

Foglio di stile CSS esterno

```
<link rel="stylesheet" href="path/to.css" type="text/css">
```

La pratica standard consiste nel posizionare i `<link>` CSS `<link>` all'interno del tag `<head>` nella parte superiore del codice HTML. In questo modo il CSS verrà caricato per primo e si applicherà alla tua pagina mentre viene caricata, piuttosto che mostrare un HTML non disegnato fino a quando il CSS non viene caricato. L'attributo `type` non è necessario in HTML5, poiché HTML5 di solito supporta i CSS.

```
<link rel="stylesheet" href="path/to.css" type="text/css">
```

e

```
<link rel="stylesheet" href="path/to.css">
```

... fai la stessa cosa in HTML5.

Un'altra pratica, anche se meno comune, consiste nell'utilizzare una dichiarazione `@import` all'interno del CSS diretto. Come questo:

```
<style type="text/css">
  @import ("path/to.css")
</style>

<style>
  @import ("path/to.css")
</style>
```

JavaScript

Sincrono

```
<script src="path/to.js"></script>
```

La pratica standard consiste nel posizionare tag `<script>` JavaScript appena prima del tag di chiusura `</body>`. Il caricamento degli script per ultimo consente di visualizzare le immagini del tuo sito più rapidamente e scoraggia il tuo JavaScript dal tentativo di interagire con elementi che non sono ancora stati caricati.

asincrono

```
<script src="path/to.js" async></script>
```

Un'altra alternativa, quando il codice Javascript che viene caricato non è necessario per

l'inizializzazione della pagina, può essere caricato in modo asincrono, accelerando il caricamento della pagina. Usando `async` il browser caricherà il contenuto dello script in parallelo e, una volta scaricato completamente, interromperà l'analisi HTML per analizzare il file Javascript.

differite

```
<script src="path/to.js" defer></script>
```

Gli script posticipati sono come gli script asincroni, con l'eccezione che l'analisi verrà eseguita solo dopo che l'HTML è stato interamente analizzato. Gli script posticipati sono garantiti per essere caricati nell'ordine della dichiarazione, allo stesso modo degli script sincroni.

<Noscript>

```
<noscript>JavaScript disabled</noscript>
```

L'elemento `<noscript>` definisce il contenuto da visualizzare se l'utente ha script disabilitati o se il browser non supporta l'uso di script. Il tag `<noscript>` può essere inserito in `<head>` o `<body>`.

favicon

```
<link rel="icon" type="image/png" href="/favicon.png">
<link rel="shortcut icon" type="image/x-icon" href="/favicon.ico">
```

Usa l' `image/png` tipo mime per i file PNG e l' `image/x-icon` per i file di icone (`*.ico`). Per la differenza, vedi [questa domanda SO](#).

Un file chiamato `favicon.ico` nella radice del tuo sito web verrà in genere caricato e applicato automaticamente, senza la necessità di un tag `<link>`. Se questo file cambia, i browser possono essere lenti e testardi sull'aggiornamento della cache.

CSS alternativo

```
<link rel="alternate stylesheet" href="path/to/style.css" title="yourTitle">
```

Alcuni browser consentono l'applicazione di fogli di stile alternativi se vengono offerti. Per impostazione predefinita, non verranno applicati, ma in genere possono essere modificati tramite le impostazioni del browser:

Firefox consente all'utente di selezionare il foglio di stile utilizzando il sottomenu **Visualizza> Stile pagina**, Internet Explorer supporta anche questa funzione (a partire da IE 8), accessibile anche da **Visualizza> Stile pagina** (almeno a partire da IE 11), ma Chrome richiede un'estensione per utilizzare la funzione (dalla versione 48). La pagina Web può anche fornire la propria interfaccia utente per consentire all'utente di

cambiare stile.

(Fonte: i [documenti MDN](#))

Feed Web

Utilizza l' `rel="alternate"` per consentire la rilevabilità dei tuoi feed Atom / RSS.

```
<link rel="alternate" type="application/atom+xml" href="http://example.com/feed.xml" />
<link rel="alternate" type="application/rss+xml" href="http://example.com/feed.xml" />
```

Vedi i documenti MDN per i [feed RSS](#) e [RSS Atomic](#) .

Collega l'attributo 'media'

```
<link rel="stylesheet" href="test.css" media="print">
```

Media specifica quale foglio di stile deve essere usato per quale tipo di media. L'uso del valore di `print` mostrerebbe solo quel foglio di stile per le pagine di stampa.

Il valore di questo attributo può essere uno qualsiasi dei [valori mediatype](#) (simile a una [query multimediale](#) CSS).

Prev e Next

Quando una pagina fa parte di una serie di articoli, per esempio, si può usare `prev` e `next` to point alle pagine che vengono prima e dopo.

```
<link rel="prev" href="http://stackoverflow.com/documentation/java/topics">
<link rel="next" href="http://stackoverflow.com/documentation/css/topics">
```

Suggerimento sulla risorsa: dns-prefetch, prefetch, prerender

preconnect

La relazione di `preconnect` è simile a `dns-prefetch` in quanto risolverà il DNS. Tuttavia, farà anche l'handshake TCP e la negoziazione TLS opzionale. Questa è una caratteristica sperimentale.

```
<link rel="preconnect" href="URL">
```

DNS-Prefetch

Informa i browser a risolvere il DNS per un URL, in modo che tutte le risorse da tale URL vengano caricate più velocemente.

```
<link rel="dns-prefetch" href="URL">
```

prefetch

Informa i browser che una data risorsa deve essere precaricata in modo che possa essere caricata più rapidamente.

```
<link rel="prefetch" href="URL">
```

DNS-Prefetch risolve solo il nome del dominio mentre il prefetch scarica / memorizza le risorse specificate.

Prerender

Informa i browser a recuperare e rendere l'URL in background, in modo che possano essere consegnati all'utente istantaneamente mentre l'utente naviga verso quell'URL. Questa è una caratteristica sperimentale.

```
<link rel="prerender" href="URL">
```

Leggi Collegamento di risorse online: <https://riptutorial.com/it/html/topic/712/collegamento-di-risorse>

Capitolo 10: Commenti

introduzione

Analogamente ad altri linguaggi di programmazione, markup e markdown, i commenti in HTML forniscono ad altri sviluppatori informazioni specifiche sullo sviluppo senza influire sull'interfaccia utente. A differenza di altri linguaggi, tuttavia, i commenti HTML possono essere utilizzati per specificare solo elementi HTML per Internet Explorer. Questo argomento spiega come scrivere commenti HTML e le loro applicazioni funzionali.

Sintassi

- `<!-- Comment text -->`

Osservazioni

Tutto ciò che inizia con `<!--` e termina con `-->` è un commento. I commenti non possono contenere due trattini adiacenti (`--`) e devono terminare esattamente con due trattini (es. `---->` non è corretto).

I commenti non sono visibili su una pagina Web e non possono essere abbinati a CSS. Possono essere utilizzati dallo sviluppatore della pagina per creare note all'interno dell'HTML o per nascondere determinati contenuti durante lo sviluppo.

Per le pagine dinamiche o interattive, nascondere e mostrare il contenuto è fatto con JavaScript e CSS piuttosto che con commenti HTML.

È possibile utilizzare JavaScript per ottenere il contenuto dei nodi di commento HTML e questi nodi possono essere creati dinamicamente, aggiunti e rimossi dal documento, ma ciò non influisce sul modo in cui viene visualizzata la pagina.

Poiché i commenti HTML fanno parte del codice sorgente della pagina, vengono scaricati nel browser insieme al resto della pagina. Generalmente il codice sorgente può essere visualizzato utilizzando l'opzione di menu del browser Web su "Visualizza sorgente" o "Visualizza origine pagina".

Examples

Creazione di commenti

I commenti HTML possono essere utilizzati per lasciare note a te stesso o ad altri sviluppatori su un punto specifico del codice. Possono essere iniziati con `<!--` e conclusi con `-->` , in questo modo:

```
<!-- I'm an HTML comment! -->
```

Possono essere incorporati in linea all'interno di altri contenuti:

```
<h1>This part will be displayed <!-- while this will not be displayed -->.</h1>
```

Possono anche estendersi su più righe per fornire ulteriori informazioni:

```
<!-- This is a multiline HTML comment.  
  Whatever is in here will not be rendered by the browser.  
  You can "comment out" entire sections of HTML code.  
-->
```

Tuttavia, non **possono** apparire all'interno di un altro tag HTML, come questo:

```
<h1 <!-- testAttribute="something" -->>This will not work</h1>
```

Questo produce HTML non valido come l'intero `<h1 <!-- testAttribute="something" -->` blocco sarebbe stato considerato un singolo tag iniziale `h1` con alcune altre informazioni non valide in esso contenute, seguito da un singolo `>` parentesi di chiusura che non fa nulla.

Per compatibilità con strumenti che tentano di analizzare HTML come XML o SGML, il corpo del tuo commento non dovrebbe contenere due trattini `--`.

Commenti condizionali per Internet Explorer

I commenti condizionali possono essere utilizzati per personalizzare il codice per diverse versioni di Microsoft Internet Explorer. Ad esempio, possono essere fornite diverse classi HTML, tag di script o fogli di stile. I commenti condizionali sono supportati nelle versioni di Internet Explorer dalla 5 alla 9. Le versioni precedenti di Internet Explorer e tutti i browser non IE sono considerati "di livello inferiore" e trattano i commenti condizionali come normali commenti HTML.

Di livello inferiore-nascosto

I commenti nascosti in basso funzionano incapsulando l'intero contenuto in quello che sembra essere un normale commento HTML. Solo IE da 5 a 9 continueranno a leggerlo come commento condizionale e nasconderanno o visualizzeranno il contenuto di conseguenza. Negli altri browser il contenuto verrà nascosto.

```
<!--[if IE]>  
  Revealed in IE 5 through 9. Commented out and hidden in all other browsers.  
<![endif]-->  
  
<!--[if lt IE 8]>  
  Revealed only in specified versions of IE 5-9 (here, IE less than 8).  
<![endif]-->  
  
<!--[if !IE]>  
  Revealed in no browsers. Equivalent to a regular HTML comment.  
<![endif]-->
```

```
<!--  
  For purposes of comparison, this is a regular HTML comment.  
-->
```

Di livello inferiore-rivelato

Questi sono leggermente diversi dai commenti nascosti di livello inferiore: solo il commento condizionale stesso è contenuto nella normale sintassi dei commenti. I browser che non supportano i commenti condizionali semplicemente li ignorano e visualizzano il resto del contenuto tra loro.

```
<!--[if IE]>-->  
  The HTML inside this comment is revealed in IE 5-9, and in all other browsers.  
<!--<![endif]-->  
  
<!--[if IE 9]>-->  
  This is revealed in specified versions of IE 5-9, and in all other browsers.  
<!--<![endif]-->  
  
<!--[if !IE]>-->  
  This is not revealed in IE 5-9. It's still revealed in other browsers.  
<!--<![endif]-->
```

Commentando spazi bianchi tra elementi in linea

Gli elementi di visualizzazione in linea, di solito come `span` o `a`, includeranno fino a un carattere di spazio bianco prima e dopo di essi nel documento. Per evitare righe molto lunghe nel markup (che è difficile da leggere) e lo spazio bianco non intenzionale (che influenza la formattazione), lo spazio bianco può essere commentato.

```
<!-- Use an HTML comment to nullify the newline character below: -->  
<a href="#">I hope there will be no extra whitespace after this!</a><!--  
--><button>Foo</button>
```

Provalo senza un commento tra gli elementi in linea e ci sarà uno spazio tra loro. A volte è necessario raccogliere il carattere dello spazio.

Codice di esempio:

```
<!-- Use an HTML comment to nullify the newline character below: -->  
<a href="#">I hope there will be no extra whitespace after this!</a><!--  
--><button>Foo</button>  
<hr>  
<!-- Without it, you can notice a small formatting difference: -->  
<a href="#">I hope there will be no extra whitespace after this!</a>  
<button>Foo</button>
```

Produzione:

I hope there will be no extra whitespace after this!

I hope there will be no extra whitespace after this!

Leggi Commenti online: <https://riptutorial.com/it/html/topic/468/commenti>

Capitolo 11: Controlli del menu di selezione

Sintassi

- `<select name=""></select>`
- `<datalist id=""></datalist>`
- `<optgroup label="Option Group"></optgroup>`
- `<option value="">Option</option>`

Examples

Seleziona Menu

L'elemento `<select>` genera un menu a discesa dal quale l'utente può scegliere un'opzione.

```
<select name="">
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
  <option value="4">Four</option>
</select>
```

Modifica della dimensione

Puoi cambiare la dimensione del menu di selezione con l'attributo `size`. Una dimensione di 0 o 1 visualizza il menu standard di menu a discesa. Una dimensione maggiore di 1 convertirà il menu a discesa in una casella che visualizza molte righe, con un'opzione per riga e una barra di scorrimento per scorrere le opzioni disponibili.

```
<select name="" size="4"></select>
```

Menu di selezione multi-opzione

Per impostazione predefinita, gli utenti possono selezionare solo un'opzione. L'aggiunta dell'attributo `multiple` consente agli utenti di selezionare più opzioni contemporaneamente e inviare tutte le opzioni selezionate con il modulo. L'utilizzo dell'attributo `multiple` converte automaticamente il menu a discesa in una casella come se avesse una dimensione definita. La dimensione predefinita quando ciò si verifica è determinata dal browser specifico che si sta utilizzando e non è possibile cambiarlo in un menu a discesa mentre consente selezioni multiple.

```
<select name="" multiple></select>
```

Quando si utilizza l'attributo `multiple`, esiste una differenza tra l'utilizzo di 0 e 1 per la dimensione, mentre non esiste alcuna differenza quando non si utilizza l'attributo. L'uso di 0 farà sì che il browser si comporti in qualsiasi modo predefinito è programmato da fare. L'utilizzo di 1 imposterà

esplicitamente la dimensione della casella risultante su una sola riga alta.

Gruppi di opzioni

Puoi raggruppare ordinatamente le opzioni all'interno di un menu di selezione per fornire un layout più strutturato in un lungo elenco di opzioni utilizzando l'elemento `<optgroup>` .

La sintassi è molto semplice, semplicemente usando l'elemento con un attributo `label` per identificare il titolo per il gruppo e contenente zero o più opzioni che dovrebbero essere all'interno di quel gruppo.

```
<select name="">
  <option value="milk">Milk</option>
  <optgroup label="Fruits">
    <option value="banana">Bananas</option>
    <option value="strawberry">Strawberries</option>
  </optgroup>
  <optgroup label="Vegetables" disabled>
    <option value="carrot">Carrots</option>
    <option value="zucchini">Zucchini</option>
  </optgroup>
</select>
```

Quando si utilizzano i gruppi di opzioni, non tutte le opzioni devono essere contenute in un gruppo. Inoltre, la disattivazione di un gruppo di opzioni disabilita tutte le opzioni all'interno del gruppo e non è possibile riattivare manualmente una singola opzione all'interno di un gruppo disabilitato.

Opzioni

Le opzioni all'interno di un menu di selezione sono ciò che l'utente selezionerà. La sintassi normale per un'opzione è la seguente:

```
<option>Some Option</option>
```

Tuttavia, è importante notare che il testo all'interno dell'elemento `<option>` stesso non viene sempre utilizzato e in sostanza diventa il valore predefinito per gli attributi che non sono specificati.

Gli attributi che controllano l'aspetto e la funzione effettivi dell'opzione sono `value` ed `label` . L'etichetta rappresenta il testo che verrà visualizzato nel menu a discesa (quello che stai guardando e farà clic per selezionarlo). Il valore rappresenta il testo che verrà inviato insieme all'invio del modulo. Se uno di questi valori viene omesso, utilizza il testo all'interno dell'elemento come valore. Quindi l'esempio che abbiamo dato sopra potrebbe essere "espanso" a questo:

```
<option label="Some Option" value="Some Option">
```

Notare l'omissione del testo interno e del tag finale, che non sono necessari per costruire effettivamente un'opzione all'interno del menu. Se fossero inclusi, il testo interno verrebbe ignorato

perché entrambi gli attributi sono già specificati e il testo non è necessario. Tuttavia, probabilmente non vedrai molte persone scriverle in questo modo. Il modo più comune in cui è scritto è con un valore che verrà inviato al server, insieme al testo interno che alla fine diventa l'attributo label, in questo modo:

```
<option value="option1">Some Option</option>
```

Selezione di un'opzione per impostazione predefinita

Puoi anche specificare una determinata opzione da selezionare nel menu per impostazione predefinita allegando ad essa l'attributo `selected`. Per impostazione predefinita, se non viene specificata alcuna opzione come selezionata nel menu, la prima opzione del menu verrà selezionata durante il rendering. Se più di una opzione ha l'attributo `selected` allegato, l'ultima opzione presente nel menu con l'attributo sarà quella selezionata per impostazione predefinita.

```
<option value="option1" selected>Some option</option>
```

Se stai utilizzando l'attributo in un menu di selezione multi-opzione, tutte le opzioni con l'attributo saranno selezionate per impostazione predefinita e nessuna sarà selezionata se nessuna opzione ha l'attributo.

```
<select multiple>
  <option value="option1" selected>Some option</option>
  <option value="option2" selected>Some option</option>
</select>
```

Datalist

Il tag `<datalist>` specifica un elenco di opzioni predefinite per un elemento `<input>`. Fornisce una funzionalità di "completamento automatico" sugli elementi `<input>`. Gli utenti vedranno un elenco a discesa di opzioni mentre scrivono.

```
<input list="Languages">

<datalist id="Languages">
  <option value="PHP">
  <option value="Perl">
  <option value="Python">
  <option value="Ruby">
  <option value="C+">
</datalist>
```

Supporto del browser

Cromo	Bordo	Mozilla	Safari	musica lirica
20.0	10.0	4.0	Non supportato	9.0

Leggi Controlli del menu di selezione online: <https://riptutorial.com/it/html/topic/722/controlli-del-menu-di-selezione>

Capitolo 12: doctype

introduzione

Doctypes - abbreviazione di "tipo di documento" - aiuta i browser a comprendere la versione di HTML in cui il documento è scritto per una migliore interpretabilità. Le dichiarazioni di doctype non sono tag HTML e appartengono alla parte superiore di un documento. Questo argomento spiega la struttura e la dichiarazione di vari doctype in HTML.

Sintassi

- `<!DOCTYPE [stringa specifica per la versione]>`

Osservazioni

La dichiarazione `<!DOCTYPE>` non è un tag HTML. È usato per specificare quale versione di HTML sta usando il documento. Questo è indicato come la dichiarazione del tipo di documento (DTD).

La dichiarazione `<!DOCTYPE>` NON è sensibile al maiuscolo / minuscolo. Per verificare se l'HTML delle tue pagine Web è valido, vai al [servizio di validazione del W3C](#).

- Alcune vecchie versioni di IE non supportano alcuni tag HTML a meno che non sia disponibile un doctype appropriato.
- È *fondamentale* dichiarare un doctype per assicurarsi che il browser non usi la modalità quirk. [Maggiori informazioni su MDN](#).

Examples

Aggiungere il Doctype

La dichiarazione `<!DOCTYPE>` dovrebbe sempre essere inclusa nella parte superiore del documento HTML, prima del tag `<html>`.

5

Vedi [Doctype HTML 5](#) per i dettagli sul Doctype HTML 5.

```
<!DOCTYPE html>
```

4.01

Vedi [Documento HTML 4.01](#) per dettagli su come questi tipi differiscono l'uno dall'altro.

Rigoroso

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

frameset

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

Documento HTML 4.01

La specifica HTML 4.01 fornisce diversi tipi di doctype che consentono di specificare diversi tipi di elementi all'interno del documento.

HTML 4.01 Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Include tutti gli elementi e gli attributi HTML, ma **non include gli elementi di presentazione o deprecati** e i **set di frame non sono consentiti** .

HTML 4.01 Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Include tutti gli elementi HTML e gli attributi e gli elementi di presentazione e deprecati, ma i **set di frame non sono consentiti** .

HTML 4.01 Frameset

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

Include tutti gli elementi e gli attributi HTML, gli elementi di presentazione e quelli deprecati. I frame sono permessi.

HTML 5 Doctype

HTML5 non è basato su SGML e pertanto non richiede un riferimento a un DTD.

Dichiarazione di Doctype HTML 5:

```
<!DOCTYPE html>
```

Case Insensitivity

Per le [specifiche HTML 5](#) di [DOCTYPE W3.org](#) :

Un DOCTYPE deve essere costituito dai seguenti componenti, in questo ordine:

1. Una stringa che corrisponde a una **distinzione tra maiuscole e minuscole ASCII** per la stringa "`<!DOCTYPE`".

pertanto anche i seguenti DOCTYPE sono validi:

```
<!doctype html>
<!dOCTyPe html>
<!DocTYpe html>
```

Questo articolo SO tratta ampiamente l'argomento: [maiuscole o minuscole doctype?](#)

Vecchi doctypes

HTML 3.2

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

HTML 3.2 è ben supportato dalla maggior parte dei browser in uso. Tuttavia, HTML 3.2 ha un supporto limitato per i fogli di stile e nessun supporto per le funzionalità di HTML 4 come i frame e l'internazionalizzazione.

HTML 2.0

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
```

HTML 2.0 è ampiamente supportato dai browser ma manca il supporto per tabelle, frame e internazionalizzazione, oltre a molti elementi e attributi di presentazione comunemente utilizzati.

Leggi doctype online: <https://riptutorial.com/it/html/topic/806/doctype>

Capitolo 13: Elementi di sezionamento

Osservazioni

Gli standard HTML5 non elencano l'elemento principale come elemento di sezione.

Examples

Articolo Element

L'elemento `<article>` contiene **contenuti autonomi** come articoli, post di blog, commenti degli utenti o un widget interattivo che potrebbe essere distribuito al di fuori del contesto della pagina, ad esempio tramite RSS.

- Quando gli elementi dell'articolo sono nidificati, il contenuto del nodo dell'articolo interno deve essere correlato all'elemento dell'articolo esterno.

Un blog (`section`) con post multipli (`article`) e commenti (`article`) potrebbe assomigliare a questo.

```
<section>
  <!-- Each individual blog post is an <article> -->
  <article>
    <header>
      <h1>Blog Post</h1>
      <time datetime="2016-03-13">13th March 2016</time>
    </header>

    <p>The article element represents a self contained article or document.</p>
    <p>The section element represents a grouping of content.</p>

    <section>
      <h2>Comments <small>relating to "Blog Post"</small></h2>

      <!-- Related comment is also a self-contained article -->
      <article id="user-comment-1">
        <p>Excellent!</p>
        <footer><p>...</p><time>...</time></footer>
      </article>
    </section>
  </article>

  <!-- ./repeat: <article> -->

</section>

<!-- Content unrelated to the blog or posts should be outside the section. -->
<footer>
  <p>This content should be unrelated to the blog.</p>
</footer>
```


Evitare l'uso non necessario

Quando il contenuto principale della pagina (escluse intestazioni, piè di pagina, barre di navigazione, ecc.) È semplicemente un gruppo di elementi. Puoi omettere `<article>` a favore dell'elemento `<main>`.

```
<article>
  <p>This doesn't make sense, this article has no real `context`.</p>
</article>
```

Invece, sostituisci l'articolo con un elemento `<main>` per indicare che questo è il contenuto `main` per questa pagina.

```
<main>
  <p>I'm the main content, I don't need to belong to an article.</p>
</main>
```

Se si utilizza un altro elemento, assicurarsi di specificare il ruolo `<main>` [ARIA](#) per interpretazione e rendering corretti su più dispositivi e browser non HTML5.

```
<section role="main">
  <p>This section is the main content of this page.</p>
</section>
```

Gli appunti:

- `<main>` discendenti [dell'elemento](#) `<main>` non sono consentiti all'interno di un `<article>`

[Fare clic qui per leggere la specifica HTML5 ufficiale per l'elemento](#) `<article>`

Elemento principale

L'elemento `<main>` contiene il **contenuto principale** per la tua pagina web. Questo contenuto è unico per la singola pagina e non dovrebbe apparire altrove sul sito. I contenuti ripetuti come intestazioni, piè di pagina, navigazione, loghi, ecc. Vengono posizionati all'esterno dell'elemento.

- L'elemento `<main>` dovrebbe essere usato sempre al massimo **una sola volta** su una singola pagina.
- L'elemento `<main>` non deve essere incluso come discendente di un `article`, `aside`, il `footer`, l'`header` o l'elemento di `nav`.

Nell'esempio seguente, stiamo visualizzando un **singolo post sul blog** (e informazioni correlate come riferimenti e commenti).

```
<body>
  <header>
    <nav>...</nav>
```

```
</header>

<main>
  <h1>Individual Blog Post</h1>
  <p>An introduction for the post.</p>

  <article>
    <h2>References</h2>
    <p>...</p>
  </article>

  <article>
    <h2>Comments</h2> ...
  </article>
</main>

<footer>...</footer>
</body>
```

- Il post del blog è contenuto all'interno dell'elemento `<main>` per indicare che questo è il contenuto principale di questa pagina (e quindi unico nel suo genere sul sito Web).
- I `<header>` e `<footer>` sono *fratelli* dell'elemento `<main>` .

Gli appunti:

La specifica HTML5 riconosce l'elemento `<main>` come un elemento di **raggruppamento** e non un elemento di *sezione* .

- **Attributi del ruolo ARIA** : `main` (*predefinito*) , `presentation`

Aggiungere un `role="main"` attributo del **ruolo ARIA** ad **altri elementi** destinati ad essere utilizzati come contenuto principale è consigliato per aiutare gli **interpreti** che non supportano HTML5 e anche per fornire più contesto per quelli che lo fanno.

L'elemento `<main>` di default ha il ruolo principale e quindi non ha bisogno di essere fornito.

[Clicca qui per leggere la specifica ufficiale HTML5 per l'elemento `<main>`](#)

Elemento Nav

L'elemento `<nav>` è destinato principalmente a essere utilizzato per le sezioni che contengono i **blocchi di navigazione principali** per il sito Web, che possono includere collegamenti ad altre parti della pagina Web (*ad esempio, ancore per un indice*) o altre pagine interamente.

Elementi in linea

Quanto segue visualizzerà un insieme in linea di collegamenti ipertestuali.

```
<nav>
  <a href="https://google.com">Google</a>
  <a href="https://www.yahoo.com">Yahoo!</a>
  <a href="https://www.bing.com">Bing</a>
</nav>
```

Utilizzare gli elementi dell'elenco quando necessario

Se il contenuto rappresenta un elenco di elementi, utilizzare un elemento dell'elenco per mostrarlo e migliorare l'esperienza dell'utente.

Nota il `role="navigation"` , *più su questo sotto*.

```
<nav role="navigation">
  <ul>
    <li><a href="https://google.com">Google</a></li>
    <li><a href="https://www.yahoo.com">Yahoo!</a></li>
    <li><a href="https://www.bing.com">Bing</a></li>
  </ul>
</nav>
```

Evitare l'uso non necessario

`<footer>` elementi `<footer>` possono avere un elenco di collegamenti ad altre parti del sito (FAQ, T & C, ecc.). In questo caso è sufficiente l'elemento `footer`, non è *necessario* avvolgere ulteriormente i collegamenti con un elemento `<nav>` in `<footer>` .

```
<!-- the <nav> is not required in the <footer> -->
<footer>
  <nav>
    <a href="#">...</a>
  </nav>
</footer>

<!-- The footer alone is sufficient -->
<footer>
  <a href="#">...</a>
</footer>
```

Gli appunti:

- `<main>` discendenti [dell'elemento](#) `<main>` non sono consentiti all'interno di `<nav>`

Aggiunta di un `role="navigation"` [ruolo ARIA](#) all'elemento `<nav>` è consigliato per aiutare i programmi utente che non supportano HTML5 e anche per fornire più contesto per quelli che lo fanno.

```
<nav role="navigation"><!-- ... --></nav>
```

Screen Reader: *(software che consente agli utenti non vedenti o ipovedenti di navigare nel sito)*

I programmi utente come gli screen reader interpretano l'elemento `<nav>` diverso a seconda delle loro esigenze.

- Potrebbe dare all'elemento `<nav>` una priorità più alta durante il rendering della pagina
- Potrebbe ritardare il rendering dell'elemento
- Potrebbe adattare la pagina in un modo specifico per adattarsi alle esigenze dell'utente

esempio: rendere i collegamenti testuali all'interno degli elementi `<nav>` più grandi per chi ha problemi di vista.

[Fai clic qui per leggere la specifica HTML5 ufficiale per l'elemento `<nav>`](#)

Elemento di sezione

L'elemento `<section>` rappresenta una sezione generica per raggruppare il contenuto tematicamente. Ogni sezione, in genere, dovrebbe essere in grado di essere identificata con un elemento di intestazione come figlio della `section`.

- Puoi usare l'elemento `<section>` in un `<article>` e viceversa.
- Ogni sezione dovrebbe avere un *tema* (un elemento di intestazione che identifica questa regione)
- Non utilizzare l'elemento `<section>` come "contenitore" di stile generale.
Se hai bisogno di un contenitore per applicare lo styling, usa invece un `<div>`.

Nell'esempio seguente, stiamo visualizzando un **singolo post sul blog** con più capitoli ogni capitolo è una sezione (*una serie di contenuti raggruppati tematicamente, che possono essere identificati dagli elementi di intestazione in ciascuna sezione*).

```
<article>
  <header>
    <h2>Blog Post</h2>
  </header>
  <p>An introduction for the post.</p>
  <section>
    <h3>Chapter 1</h3>
    <p>...</p>
  </section>
  <section>
    <h3>Chapter 2</h3>
    <p>...</p>
  </section>
  <section>
    <h3>Comments</h3> ...
  </section>
</article>
```

Gli appunti:

Gli sviluppatori dovrebbero usare l'elemento **dell'articolo** quando ha senso per sindacare il contenuto dell'elemento.

[Clicca qui per leggere la specifica ufficiale HTML5 per l'elemento `<main>`](#)

Elemento di intestazione

L'elemento `<header>` rappresenta il contenuto introduttivo per il contenuto della sezione antenato più prossimo o per l'elemento radice di sezionamento. Un `<header>` genere contiene un gruppo di aiuti introduttivi o di navigazione.

Nota: l'elemento dell'intestazione non è un contenuto di sezione; non introduce una nuova sezione.

Esempi:

```
<header>
  <p>Welcome to...</p>
  <h1>Voidwars!</h1>
</header>
```

In questo esempio, `<article>` ha un `<header>` .

```
<article>
  <header>
    <h1>Flexbox: The definitive guide</h1>
  </header>
  <p>The guide about Flexbox was supposed to be here, but it turned out Wes wasn't a Flexbox expert either.</p>
</article>
```

Raccomandazione proposta dal W3C

Elemento di piè di pagina

L'elemento `<footer>` contiene la parte footer della pagina.

Ecco un esempio per l'elemento `<footer>` che contiene il tag di paragrafo `p` .

```
<footer>
  <p>All rights reserved</p>
</footer>
```

Leggi Elementi di sezionamento online: <https://riptutorial.com/it/html/topic/311/elementi-di-sezionamento>

Capitolo 14: Elementi multimediali

Parametri

Attributo	Dettagli
larghezza	Imposta la larghezza dell'elemento in pixel.
altezza	Imposta l'altezza dell'elemento in pixel.
<source>	Definisce le risorse dei file audio o video
traccia	Definisce la traccia del testo per gli elementi multimediali
controlli	Visualizza i controlli
riproduzione automatica	Avvia automaticamente la riproduzione del file multimediale
ciclo continuo	Riproduce i media in un ciclo ripetuto
smorzato	Riproduce i media senza audio
manifesto	Assegna un'immagine da visualizzare fino a quando non viene caricato un video

Osservazioni

Supporto nei browser

caratteristica	Cromo	Firefox (Gecko)	Internet Explorer	musica lirica	Safari
Supporto di base	3.0	3.5 (1.9.1)	9.0	10.50	3.1
<audio> : PCM in WAVE	(Sì)	3.5 (1.9.1)	Nessun supporto	10.50	3.1
<audio> : Vorbis in WebM	(Sì)	4.0 (2.0)	Nessun supporto	10.60	3.1
<audio> : Streaming Vorbis / Opus in WebM tramite MSE	?	36.0	?	?	?
<audio> : Vorbis in Ogg	(Sì)	3.5 (1.9.1)	Nessun	10.50	Nessun

caratteristica	Cromo	Firefox (Gecko)	Internet Explorer	musica lirica	Safari
			supporto		supporto
<audio> : MP3	(Sì)	(Sì)	9.0	(Sì)	3.1
<audio> : MP3 in MP4	?	?	?	?	(Sì)
<audio> : AAC in MP4	(Sì)	(Sì)	9.0	(Sì)	3.1
<audio> : Opus in Ogg	27,0	15.0 (15.0)	?	?	?
<video> : VP8 e Vorbis in WebM	6.0	4.0 (2.0)	9.0	10.60	3.1
<video> : VP9 e Opus in WebM	29,0	28.0 (28.0)	?	(Sì)	?
<video> : Streaming WebM tramite MSE	?	42.0 (42.0)	?	?	?
<video> : Theora e Vorbis in Ogg	(Sì)	3.5 (1.9.1)	Nessun supporto	10.50	Nessun supporto
<video> : H.264 e MP3 in MP4	(Sì)	(Sì)	9.0	(Sì)	(Sì)
<video> : H.264 e AAC in MP4	(Sì)	(Sì)	9.0	(Sì)	3.1
qualsiasi altro formato	Nessun supporto	Nessun supporto	Nessun supporto	Nessun supporto	3.1

Examples

Usando l'elemento per visualizzare contenuti audio / video

Usa l'elemento HTML `<audio>` per incorporare il contenuto video / audio in un documento. L'elemento video / audio contiene una o più sorgenti video / audio. Per specificare una fonte, utilizzare l'attributo `src` o l'elemento `<source>` ; il browser sceglierà quello più adatto.

Esempio di tag audio:

```
<!-- Simple video example -->
<video src="videofile.webm" autoplay poster="posterimage.jpg">
  Sorry, your browser doesn't support embedded videos,
  but don't worry, you can <a href="videofile.webm">download it</a>
  and watch it with your favorite video player!
</video>
```

```

<!-- Video with subtitles -->
<video src="foo.webm">
  <track kind="subtitles" src="foo.en.vtt" srclang="en" label="English">
  <track kind="subtitles" src="foo.sv.vtt" srclang="sv" label="Svenska">
</video>
<!-- Simple video example -->
<video width="480" controls poster="https://archive.org/download/WebmVp8Vorbis/webmvp8.gif" >
  <source src="https://archive.org/download/WebmVp8Vorbis/webmvp8.webm" type="video/webm">
  <source src="https://archive.org/download/WebmVp8Vorbis/webmvp8_512kb.mp4" type="video/mp4">
  <source src="https://archive.org/download/WebmVp8Vorbis/webmvp8.ogv" type="video/ogg">
  Your browser doesn't support HTML5 video tag.
</video>

```

Esempio di tag audio:

```

<!-- Simple audio playback -->
<audio src="http://developer.mozilla.org/@api/deki/files/2926/=AudioTest_(1).ogg" autoplay>
  Your browser does not support the <code>audio</code> element.
</audio>

<!-- Audio playback with captions -->
<audio src="foo.ogg">
  <track kind="captions" src="foo.en.vtt" srclang="en" label="English">
  <track kind="captions" src="foo.sv.vtt" srclang="sv" label="Svenska">
</audio>

```

Audio

HTML5 fornisce un nuovo standard per l'incorporamento di un file audio su una pagina web.

Puoi incorporare un file audio in una pagina utilizzando l'elemento `<audio>` :

```

<audio controls>
  <source src="file.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>

```

video

Puoi incorporare anche un video in una pagina web utilizzando l'elemento `<video>` :

```

<video width="500" height="700" controls>
  <source src="video.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>

```

Intestazione o sfondo del video

Aggiunta di un video che riproduce automaticamente su un loop e non ha controlli o suoni. Perfetto per un'intestazione o uno sfondo video.

```

<video width="1280" height="720" autoplay muted loop poster="video.jpg" id="videobg">

```



```
<source src="video.mp4" type="video/mp4">
<source src="video.webm" type="video/webm">
<source src="video.ogv" type="video/ogg">
</video>
```

Questo CSS fornisce un fallback se il video non può essere caricato. Nota che è consigliato utilizzare il primo frame del video come poster video.jpg.

```
#videobg {
  background: url(video.jpg) no-repeat;
  background-size: cover;
}
```

Leggi Elementi multimediali online: <https://riptutorial.com/it/html/topic/2111/elementi-multimediali>

Capitolo 15: Elementi vuoti

introduzione

Non tutti i tag HTML hanno la stessa struttura. Mentre la maggior parte degli elementi richiede un tag di apertura, un tag di chiusura e un contenuto, alcuni elementi, noti come elementi void, richiedono solo un tag di apertura poiché essi stessi non contengono alcun elemento. Questo argomento spiega e dimostra l'uso corretto degli elementi void in HTML

Osservazioni

Un elemento void non può avere alcun contenuto ma può avere attributi. Gli elementi vuoti si chiudono automaticamente, quindi non devono avere un tag di chiusura.

In [HTML5](#) , i seguenti elementi sono nulli:

- `area`
- `base`
- `br`
- `col`
- `embed`
- `hr`
- `img`
- `input`
- `keygen`
- `link`
- `meta`
- `param`
- `source`
- `track`
- `wbr`

Examples

Elementi vuoti

HTML 4.01 / XHTML 1.0 Strict include i seguenti elementi di annullamento:

- `area` - `area` cliccabile, definita in un'immagine
- `base` : specifica un URL di base da cui tutti i collegamenti di base
- `br` - line break
- `col` - colonna in una tabella [deprecato]
- `hr` - horizontal rule (line)
- `img` - immagine
- `input` - campo in cui gli utenti inseriscono i dati
- `link` : collega una risorsa esterna al documento
- `meta` : fornisce informazioni sul documento

- `param` - definisce i parametri per i plugin

Gli standard HTML 5 includono tutti i tag non deprecati dall'elenco precedente e

- `command` - rappresenta un comando che gli utenti possono invocare [obsoleto]
- `keygen` - facilita la generazione di chiavi pubbliche per i certificati Web [deprecato]
- `source` : specifica le origini multimediali per elementi di `picture` , `audio` e `video`

L'esempio seguente **non** include elementi void:

```
<div>
  <a href="http://stackoverflow.com/">
    <h3>Click here to visit <i>Stack Overflow!</i></h3>
  </a>
  <button onclick="alert('Hello!');">Say Hello!</button>
  <p>My favorite language is <b>HTML</b>. Here are my others:</p>
  <ol>
    <li>CSS</li>
    <li>JavaScript</li>
    <li>PHP</li>
  </ol>
</div>
```

Nota come ogni elemento ha un tag di apertura, un tag di chiusura e testo o altri elementi all'interno dei tag di apertura e chiusura. I tag vuoti, tuttavia, sono mostrati nell'esempio seguente:

```

<br>
<hr>
<input type="number" placeholder="Enter your favorite number">
```

Ad eccezione del tag `img`, tutti questi elementi void hanno solo un tag di apertura. Il tag `img`, a differenza di qualsiasi altro tag, ha una chiusura automatica / prima del segno maggiore del tag di apertura. È consigliabile disporre di uno spazio prima della barra.

Leggi Elementi vuoti online: <https://riptutorial.com/it/html/topic/1449/elementi-vuoti>

Capitolo 16: Elemento di progresso

Parametri

Parametro	Valore
max	Quanto lavoro richiede l'attività in totale
valore	Quanto del lavoro è già stato fatto
posizione	Questo attributo restituisce la posizione corrente dell'elemento <code><progress></code>
etichette	Questo attributo restituisce un elenco di etichette di elementi <code><progress></code> (se presenti)

Osservazioni

L'elemento `<progress>` non è supportato nelle versioni di Internet Explorer meno di 10

L'elemento `<progress>` è l'elemento sbagliato da utilizzare per qualcosa che è solo un indicatore, piuttosto che il progresso del compito. Ad esempio, mostrare l'utilizzo dello spazio su disco utilizzando l'elemento `<progress>` è inappropriato. Invece, l'elemento `<meter>` è disponibile per questo tipo di casi d'uso.

Examples

Progresso

L'elemento `<progress>` è nuovo in HTML5 e viene utilizzato per rappresentare lo stato di avanzamento di un'attività

```
<progress value="22" max="100"></progress>
```

Questo crea una barra piena del 22%

Modifica del colore di una barra di avanzamento

Le barre di avanzamento possono essere abbinate al selettore di `progress[value]`.

Questo esempio fornisce una barra di avanzamento di una larghezza di 250px e un'altezza di 20px

```
progress[value] {  
  width: 250px;  
  height: 20px;  
}
```

```
}
```

Le barre di avanzamento possono essere particolarmente difficili da stile.

Chrome / Safari / Opera

Questi browser usano il selettore `-webkit-appearance` per `-webkit-appearance` il tag `progress`. Per sovrascriverlo, possiamo resettare l'aspetto.

```
progress[value] {  
  -webkit-appearance: none;  
  appearance: none;  
}
```

Ora possiamo modellare il contenitore stesso

```
progress[value]::-webkit-progress-bar {  
  background-color: "green";  
}
```

Firefox

Firefox disegna la barra di avanzamento in modo leggermente diverso. Dobbiamo usare questi stili

```
progress[value] {  
  -moz-appearance: none;  
  appearance: none;  
  border: none; /* Firefox also renders a border */  
}
```

Internet Explorer

Internet Explorer 10+ supporta l'elemento di `progress`. Tuttavia, non supporta la proprietà `background-color`. Dovrai invece usare la proprietà `color`.

```
progress[value] {  
  -webkit-appearance: none;  
  -moz-appearance: none;  
  appearance: none;  
  
  border: none; /* Remove border from Firefox */  
  
  width: 250px;  
  height: 20px;  
  
  color: blue;  
}
```

Fallback HTML

Per i browser che non supportano l'elemento di `progress` , è possibile utilizzarlo come soluzione alternativa.

```
<progress max="100" value="20">
  <div class="progress-bar">
    <span style="width: 20%; ">Progress: 20%</span>
  </div>
</progress>
```

I browser che supportano il tag di avanzamento ignoreranno il div nested all'interno. I browser legacy che non sono in grado di identificare il tag `progress` eseguiranno il rendering del div.

Leggi Elemento di progresso online: <https://riptutorial.com/it/html/topic/5055/elemento-di-progresso>

Capitolo 17: Elemento di uscita

Parametri

Attributo	Descrizione
Globale	Attributi disponibili per qualsiasi elemento HTML5. Per una documentazione completa di questi attributi vedere: Attributi globali MDN
nome	Una stringa che rappresenta il nome di un output. Come elemento del modulo, l'output può essere referenziato dal suo nome usando la proprietà <code>document.forms</code> . Questo attributo viene utilizzato anche per la raccolta di valori su un modulo di invio.
per	Un elenco separato dagli spazi degli id degli elementi del modulo (ad es. <code><inputs id="inp1"> for value is "inp1")</code> per il quale l'output è destinato a visualizzare i calcoli.
modulo	Una stringa che rappresenta il <code><form></code> che è associato all'output. Se l'output è effettivamente al di fuori di <code><form></code> , questo attributo garantirà che l'output appartiene ancora a <code><form></code> e soggetto alle raccolte e ai sottomessi di <code><form></code> .

Examples

Elemento di output che utilizza gli attributi For e Form

La seguente demo presenta un uso dell'elemento `<output>` degli attributi `[for]` e `[form]`. Tieni presente che `<output>` **bisogna di JavaScript** per funzionare. JavaScript inline è comunemente usato nei moduli come dimostra questo esempio. Sebbene gli elementi `<input>` siano `type="number"`, i loro `value` s non sono numeri, sono testo. Pertanto, se è necessario calcolare il `value` s, è necessario convertire ciascun `value` in un numero utilizzando metodi quali: `parseInt()`, `parseFloat()`, `Number()`, ecc.

Dimostrazione dal vivo

```
<!--form1 will collect the values of in1 and in2 on 'input' event.-->
<!--out1 value will be the sum of in1 and in2 values.-->

<form id="form1" name="form1" oninput="out1.value = parseInt(in1.value, 10) +
parseInt(in2.value, 10)">

  <fieldset>

    <legend>Output Example</legend>

    <input type="number" id="in1" name="in1" value="0">
  <br/>
```

```
+
<input type="number" id="in2" name="in2" value="0">

</fieldset>

</form>

<!--[for] attribute enables out1 to display calculations for in1 and in2.-->
<!--[form] attribute designates form1 as the form owner of out1 even if it isn't a
descendant.-->

<output name="out1" for="in1 in2" form="form1">0</output>
```

Elemento di output con attributi

```
<output name="out1" form="form1" for="inp1 inp2"></output>
```

Leggi Elemento di uscita online: <https://riptutorial.com/it/html/topic/723/elemento-di-uscita>

Capitolo 18: Elemento div

introduzione

L'elemento div in HTML è un elemento contenitore che incapsula altri elementi e può essere utilizzato per raggruppare e separare parti di una pagina Web. Un div da solo non rappresenta in sé nulla ma è un potente strumento nel web design. Questo argomento riguarda lo scopo e le applicazioni dell'elemento div.

Sintassi

- `<div>example div</div>`

Examples

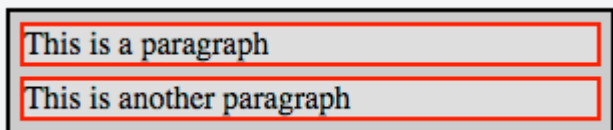
annidamento

È pratica comune posizionare più `<div>` all'interno di un altro `<div>`. Questo è solitamente definito come elementi di "nidificazione" e consente di suddividere ulteriormente gli elementi in sottosezioni o di aiutare gli sviluppatori con lo stile CSS.

`<div class="outer-div">` è usato per raggruppare due elementi `<div class="inner-div">`; ognuno contenente un elemento `<p>`.

```
<div class="outer-div">
  <div class="inner-div">
    <p>This is a paragraph</p>
  </div>
  <div class="inner-div">
    <p>This is another paragraph</p>
  </div>
</div>
```

Ciò produrrà il seguente risultato (stili CSS applicati per chiarezza):



Nidificazione di elementi inline e di blocco Mentre si annidano gli elementi, è necessario tenere presente che esistono elementi in linea e di blocco. mentre gli elementi di blocco "aggiungono un'interruzione di riga sullo sfondo", ciò significa che gli altri elementi nidificati vengono mostrati automaticamente nella riga successiva, gli elementi in linea possono essere posizionati uno accanto all'altro per impostazione predefinita

Evita nidificazione profonda `<div>`

Un layout di contenitori nidificati profondi e spesso usati mostra uno stile di codifica errato.

Gli angoli arrotondati o alcune funzioni simili spesso creano un tale codice HTML. Per la maggior parte dei browser di ultima generazione ci sono le controparti CSS3. Cerca di utilizzare il minor numero possibile di elementi HTML per aumentare il contenuto per taggare il rapporto e ridurre il carico della pagina, ottenendo un posizionamento migliore nei motori di ricerca.

sezione `div` elemento non deve essere nidificato più profondo di 6 livelli.

Utilizzo di base

L'elemento `<div>` solito non ha un significato semantico specifico di per sé, rappresenta semplicemente una divisione ed è tipicamente usato per raggruppare e incapsulare altri elementi all'interno di un documento HTML e separarli da altri gruppi di contenuti. In quanto tale, ogni `<div>` è meglio descritto dal suo contenuto.

```
<div>
  <p>Hello! This is a paragraph.</p>
</div>
```

L'elemento `div` è in genere un [elemento a livello di blocco](#), ovvero separa un blocco di un documento HTML e occupa la larghezza massima della pagina. I browser in genere hanno la seguente regola CSS predefinita:

```
div {
  display: block;
}
```

È fortemente incoraggiato dal **World Wide Web Consortium (W3C)** per visualizzare l'elemento `div` come elemento di ultima istanza, per quando nessun altro elemento è adatto. L'uso di elementi più appropriati anziché l'elemento `div` consente una migliore accessibilità per i lettori e una più semplice manutenibilità per gli autori.

Ad esempio, un post sul blog verrebbe contrassegnato usando `<article>`, un capitolo che usa `<section>`, gli aiuti alla navigazione di una pagina usando `<nav>` e un gruppo di controlli di form usando `<fieldset>`.

Gli elementi `div` possono essere utili per scopi stilistici o per racchiudere più paragrafi all'interno di una sezione che devono essere annotati tutti in un modo simile.

Leggi Elemento div online: <https://riptutorial.com/it/html/topic/1468/elemento-div>

Capitolo 19: Elemento etichetta

Sintassi

- `<label>Example <input type="radio" name="r"></label>` // Disposizione di un elemento di controllo
- `<label for="rad1">Example</label> <input id="rad1" type="radio" name="r">` // Utilizzo for attributo

Parametri

attributi	Descrizione
per	Riferimento all'elemento ID di destinazione. Vale a dire: <code>for="surname"</code>
modulo	HTML5 , [Obsoleto] Riferimento al modulo contenente l'elemento target. Gli elementi dell'etichetta sono previsti all'interno di un elemento <code><form></code> . Se viene fornito il <code>form="someFormId"</code> ciò consente di posizionare l'etichetta ovunque nel documento.

Examples

Uso di base

Forma semplice con etichette ...

```
<form action="/login" method="POST">

  <label for="username">Username:</label>
  <input id="username" type="text" name="username" />

  <label for="pass">Password:</label>
  <input id="pass" type="password" name="pass" />

  <input type="submit" name="submit" />

</form>
```

5

```
<form id="my-form" action="/login" method="POST">

  <input id="username" type="text" name="username" />

  <label for="pass">Password:</label>
  <input id="pass" type="password" name="pass" />

  <input type="submit" name="submit" />
```

```
</form>

<label for="username" form="my-form">Username:</label>
```

A proposito di etichetta

L'elemento `<label>` viene utilizzato per fare riferimento a un elemento di azione modulo. Nell'ambito **dell'interfaccia utente** è usato per facilitare il target / selezione di elementi come `Type` `radio` o `checkbox`.

`<label>` come wrapper

Può racchiudere l'elemento d'azione desiderato

```
<label>
  <input type="checkbox" name="Cats">
  I like Cats!
</label>
```

(Cliccando sul testo l' `input` destinazione cambierà stato / valore)

`<label>` come riferimento

Usando l'attributo `for` non è necessario posizionare l'elemento di controllo come discendente `label`, ma il valore `for` deve corrispondere al suo ID

```
<input id="cats" type="checkbox" name="Cats">
<label for="cats" >I like Cats!</label>
```

Nota

Non utilizzare più di un elemento di controllo all'interno di un elemento `<label>`

Leggi Elemento etichetta online: <https://riptutorial.com/it/html/topic/1704/elemento-etichetta>

Capitolo 20: elenchi

introduzione

L'HTML offre tre modi per specificare gli elenchi: elenchi ordinati, elenchi non ordinati e elenchi di descrizioni. Gli elenchi ordinati utilizzano sequenze ordinali per indicare l'ordine degli elementi della lista, gli elenchi non ordinati utilizzano un simbolo definito come un punto elenco per elencare gli elementi in ordine non ordinato e gli elenchi di descrizioni utilizzano i rientri per elencare gli elementi con i relativi figli. Questo argomento spiega l'implementazione e la combinazione di questi elenchi nel markup HTML.

Sintassi

- `<ol> ordered list items </ol>`
- `<ul> unordered list items </ul>`
- `<li> list item (ordered and not) </li>`
- `<dl> description list items </dl>`
- `<dt> description list title </dt>`
- `<dd> description list description </dd>`

Osservazioni

Guarda anche

È possibile aggiungere una proprietà CSS tipo `list-style` a un tag `<ul>` per modificare il tipo di icona utilizzata per contrassegnare ogni elemento dell'elenco, ad esempio `<ul style="list-style-type:disc">`. Sono consentiti i seguenti tipi di stile elenco:

- `"list-style-type: disc"` è il punto predefinito.
- `"list-style-type: circle"` è un cerchio vuoto.
- `"list-style-type: square"` è un quadrato pieno.
- `"list-style-type: none"` non utilizza alcun segno.

È inoltre possibile aggiungere un attributo `type` a un tag `<ol>` per modificare la modalità di esecuzione della numerazione, ad esempio `<ol type="1">`. Sono consentiti i seguenti tipi:

- `type = "1"` è il modulo predefinito.
- `type = "A"` utilizza le lettere maiuscole in ordine alfabetico
- `type = "a"` usa lettere minuscole in ordine alfabetico
- `type = "I"` utilizza numeri romani con caratteri maiuscoli
- `type = "i"` usa numeri romani con caratteri minuscoli

Examples

Lista non ordinata

Un elenco non ordinato può essere creato con il tag `<ul>` e ogni elemento di elenco può essere creato con il tag `<li>` come mostrato nell'esempio seguente:

```
<ul>
  <li>Item</li>
  <li>Another Item</li>
  <li>Yet Another Item</li>
</ul>
```

Questo produrrà un elenco puntato (che è lo stile predefinito):

- Articolo
- Un altro oggetto
- Ancora un altro oggetto

Dovresti usare `ul` per visualizzare un elenco di elementi, in cui l'ordine degli articoli non è importante. Se la modifica dell'ordine degli articoli rende l'elenco errato, è necessario utilizzare `<ol>`.

Lista ordinata

Un elenco ordinato può essere creato con il tag `<ol>` e ogni elemento di elenco può essere creato con il tag `<li>` come nell'esempio seguente:

```
<ol>
  <li>Item</li>
  <li>Another Item</li>
  <li>Yet Another Item</li>
</ol>
```

Questo produrrà un elenco numerato (che è lo stile predefinito):

1. Articolo
2. Un altro oggetto
3. Ancora un altro oggetto

Modifica manuale dei numeri

Ci sono un paio di modi in cui puoi giocare con quali numeri appaiono sugli elementi di lista in un elenco ordinato. Il primo modo è impostare un numero iniziale, usando l'attributo `start`. L'elenco inizierà da questo numero definito e continuerà ad aumentare di uno come al solito.

```
<ol start="3">
  <li>Item</li>
  <li>Some Other Item</li>
  <li>Yet Another Item</li>
</ol>
```

Questo produrrà un elenco numerato (che è lo stile predefinito):

3. Articolo
4. Qualche altro oggetto
5. Ancora un altro oggetto

È inoltre possibile impostare in modo esplicito un determinato elemento di elenco su un numero specifico. Ulteriori elementi dell'elenco dopo uno con un valore specificato continueranno ad aumentare di uno dal valore di quell'elemento dell'elenco, ignorando dove si trovava l'elenco genitore.

```
<li value="7"></li>
```

Vale anche la pena notare che, utilizzando l'attributo `value` direttamente su un elemento della lista, è possibile sovrascrivere il sistema di numerazione esistente di un elenco ordinato riavviando la numerazione ad un valore inferiore. Quindi, se l'elenco genitore aveva già raggiunto il valore 7 e aveva rilevato un elemento di elenco al valore 4, quell'elemento dell'elenco sarebbe ancora visualizzato come 4 e continuerà a contare da quel punto nuovamente.

```
<ol start="5">
  <li>Item</li>
  <li>Some Other Item</li>
  <li value="4">A Reset Item</li>
  <li>Another Item</li>
  <li>Yet Another Item</li>
</ol>
```

Quindi l'esempio sopra produrrà una lista che segue il modello di numerazione di 5, 6, 4, 5, 6 - ricominciando da un numero inferiore al precedente e duplicando il numero 6 nella lista.

Nota: gli attributi `start` e `value` accettano solo un numero, anche se l'elenco ordinato è impostato per la visualizzazione come numeri o lettere in caratteri romani.

5

È possibile invertire la numerazione con l'aggiunta di `reversed` nel vostro `ol` elemento:

```
<ol reversed>
  <li>Item</li>
  <li>Some Other Item</li>
  <li value="4">A Reset Item</li>
  <li>Another Item</li>
  <li>Yet Another Item</li>
</ol>
```

La numerazione inversa è utile se si aggiunge continuamente a un elenco, ad esempio con nuovi episodi o presentazioni di podcast e si desidera che vengano visualizzati per primi gli elementi più recenti.

Modifica del tipo di numero

Puoi facilmente cambiare il tipo di numero mostrato nel marcatore dell'elemento di lista usando l'attributo `type`

```
<ol type="1|a|A|i|I">
```

genere	Descrizione	Esempi
1	Valore predefinito: numeri decimali	1,2,3,4
a	Ordine alfabetico (lettere minuscole)	a, b, c, d
A	Ordine alfabetico (maiuscolo)	A, B, C, D
i	Numeri romani (lettere minuscole)	i, ii, iii, iv
I	Numeri romani (maiuscoli)	I, II, III, IV

È necessario utilizzare `ol` per visualizzare un elenco di elementi, in cui gli articoli sono stati ordinati intenzionalmente e l'ordine deve essere enfatizzato. Se la modifica dell'ordine degli articoli NON rende l'elenco errato, devi usare `<ul>`.

Elenco di descrizione

Un elenco di descrizione (o *elenco di definizioni*, come veniva chiamato prima di HTML5) può essere creato con l'elemento `dl`. Consiste di gruppi nome-valore, dove il nome è dato nell'elemento `dt` e il valore è dato nell'elemento `dd`.

```
<dl>
  <dt>name 1</dt>
  <dd>value for 1</dd>
  <dt>name 2</dt>
  <dd>value for 2</dd>
</dl>
```

Dimostrazione dal vivo

Un gruppo nome-valore può avere più di un nome e / o più di un valore (che rappresentano alternative):

```
<dl>

  <dt>name 1</dt>
  <dt>name 2</dt>
  <dd>value for 1 and 2</dd>

  <dt>name 3</dt>
  <dd>value for 3</dd>
  <dd>value for 3</dd>

</dl>
```


Liste annidate

È possibile nidificare elenchi per rappresentare voci secondarie di un elemento di elenco.

```
<ul>
  <li>item 1</li>
  <li>item 2
    <ul>
      <li>sub-item 2.1</li>
      <li>sub-item 2.2</li>
    </ul>
  </li>
  <li>item 3</li>
</ul>
```

- articolo 1
- elemento 2
 - sotto-voce 2.1
 - sottovoce 2.2
- articolo 3

L'elenco nidificato deve essere figlio dell'elemento `li` .

Puoi nidificare anche diversi tipi di lista:

```
<ol>
  <li>Hello, list!</li>
  <li>
    <ul>
      <li>Hello, nested list!</li>
    </ul>
  </li>
</ol>
```

Leggi elenchi online: <https://riptutorial.com/it/html/topic/393/elenchi>

Capitolo 21: Entità personaggio

Examples

Caratteri speciali comuni

Alcuni caratteri possono essere riservati per HTML e non possono essere utilizzati direttamente in quanto potrebbero ostruire i codici HTML effettivi. Ad esempio, provare a visualizzare le parentesi angolari sinistra e destra (< >) nel codice sorgente potrebbe causare risultati imprevisti nell'output. Allo stesso modo, gli spazi bianchi come scritti nel codice sorgente potrebbero non essere visualizzati come previsto nell'output HTML. Alcuni, come ☎, non sono disponibili nel set di caratteri ASCII.

A tale scopo, vengono create entità di carattere. Questi sono di forma `&entity_name;` o `&entity_number;`. Le seguenti sono alcune delle entità HTML disponibili.

Personaggio	Descrizione	Nome dell'entità	Numero di entità
" "	spazio irrinunciabile	<code>&nbsp;</code>	<code>&#160;</code>
"<"	meno di	<code>&lt;</code>	<code>&#60;</code>
">"	più grande di	<code>&gt;</code>	<code>&#62;</code>
"&"	ampersand	<code>&amp;</code>	<code>&#38;</code>
"_"	trattino	<code>&mdash;</code>	<code>&#8212;</code>
"-"	it dash	<code>&ndash;</code>	<code>&#8211;</code>
"©"	diritto d'autore	<code>&copy;</code>	<code>&#169;</code>
"®"	marchio registrato	<code>&reg;</code>	<code>&#174;</code>
"™"	marchio	<code>&trade;</code>	<code>&#8482;</code>
"☎"	Telefono	<code>&phone;</code>	<code>&#9742;</code>

Quindi, per scrivere

© 2016 Stack Exchange Inc.

viene utilizzato il seguente codice HTML:

```
<b>&copy; 2016 Stack Exchange Inc.</b>
```

Entità dei caratteri in HTML

Molti simboli e caratteri speciali sono richiesti durante lo sviluppo di una pagina Web in html, ma come sappiamo che a volte l'uso di caratteri può interferire con il codice html che ha alcuni caratteri riservati e alcuni caratteri non disponibili sulla tastiera. Quindi, per evitare il conflitto e allo stesso tempo essere in grado di utilizzare simboli diversi nel nostro codice w3 org ci fornisce "Entità caratteri".

Le entità dei caratteri sono predefinite con "Nome entità" - & nome_entità; e "Numero di entità" - & numero_entità; quindi dobbiamo usare uno dei due affinché il simbolo richiesto sia visualizzato sulla nostra pagina.

L'elenco di poche entità di caratteri può essere trovato su <https://dev.w3.org/html5/html-author/charref>

Un semplice esempio con l'uso dell'entità carattere per "lente d'ingrandimento":

```
<input type="text" placeholder=" 🔍 Search" />
```

che rende come

Leggi Entità personaggio online: <https://riptutorial.com/it/html/topic/5229/entita-personaggio>

Capitolo 22: Formattazione del testo

introduzione

Mentre la maggior parte dei tag HTML viene utilizzata per creare elementi, HTML fornisce anche tag di formattazione nel testo per applicare specifici stili di testo a porzioni di testo. Questo argomento include esempi di formattazione del testo HTML come evidenziazione, grassetto, sottolineatura, pedice e testo colpito.

Sintassi

- `<abbr>Abbreviation</abbr>`
- `<b>Bold Text</b>`
- `<del>Deleted Text</del>`
- `<em>Emphasized Text</em>`
- `<i>Italic Text</i>`
- `<ins>Inserted Text</ins>`
- `<mark>Marked (or Highlighted) Text</mark>`
- `<s>Stricken Text</s>`
- `<strong>Strong Text</strong>`
- `<sub>Subscript Text</sub>`
- `<sup>Superscript Text</sup>`
- `<u>Underlined Text</u>`

Examples

Grassetto, Corsivo e Sottolineato

Testo grassetto

In grassetto, utilizza i `<strong>` o `<b>` :

```
<strong>Bold Text Here</strong>
```

o

```
<b>Bold Text Here</b>
```

Qual è la differenza? Semantica. `<strong>` viene utilizzato per indicare che il testo è fondamentale o semanticamente *importante* per il testo circostante, mentre `<b>` non indica tale importanza e rappresenta semplicemente il testo che deve essere evidenziato in grassetto.

Se dovessi usare `<b>` un programma di sintesi vocale non direi la parola (e) in modo diverso rispetto a qualsiasi altra parola intorno ad esso - stai semplicemente attirando l'attenzione su di loro senza aggiungere alcuna importanza aggiuntiva. Usando `<strong>` , però, lo stesso programma vorrebbe pronunciare quelle parole con un diverso tono di voce per comunicare che il

testo è importante in qualche modo.

Testo in corsivo

Per scrivere in corsivo il testo, usa i `<em>` o `<i>` :

```
<em>Italicized Text Here</em>
```

o

```
<i>Italicized Text Here</i>
```

Qual è la differenza? Semantica. `<em>` è usato per indicare che il testo dovrebbe avere un'enfasi aggiuntiva che dovrebbe essere sottolineata, mentre `<i>` rappresenta semplicemente un testo che dovrebbe essere separato dal normale testo che lo circonda.

Ad esempio, se si desidera sottolineare l'azione all'interno di una frase, è possibile farlo sottolineando in corsivo tramite `<em>` : "Dovresti solo *inviare* la modifica già?"

Ma se stavi identificando un libro o un giornale che in genere dovresti scrivere in corsivo stilisticamente, useresti semplicemente `<i>` : "Sono stato costretto a leggere *Romeo e Giulietta* al liceo."

Testo sottolineato

Mentre l'elemento `<u>` stesso era deprecato in HTML 4, è stato reintrodotta con un significato semantico alternativo in HTML 5 - per rappresentare un'annotazione non testuale, non testuale. È possibile utilizzare un rendering di questo tipo per indicare il testo scritto erroneamente sulla pagina o per un marchio di nome proprio cinese.

```
<p>This paragraph contains some <u>mispelled</u> text.</p>
```

evidenziando

L'elemento `<mark>` è nuovo in HTML5 e viene utilizzato per contrassegnare o evidenziare il testo in un documento "a causa della sua rilevanza in un altro contesto". ¹

L'esempio più comune sarebbe nei risultati di una ricerca in cui l'utente ha inserito una query di ricerca e i risultati sono mostrati evidenziando la query desiderata.

```
<p>Here is some content from an article that contains the <mark>searched query</mark> that we are looking for. Highlighting the text will make it easier for the user to find what they are looking for.</p>
```

Produzione:

Here is some content from an article that contains the **searched query** that we are looking for. Highlighting the text will make it easier for the user to find what they are looking for.

Una formattazione standard comune è un testo nero su sfondo giallo, ma può essere modificato con CSS.

Inserito, eliminato o colpito

Per contrassegnare il testo come inserito, utilizza il tag `<ins>` :

```
<ins>New Text</ins>
```

Per contrassegnare il testo come eliminato, utilizza il tag `<del>` :

```
<del>Deleted Text</del>
```

Per colpire il testo, usa il tag `<s>` :

```
<s>Struck-through text here</s>
```

Apice e pedice

Per spostare il testo verso l'alto o verso il basso, puoi utilizzare i tag `<sup>` e `<sub>` .

Per creare un apice:

```
<sup>superscript here</sup>
```

Per creare pedice:

```
<sub>subscript here</sub>
```

Abbreviazione

Per contrassegnare alcune espressioni come abbreviazioni, usa il tag `<abbr>` :

```
<p>I like to write <abbr title="Hypertext Markup Language">HTML</abbr>!</p>
```

Se presente, l'attributo `title` viene utilizzato per presentare la descrizione completa di tale abbreviazione.

Leggi Formattazione del testo online: <https://riptutorial.com/it/html/topic/526/formattazione-del-testo>

Capitolo 23: HTML 5 Cache

Osservazioni

Il file manifest è un semplice file di testo, che indica al browser cosa memorizzare nella cache (e cosa non memorizzare mai nella cache). L'estensione del file consigliata per i file manifest è: ".appcache" Il file manifest ha tre sezioni:

CACHE MANIFEST - I file elencati sotto questa intestazione verranno memorizzati nella cache dopo essere stati scaricati per la prima volta

NETWORK - I file elencati sotto questa intestazione richiedono una connessione al server e non verranno mai memorizzati nella cache

FALLBACK - I file elencati sotto questa intestazione specificano le pagine di fallback se una pagina non è accessibile

Examples

Esempio di base della cache Html 5

questo è il nostro file index.html

```
<!DOCTYPE html>
<html manifest="index.appcache">
<body>
  <p>Content</p>
</body>
</html>
```

quindi creeremo il file index.appcache con i seguenti codici

```
CACHE MANIFEST
index.html
```

scrivi quei file che vuoi salvare nella cache, carica index.html, poi vai in modalità offline e ricarica la scheda

Nota: i due file devono essere nella stessa cartella in questo esempio

Leggi HTML 5 Cache online: <https://riptutorial.com/it/html/topic/8024/html-5-cache>

Capitolo 24: I paragrafi

introduzione

I paragrafi sono l'elemento HTML più elementare. Questo argomento spiega e dimostra l'utilizzo dell'elemento paragrafo in HTML.

Parametri

Colonna	Colonna
<p>	Definisce un paragrafo
 	Inserisce un'interruzione di riga singola
<pre>	Definisce il testo preformattato

Examples

Paragrafi HTML

L'elemento HTML <p> definisce un **paragrafo** :

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

Display-

Non puoi essere sicuro di come verrà visualizzato l'HTML.

Schermi grandi o piccoli e finestre ridimensionate creeranno risultati diversi.

Con HTML, non puoi modificare l'output aggiungendo spazi extra o linee extra nel tuo codice HTML.

Il browser rimuoverà spazi aggiuntivi e linee extra quando viene visualizzata la pagina:

```
<p>This is          another    paragraph, extra spaces  will be  removed by
browsers</p>
```

Leggi I paragrafi online: <https://riptutorial.com/it/html/topic/7997/i-paragrafi>

Capitolo 25: IFrame

Parametri

Attributo	Dettagli
<code>name</code>	Consente di impostare il nome dell'elemento, per essere utilizzato con un <code>a</code> tag per cambiare del iframe <code>src</code> .
<code>width</code>	Imposta la larghezza dell'elemento in pixel.
<code>height</code>	Imposta l'altezza dell'elemento in pixel.
<code>src</code>	Specifica la pagina che verrà visualizzata nel frame.
<code>srcdoc</code>	Specifica il contenuto che verrà visualizzato nel frame, supponendo che il browser lo supporti. Il contenuto deve essere HTML valido.
<code>sandbox</code>	Quando impostato, il contenuto dell'iframe viene considerato di origine univoca e le funzionalità che includono script, plug-in, moduli e popup saranno disabilitate. Le restrizioni possono essere ridotte in modo selettivo aggiungendo un elenco di valori separati da spazi. Vedere la tabella in Osservazioni per i possibili valori.
<code>allowfullscreen</code>	Se consentire ai contenuti dell'iframe di utilizzare <code>requestFullscreen()</code>

Osservazioni

Un iframe viene utilizzato per incorporare un altro documento nel documento HTML corrente.

È possibile utilizzare gli iframe per la visualizzazione:

- altre pagine HTML sullo stesso dominio;
- altre pagine HTML su un altro dominio (vedi sotto - Politica della stessa origine);
- Documenti PDF (anche se IE potrebbe avere qualche problema, [questa domanda SO](#) potrebbe aiutare);

È necessario utilizzare un iframe come ultima risorsa, in quanto ha problemi con il segnalibro e la navigazione, e ci sono sempre opzioni migliori oltre a un iframe. [Questa domanda SO](#) dovrebbe aiutarti a capire di più sugli alti e bassi degli iframe.

Politica della stessa origine

Alcuni siti non possono essere visualizzati utilizzando un iframe, perché applicano una politica

chiamata [Stessa politica dell'origine](#) . Ciò significa che il sito su cui si trova l'iframe deve trovarsi nello stesso dominio di quello da visualizzare.

Questa politica si applica anche alla manipolazione di contenuti che vivono all'interno di un iFrame. Se iFrame accede a contenuti da un dominio diverso, non sarai in grado di accedere o manipolare il contenuto all'interno di un iFrame.

L'elemento `iframe` su [W3C](#)

attributo `sandbox`

L'attributo `sandbox` , se impostato, aggiunge ulteriori restrizioni all'iframe. Un elenco separato di token può essere usato per attenuare queste restrizioni.

Valore	Dettagli
<code>allow-forms</code>	Consente di inviare moduli.
<code>allow-pointer-lock</code>	Abilita l'API del puntatore JavaScript.
<code>allow-popups</code>	I popup possono essere creati usando <code>window.open()</code> o <code></code>
<code>allow-same-origin</code>	Il documento iframe usa la sua vera origine invece di essere dato a uno unico. Se usato con <code>allow-scripts</code> il documento iframe può rimuovere tutto il sandboxing se è della stessa origine del documento genitore.
<code>allow-scripts</code>	Abilita gli script. Il documento iframe e il documento principale possono essere in grado di comunicare tra loro utilizzando l'API <code>postMessage()</code> . Se usato con <code>allow-same-origin</code> il documento iframe può rimuovere tutto il sandboxing se è della stessa origine del documento genitore.
<code>allow-top-navigation</code>	Consente al contenuto dell'iframe di cambiare la posizione del documento di livello superiore.

Examples

Nozioni di base su una cornice in linea

Il termine "iFrame" significa Inline Frame. Può essere usato per includere un'altra pagina nella tua pagina. Questo produrrà una piccola cornice che mostra il contenuto esatto di `base.html` .

```
<iframe src="base.html"></iframe>
```

Impostazione della dimensione del fotogramma

L'Iframe può essere ridimensionato usando gli attributi `width` e `height` , dove i valori sono rappresentati in pixel (valori di percentuale consentiti HTML 4.01, ma HTML 5 consente solo valori in pixel CSS).

```
<iframe src="base.html" width="800" height="600"></iframe>
```

Usando Ancore con IFrames

Di norma, viene avviata una modifica della pagina Web all'interno di un Iframe con l'Iframe, ad esempio facendo clic su un collegamento all'interno dell'Iframe. Tuttavia, è possibile modificare il contenuto di un IFrame dall'esterno dell'Iframe. È possibile utilizzare un tag di ancoraggio il cui attributo `href` è impostato sull'URL desiderato e il cui attributo `target` è impostato sull'attributo del `name` dell'iframe.

```
<iframe src="webpage.html" name="myIframe"></iframe>
<a href="different_webpage.html" target="myIframe">Change the Iframe content to
different_webpage.html</a>
```

Utilizzando l'attributo "srcdoc"

È possibile utilizzare l'attributo `srcdoc` (anziché l'attributo `src`) per specificare il contenuto esatto dell'iframe come un intero documento HTML. Questo produrrà un IFrame con il testo "Gli IFrame sono fantastici!"

```
<iframe srcdoc="<p>IFrames are cool!</p>"></iframe>
```

Se l'attributo `srcdoc` non è supportato dal browser, l'Iframe ricadrà invece sull'uso dell'attributo `src` , ma se entrambi gli attributi `src` e `srcdoc` sono presenti e supportati dal browser, `srcdoc` ha la precedenza.

```
<iframe srcdoc="<p>IFrames are cool!</p>" src="base.html"></iframe>
```

Nell'esempio precedente, se il browser non supporta l'attributo `srcdoc` , visualizzerà invece il contenuto della pagina `base.html` .

sandboxing

Quanto segue incorpora una pagina Web non affidabile con tutte le restrizioni abilitate

```
<iframe sandbox src="http://example.com/"></iframe>
```

Per consentire alla pagina di eseguire script e inviare moduli, aggiungere `allow-scripts` e `allow-forms` `sandbox` attribute .

```
<iframe sandbox="allow-scripts allow-forms" src="http://example.com/"></iframe>
```

Se c'è un contenuto non attendibile (come i commenti degli utenti) sullo stesso dominio della pagina web padre, un iframe può essere utilizzato per disabilitare gli script pur consentendo al documento principale di interagire con il suo contenuto utilizzando JavaScript.

```
<iframe sandbox="allow-same-origin allow-top-navigation"
src="http://example.com/untrusted/comments/page2">
```

Il documento principale può aggiungere listener di eventi e ridimensionare l'IFrame per adattarlo ai suoi contenuti. Questo, insieme a `allow-top-navigation`, può far sembrare che l'iframe sandbox sia parte del documento genitore.

Questa sandbox non sostituisce l'input di sanificazione ma può essere utilizzata come parte di una strategia di [difesa in profondità](#).

Sappi anche che questa sandbox può essere sovvertita da un utente malintenzionato che convince un utente a visitare direttamente la sorgente dell'iframe. L'intestazione HTTP della [politica di sicurezza del contenuto](#) può essere utilizzata per mitigare questo attacco.

Leggi IFrame online: <https://riptutorial.com/it/html/topic/499/iframe>

Capitolo 26: immagini

Sintassi

- `<img src="" alt="">`

Parametri

parametri	Dettagli
src	Specifica l'URL dell'immagine
srcset	Immagini da utilizzare in diverse situazioni (ad es. Display ad alta risoluzione, piccoli monitor, ecc.)
sizes	Dimensioni dell'immagine tra i punti di interruzione
crossorigin	Come l'elemento gestisce le richieste di crossorigin
usemap	Nome della mappa immagine da utilizzare
ismap	Se l'immagine è una mappa immagine lato server
alt	Testo alternativo che dovrebbe essere visualizzato se per qualche motivo l'immagine non può essere visualizzata
width	Specifica la larghezza dell'immagine (opzionale)
height	Specifica l'altezza dell'immagine (opzionale)

Examples

Creare un'immagine

Per aggiungere un'immagine a una pagina, usa il tag immagine.

I tag immagine (`img`) non hanno tag di chiusura. I due attributi principali che si danno al tag `img` sono `src` , l'origine dell'immagine e `alt` , che è un testo alternativo che descrive l'immagine.

```

```

Puoi anche ottenere immagini da un URL web:

```

```

Nota: le immagini non sono tecnicamente inserite in una pagina HTML, le immagini sono collegate a pagine HTML. Il tag `<img>` crea uno spazio di attesa per l'immagine di riferimento.

È anche possibile incorporare immagini direttamente all'interno della pagina utilizzando base64:

```

```

Suggerimento: per collegare un'immagine a un altro documento, annidare semplicemente il tag `<img>` all'interno dei tag `<a>`.

Larghezza e altezza dell'immagine

Nota: gli attributi `width` e `height` *non* sono deprecati sulle immagini e non sono mai stati. Il loro uso è stato reso molto più severo.

Le dimensioni di un'immagine possono essere specificate utilizzando gli attributi `width` e `height` del tag immagine:

```

```

Specificando la `width` e l' `height` di un'immagine, la struttura fornisce al browser un suggerimento su come dovrebbe essere la pagina, anche se si sta solo specificando la dimensione effettiva dell'immagine. Se le dimensioni dell'immagine non sono specificate, il browser dovrà ricalcolare il layout della pagina dopo che l'immagine è stata caricata, il che potrebbe causare il "saltare" durante il caricamento della pagina.

4.1

È possibile assegnare all'immagine una larghezza e un'altezza nel numero di pixel CSS o in una percentuale delle dimensioni effettive dell'immagine.

Questi esempi sono tutti validi:

```




```

5

La larghezza e l'altezza dell'immagine devono essere specificate nei pixel CSS; un valore percentuale non è più un valore valido. Inoltre, se vengono specificati entrambi gli attributi, devono inserirsi in [una delle tre formule](#) che mantengono le proporzioni. Sebbene valido, non è necessario utilizzare gli attributi `width` e `height` per allungare un'immagine a una dimensione più grande.

Questi esempi sono validi:

```


```

```

```

Questo esempio non è raccomandato:

```

```

Questi esempi non sono validi:

```
  

```

Scegliere il testo alternativo

Alt-text è utilizzato dagli screen reader per gli utenti ipovedenti e dai motori di ricerca. È quindi importante scrivere un buon testo alternativo per le tue immagini.

Il testo dovrebbe apparire corretto anche se si sostituisce l'immagine con il suo attributo alt. Per esempio:

```
<!-- Incorrect -->  
 An anonymous user wrote:  
<blockquote>Lorem ipsum dolor sed.</blockquote>  
<a href="https://google.com/"></a> /  
<a href="https://google.com/"></a>
```

Senza le immagini, sembrerebbe:

Avatar utente anonimo Un utente anonimo ha scritto:

Lorem ipsum dolor sed.

[Modifica icona](#) / [Elimina icona](#)

Per correggere questo:

- Rimuovi il testo alternativo per l'avatar. Questa immagine aggiunge informazioni agli utenti vedenti (un'icona facilmente identificabile per mostrare che l'utente è anonimo) ma queste informazioni sono già disponibili nel testo. ¹
- Rimuovi l'icona dal testo alternativo per le icone. Sapere che questa sarebbe un'icona se fosse lì non aiuta a comunicare il suo vero scopo.

```
<!-- Correct -->  
 An anonymous user wrote:  
<blockquote>Lorem ipsum dolor sed.</blockquote>  
<a href="https://google.com/"></a> /  
<a href="https://google.com/"></a>
```

Un utente anonimo ha scritto:

Lorem ipsum dolor sed.

Le note

¹ C'è una differenza semantica tra l'inclusione di un attributo alt vuoto e l'esclusione del tutto. Un attributo alt vuoto indica che l'immagine *non* è una parte fondamentale del contenuto (come è vero in questo caso - è solo un'immagine additiva che non è necessaria per comprendere il resto) e quindi può essere omessa dal rendering. Tuttavia, la mancanza di un attributo alt indica che l'immagine è una parte fondamentale del contenuto e che semplicemente non esiste alcun equivalente testuale disponibile per il rendering.

Immagine reattiva usando l'attributo srcset

Usando srcset con le dimensioni

```

```

`sizes` sono come le domande dei media, che descrivono lo spazio che l'immagine assume nella vista.

- se viewport è più grande di 1200px, l'immagine è esattamente 580px (ad esempio il nostro contenuto è centrato in un contenitore con una larghezza massima di 1200 px. L'immagine ne prende la metà, meno i margini).
- se viewport è tra 640px e 1200px, l'immagine prende il 48% di viewport (ad esempio, l'immagine viene ridimensionata con la nostra pagina e occupa metà larghezza della vista meno i margini).
- se viewport è di qualsiasi altra dimensione, nel nostro caso inferiore a 640px, l'immagine prende il 98% di viewport (ad esempio, l'immagine viene ridimensionata con la nostra pagina e occupa tutta la larghezza del viewport meno i margini). **Le condizioni del supporto devono essere omesse per l'ultimo articolo.**

`srcset` sta solo dicendo al browser quali immagini abbiamo a disposizione e quali sono le loro dimensioni.

- `img/hello-300.jpg` è largo 300 px,
- `img/hello-600.jpg` è largo `img/hello-600.jpg` px,
- `img/hello-900.jpg` è largo 900px,
- `img/hello-1200.jpg` è largo 1200px

`src` è sempre una sorgente di immagini obbligatoria. In caso di utilizzo con `srcset`, `src` mostrerà

un'immagine di riserva nel caso in cui il browser non supporti `srcset` .

Utilizzo di `srcset` senza dimensioni

```

```

`srcset` fornisce un elenco di immagini disponibili, con descrittore `x` rapporto dispositivo-pixel.

- se il rapporto pixel del dispositivo è 1, usa `img/hello-300.jpg`
- se il rapporto pixel del dispositivo è 2, usa `img/hello-600.jpg`
- se il rapporto pixel del dispositivo è 3, usa `img/hello-1200.jpg`

`src` è sempre una sorgente di immagini obbligatoria. In caso di utilizzo con `srcset` , `src` mostrerà un'immagine di riserva nel caso in cui il browser non supporti `srcset` .

Immagine reattiva usando l'elemento dell'immagine

Codice

```
<picture>
  <source media="(min-width: 600px)" srcset="large_image.jpg">
  <source media="(min-width: 450px)" srcset="small_image.jpg">
  
</picture>
```

uso

Per visualizzare immagini diverse con larghezza dello schermo diversa, è necessario includere tutte le immagini utilizzando il tag sorgente in un tag immagine come mostrato nell'esempio precedente.

Risultato

- Su schermi con larghezza dello schermo > 600px, mostra `large_image.jpg`
- Su schermi con larghezza dello schermo > 450 px, mostra `small_image.jpg`
- Su schermi con altra larghezza dello schermo, mostra `default_image.jpg`

Leggi immagini online: <https://riptutorial.com/it/html/topic/587/immagini>

Capitolo 27: Includi codice JavaScript in HTML

Sintassi

- `<script type="text/javascript"> //some code </script>`
- `<script type="text/javascript" src="URL"></script>`
- `<script type="text/javascript" src="URL" async>//async code</script>`

Parametri

Attributo	Dettagli
src	Specifica il percorso di un file JavaScript. O un URL relativo o assoluto.
type	Specifica il tipo MIME. Questo attributo è obbligatorio in HTML4, ma facoltativo in HTML5.
async	Specifica che lo script deve essere eseguito in modo asincrono (solo per script esterni). Questo attributo non richiede alcun valore (eccetto XHTML).
defer	Specifica che lo script deve essere eseguito quando la pagina ha terminato l'analisi (solo per script esterni). Questo attributo non richiede alcun valore (eccetto XHTML).
charset	Specifica la codifica dei caratteri utilizzata in un file di script esterno, ad esempio UTF-8
crossorigin	Come l'elemento gestisce le richieste di crossorigin
nonce	Il nonce crittografico utilizzato in <i>Content Security Policy</i> controlla CSP3

Osservazioni

Se il codice JavaScript incorporato (file) viene utilizzato per manipolare <http://stackoverflow.com/documentation/javascript/503/document-object-model-dom> Elements, posiziona i `<script></script>` subito **prima della chiusura** `</body>` **tag** o utilizza metodi o librerie JavaScript (come [jQuery](#) per gestire una varietà di browser) che assicura che il DOM sia letto e pronto per essere manipolato.

Examples

Collegamento a un file JavaScript esterno

```
<script src="example.js"></script>
```

L'attributo `src` funziona come l'attributo `href` su `anchor`: puoi specificare un URL assoluto o relativo. L'esempio sopra rimanda a un file all'interno della stessa directory del documento HTML. Questo è tipicamente aggiunto all'interno dei tag `<head>` nella parte superiore del documento html

Include direttamente il codice JavaScript

Invece di collegare a un file esterno, puoi anche includere il codice JS così com'è nel tuo codice HTML:

```
<script>
// JavaScript code
</script>
```

Incluso un file JavaScript in esecuzione in modo asincrono

```
<script type="text/javascript" src="URL" async></script>
```

Gestione Javascript disabilitato

È possibile che il browser del client non supporti Javascript o che l'esecuzione di Javascript sia disabilitata, forse per motivi di sicurezza. Per poter dire agli utenti che si suppone che uno script venga eseguito nella pagina, è possibile utilizzare il tag `<noscript>`. Il contenuto di `<noscript>` viene visualizzato ogni volta che Javascript è disabilitato per la pagina corrente.

```
<script>
  document.write("Hello, world!");
</script>
<noscript>This browser does not support Javascript.</noscript>
```

Leggi [Includi codice JavaScript in HTML online](https://riptutorial.com/it/html/topic/3719/includi-codice-javascript-in-html): <https://riptutorial.com/it/html/topic/3719/includi-codice-javascript-in-html>

Capitolo 28: Incorporare

Parametri

parametri	Dettagli
src	Indirizzo della risorsa
type	Tipo di risorsa incorporata
width	Dimensione orizzontale
height	Dimensione verticale

Examples

Utilizzo di base

Il tag `embed` è nuovo in HTML5. Questo elemento fornisce un punto di integrazione per un'applicazione esterna (in genere non HTML) o un contenuto interattivo.

```
<embed src="myflash.swf">
```

Definire il tipo MIME

Il tipo **MIME** deve essere definito utilizzando l'attributo `type`.

```
<embed type="video/mp4" src="video.mp4" width="640" height="480">
```

Leggi Incorporare online: <https://riptutorial.com/it/html/topic/8123/incorporare>

Capitolo 29: Input Control Elements

introduzione

Un componente chiave dei sistemi web interattivi, i tag di input sono elementi HTML progettati per assumere una forma specifica di input dagli utenti. Diversi tipi di elementi di input possono regolare i dati immessi per adattarsi a un formato specificato e fornire sicurezza per l'immissione della password.

Sintassi

- `<input type="" name="" value="">`

Parametri

Parametro	Dettagli
classe	Indica la classe dell'ingresso
id	Indica l'ID dell'ingresso
genere	Identifica il tipo di controllo di input da visualizzare. I valori accettabili sono <code>hidden</code> , <code>text</code> , <code>tel</code> , <code>url</code> , <code>email</code> , <code>password</code> , <code>date</code> , <code>time</code> , <code>number</code> , <code>range</code> , <code>color</code> , <code>checkbox</code> , <code>radio</code> , <code>file</code> , <code>submit</code> , <code>image</code> , <code>reset</code> e <code>button</code> . Il valore predefinito è <code>text</code> se non specificato, se il valore non è valido o se il browser non supporta il tipo specificato.
nome	Indica il nome dell'ingresso
Disabilitato	Valore booleano che indica che l'input deve essere disabilitato. I controlli disabilitati non possono essere modificati, non vengono inviati all'invio del modulo e non possono ricevere lo stato attivo.
verificato	Quando il valore dell'attributo <code>type</code> è <code>radio</code> o <code>checkbox</code> , la presenza di questo attributo booleano indica che il controllo è selezionato per impostazione predefinita; altrimenti viene ignorato.
multiplo	HTML5 Indica che è possibile passare più file o valori (si applica solo agli input di tipo <code>file</code> e <code>email</code>)
segnaposto	HTML5 Un suggerimento all'utente di ciò che può essere inserito nel controllo. Il testo segnaposto non deve contenere ritorni a capo o line feed
completamento automatico	HTML5 Indica se il valore del controllo può essere completato automaticamente dal browser.

Parametro	Dettagli
sola lettura	Valore booleano che indica che l'input non è modificabile. I comandi di sola lettura sono ancora inviati sull'invio del modulo, ma non riceveranno il focus. HTML5: questo attributo viene ignorato quando il valore dell'attributo <code>type</code> è impostato su <code>hidden</code> , <code>range</code> , <code>color</code> , <code>checkbox</code> , <code>radio</code> , <code>file</code> o <code>button</code> .
necessario	HTML5 Indica che un valore deve essere presente o che l'elemento deve essere controllato per poter inviare il modulo
alt	Un testo alternativo per le immagini, nel caso in cui non vengano visualizzate.
messa a fuoco automatica	L'elemento <code><input></code> dovrebbe ottenere lo stato attivo quando viene caricata la pagina.
valore	Specifica il valore dell'elemento <code><input></code> .
passo	L'attributo <code>step</code> specifica gli intervalli numerici legali. Funziona con i seguenti tipi di input: <code>number</code> , <code>range</code> , <code>date</code> , <code>date</code> <code>datetime-local</code> , <code>month</code> , <code>time</code> e <code>week</code> .

Osservazioni

Come con altri elementi void HTML5, `<input>` è a chiusura automatica e può essere scritto `<input />` . HTML5 non richiede questa barra.

I seguenti sono tipi di input validi in HTML:

- `button`
- `checkbox`
- `file`
- `hidden`
- `image`
- `password`
- `radio`
- `reset`
- `submit`
- `text` (valore predefinito)

5

Di seguito sono riportati i tipi di input introdotti come parte dello standard HTML 5. Alcuni di questi tipi non sono supportati da tutti i browser web. Nel caso in cui un tipo non sia supportato, l'elemento di input verrà impostato automaticamente sul tipo di `text` .

- `color`
- `date`
- `datetime` (Obsoleto e obsoleto)

- `datetime-local`
- `email`
- `month`
- `number`
- `range`
- `search`
- `tel`
- `time`
- `url`
- `week`

Per verificare quali browser supportano i vari tipi, puoi andare su caniuse.com .

Examples

Casella di controllo e pulsanti di opzione

Panoramica

Caselle di controllo e pulsanti radio sono scritti con il tag HTML `<input>` e il loro comportamento è definito nelle [specifiche HTML](#) .

La casella di controllo o il pulsante di opzione più semplice è un elemento `<input>` con un attributo `type` di `checkbox` o `radio` , rispettivamente:

```
<input type="checkbox">
<input type="radio">
```

Un singolo elemento di casella di controllo stand-alone viene utilizzato per una singola opzione binaria come una domanda sì o no. Le caselle di controllo sono indipendenti, il che significa che l'utente può selezionare tutte le opzioni che desidera in un gruppo di caselle di controllo. In altre parole, il controllo di una casella di controllo *non* deselecta le altre caselle di controllo nel gruppo di caselle di controllo.

I pulsanti di opzione vengono di solito in gruppi (se non sono raggruppati con un altro pulsante di opzione, probabilmente si intende utilizzare una casella di controllo) identificati utilizzando lo stesso attributo del `name` su tutti i pulsanti all'interno di quel gruppo. La selezione dei pulsanti di opzione si *escludono a vicenda* , il che significa che l'utente può selezionare solo una scelta tra un gruppo di pulsanti di opzione. Quando viene selezionato un pulsante di opzione, qualsiasi altro pulsante di opzione con lo stesso `name` precedentemente selezionato viene deselectato.

Esempio:

```
<input type="radio" name="color" id="red" value="#F00">
<input type="radio" name="color" id="green" value="#0F0">
<input type="radio" name="color" id="blue" value="#00F">
```

Se visualizzati, i pulsanti di opzione appaiono come un cerchio (deselezionato) o un cerchio pieno

(selezionato). Le caselle di controllo appaiono come un quadrato (deselezionato) o un quadrato pieno (selezionato). A seconda del browser e del sistema operativo, il quadrato a volte ha angoli arrotondati.

attributi

caselle di controllo e pulsanti di opzione hanno un numero di attributi per controllare il loro comportamento:

`value`

Come qualsiasi altro elemento di input, l'attributo `value` specifica il valore stringa da associare al pulsante in caso di invio del modulo. Tuttavia, le caselle di controllo e i pulsanti di opzione sono speciali in quanto quando il valore viene omissso, viene impostato su `on` quando viene inviato, anziché inviare un valore vuoto. L'attributo `value` non si riflette nell'aspetto del pulsante.

`checked`

L'attributo `checked` specifica lo stato iniziale di una casella di controllo o di un pulsante di opzione. Questo è un attributo booleano e può essere omissso.

Ognuno di questi sono modi equivalenti validi per definire un pulsante di opzione selezionato:

```
<input checked>
<input checked="">
<input checked="checked">
<input checked="ChEcKeD">
```

L'assenza dell'attributo `checked` è l'unica sintassi valida per un pulsante non controllato:

```
<input type="radio">
<input type="checkbox">
```

Quando si ripristina un `<form>`, le caselle di controllo e i pulsanti di opzione tornano allo stato dell'attributo `checked`.

Accessibilità

etichette

Per dare un contesto ai pulsanti e mostrare agli utenti a cosa serve ciascun pulsante, ognuno di essi dovrebbe avere un'etichetta. Questo può essere fatto usando un elemento `<label>` per avvolgere il pulsante. Inoltre, rende l'etichetta selezionabile, quindi si seleziona il pulsante corrispondente.

Esempio:


```
<label>
  <input type="radio" name="color" value="#F00">
  Red
</label>
```

o con un elemento `<label>` con un attributo `for` impostato `id` del pulsante:

```
<input type="checkbox" name="color" value="#F00" id="red">
<label for="red">Red</label>
```

Gruppi di pulsanti

Poiché ciascun pulsante di opzione influisce sugli altri componenti del gruppo, è comune fornire un'etichetta o un contesto per l'intero gruppo di pulsanti di opzione.

Per fornire un'etichetta per l'intero gruppo, i pulsanti radio dovrebbero essere inclusi in un elemento `<fieldset>` con un elemento `<legend>` suo interno.

Esempio:

```
<fieldset>
  <legend>Theme color:</legend>
  <p>
    <input type="radio" name="color" id="red" value="#F00">
    <label for="red">Red</label>
  </p>
  <p>
    <input type="radio" name="color" id="green" value="#0F0">
    <label for="green">Green</label>
  </p>
  <p>
    <input type="radio" name="color" id="blue" value="#00F">
    <label for="blue">Blue</label>
  </p>
</fieldset>
```

Le caselle di controllo possono anche essere raggruppate in modo simile, con un `fieldset` e una legenda che identificano il gruppo di caselle di controllo correlate. Tuttavia, tieni presente che le caselle di controllo *non* devono condividere lo stesso nome perché non si escludono a vicenda. In questo modo il modulo invierà più valori per la stessa chiave e non tutti i linguaggi sul lato server lo gestiranno allo stesso modo (comportamento non definito). Ogni casella di controllo dovrebbe avere un nome univoco o utilizzare un insieme di parentesi quadre (`[]`) per indicare che il modulo deve inviare una matrice di valori per quella chiave. Il metodo scelto dipende dal modo in cui si intende gestire i dati del modulo lato client o lato server. Si dovrebbe anche tenere la leggenda breve, poiché alcune combinazioni di browser e screen reader leggono la legenda prima di ogni campo di input nel `fieldset`.

Nascosto

```
<input type="hidden" name="inputName" value="inputValue">
```

Un input nascosto non sarà visibile all'utente, ma il suo valore verrà inviato al server quando il modulo viene comunque inviato.

Parola d'ordine

```
<input type="password" name="password">
```

L'elemento di input con un attributo `type` il cui valore è `password` crea un campo di testo a riga singola simile al `type=text` `input type=text`, ad eccezione del fatto che il testo non viene visualizzato quando l'utente lo inserisce.

```
<input type="password" name="password" placeholder="Password">
```

Il testo segnaposto viene mostrato in testo normale e viene sovrascritto automaticamente quando un utente inizia a digitare.



Nota: alcuni browser e sistemi modificano il comportamento predefinito del campo della password per visualizzare anche il carattere digitato più recentemente per una durata breve, in questo modo:



Sottoscrivi

```
<input type="submit" value="Submit">
```

Un input di invio crea un pulsante che invia il modulo al suo interno quando viene fatto clic.

Puoi anche utilizzare l'elemento `<button>` se hai bisogno di un pulsante di invio che può essere più facilmente stilizzato o contenere altri elementi:

```
<button type="submit">
   Submit
</button>
```

File

```
<input type="file" name="fileSubmission">
```

Gli input di file consentono agli utenti di selezionare un file dal loro filesystem locale per l'uso con la pagina corrente. Se utilizzati insieme a un elemento del `form`, possono essere utilizzati per consentire agli utenti di caricare file su un server (per ulteriori informazioni, consultare [Caricamento di file](#)).

L'esempio seguente consente agli utenti di utilizzare l'input del `file` per selezionare un file dal proprio filesystem e caricarlo su uno script sul server denominato `upload_file.php`.

```
<form action="upload_file.php" method="post" enctype="multipart/form-data">
  Select file to upload:
  <input type="file" name="fileSubmission" id="fileSubmission">
  <input type="submit" value="Upload your file" name="submit">
</form>
```

Più file

Aggiungendo l'attributo `multiple` l'utente sarà in grado di selezionare **più di un file**:

```
<input type="file" name="fileSubmission" id="fileSubmission" multiple>
```

Accetta file

L'attributo `Accept` specifica i tipi di file che l'utente può selezionare. Ad esempio `.png`, `.gif`, `.jpeg`.

```
<input type="file" name="fileSubmission" accept="image/x-png,image/gif,image/jpeg" />
```

Input Validation

La convalida dell'input HTML viene eseguita automaticamente dal browser in base agli attributi speciali sull'elemento di input. Potrebbe parzialmente o completamente sostituire la convalida dell'input JavaScript. Questo tipo di convalida può essere aggirato dall'utente tramite richieste HTTP appositamente predisposte, quindi non sostituisce la convalida dell'input lato server. La convalida si verifica solo quando si tenta di inviare il modulo, quindi tutti gli input limitati devono essere contenuti in un modulo per consentire la convalida (a meno che non si stia utilizzando JavaScript). Tieni presente che gli input disabilitati o di sola lettura non attivano la convalida.

Alcuni tipi di input più recenti (come `email`, `url`, `tel`, `date` e molti altri) vengono automaticamente convalidati e non richiedono i propri vincoli di convalida.

5

necessario

Utilizzare l'attributo `required` per indicare che un campo deve essere completato per poter passare la convalida.

```
<input required>
```

Lunghezza minima / massima

Utilizzare gli attributi `minlength` e `maxlength` per indicare i requisiti di lunghezza. La maggior parte dei browser impedisce all'utente di digitare più di un *massimo di* caratteri nella casella, impedendo loro di rendere la loro voce non valida ancor prima di tentare l'invio.

```
<input minlength="3">
<input maxlength="15">
<input minlength="3" maxlength="15">
```

Specifica di un intervallo

Utilizza gli attributi `min` e `max` per limitare l'intervallo di numeri che un utente può inserire in un input di tipo `number` o `range`

```
Marks: <input type="number" size="6" name="marks" min="0" max="100" />
Subject Feedback: <input type="range" size="2" name="feedback" min="1" max="5" />
```

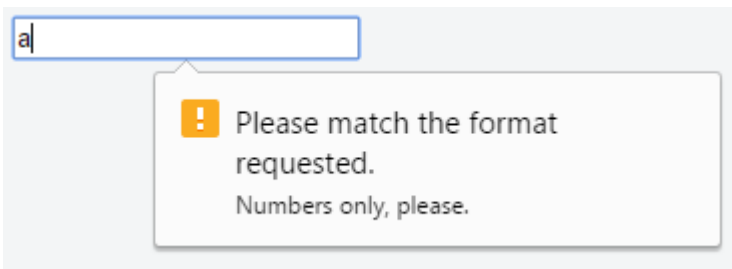
5

Abbina un modello

Per un maggiore controllo, utilizzare l'attributo `pattern` per specificare qualsiasi espressione regolare che deve essere abbinata per poter passare la convalida. È inoltre possibile specificare un `title`, che è incluso nel messaggio di convalida se il campo non passa.

```
<input pattern="\d*" title="Numbers only, please.">
```

Ecco il messaggio mostrato in Google Chrome versione 51 quando tenti di inviare il modulo con un valore non valido all'interno di questo campo:



Non tutti i browser visualizzano un messaggio per i pattern non validi, sebbene vi sia pieno supporto tra i browser più utilizzati.

Controlla l'ultimo supporto su [CanIUse](#) e implementalo di conseguenza.

5

Accetta il tipo di file

Per i campi di input di tipo `file`, è possibile accettare solo determinati tipi di file, come video, immagini, audio, estensioni di file specifiche o determinati [tipi di media](#). Per esempio:

```
<input type="file" accept="image/*" title="Only images are allowed">
```

È possibile specificare più valori con una virgola, ad esempio:

```
<input type="file" accept="image/*,.rar,application/zip">
```

Nota: aggiungendo `novalidate` all'attributo `novalidate` all'elemento `form` o `formnovalidate` al pulsante di invio, si impedisce la convalida sugli elementi del modulo. Per esempio:

```
<form>
  <input type="text" name="name" required>
  <input type="email" name="email" required>
  <input pattern="\d*" name="number" required>

  <input type="submit" value="Publish"> <!-- form will be validated -->
  <input type="submit" value="Save" formnovalidate> <!-- form will NOT be validated -->
</form>
```

Il modulo contiene i campi necessari per "pubblicare" la bozza ma non sono necessari per "salvare" la bozza.

Reset

```
<input type="reset" value="Reset">
```

Un input di tipo `reset` crea un pulsante che, quando viene premuto, reimposta tutti gli input nel modulo in cui è contenuto al loro stato predefinito.

- Il testo in un campo di input verrà reimpostato su bianco o il suo valore predefinito (specificato utilizzando l'attributo `value`).
- Qualsiasi opzione (s) in un menu di selezione verrà deselezionata a meno che non abbiano l'attributo `selected`.
- Tutte le caselle di controllo e le caselle radio saranno deselezionate a meno che non abbiano l'attributo `checked`.

Nota: un pulsante di ripristino deve essere all'interno o collegato a (tramite l'attributo `form`) un elemento `<form>` per avere un effetto. Il pulsante resetterà solo gli elementi all'interno di questo modulo.

Numero

5

```
<input type="number" value="0" name="quantity">
```

L'elemento Input con un attributo `type` il cui valore è `number` rappresenta un controllo preciso per l'impostazione del valore dell'elemento su una stringa che rappresenta un numero.

Si prega di notare che questo campo non garantisce di avere un numero corretto. Permette solo tutti i simboli che potrebbero essere utilizzati in qualsiasi numero reale, ad esempio l'utente sarà in grado di inserire valori come `e1e-,0`.

Tel

```
<input type="tel" value="+8400000000">
```

L'elemento di input con un attributo type il cui valore è `tel` rappresenta un controllo di modifica in testo semplice di una riga per l'immissione di un numero di telefono.

E-mail

5

`<input type="email">` viene utilizzato per i campi di input che devono contenere un indirizzo e-mail.

```
<form>
  <label>E-mail: <label>
    <input type="email" name="email">
  </form>
```

L'indirizzo e-mail può essere automaticamente convalidato quando inviato in base al supporto del browser.

Pulsante

```
<input type="button" value="Button Text">
```

I pulsanti possono essere utilizzati per attivare azioni sulla pagina, senza inviare il modulo. Puoi anche usare l'elemento `<button>` se hai bisogno di un pulsante che può essere più facilmente stilizzato o contenere altri elementi:

```
<button type="button">Button Text</button>
```

I pulsanti vengono in genere utilizzati con un evento "onclick":

```
<input type="button" onclick="alert('hello world!')" value="Click Me">
```

0

```
<button type="button" onclick="alert('hello world!')">Click Me</button>
```

attributi

[name]

Il `name` del pulsante, che viene inviato con i dati del modulo.

[type]

Il `type` del pulsante.

I valori possibili sono:

`submit` : il pulsante invia i dati del modulo al server. Questo è l'impostazione predefinita se l'attributo non è specificato o se l'attributo viene modificato dinamicamente in un valore vuoto o non valido.

`reset` : il pulsante ripristina tutti i controlli ai loro valori iniziali.

`button` : il pulsante non ha un comportamento predefinito. Può avere script sul lato client associati agli eventi dell'elemento, che vengono attivati quando si verificano gli eventi.

`menu` : il pulsante apre un menu popup definito tramite l'elemento designato.

[value]

Il valore iniziale del pulsante.

5

Attributi aggiuntivi per i pulsanti di invio

Attributo	Descrizione
<code>form</code>	Specifica l'ID del modulo a cui appartiene il pulsante. Se nessuno è specificato, apparterrà al suo elemento modulo ancestor (se ne esiste uno).
<code>formaction</code>	Specifica dove inviare i dati del modulo quando il modulo viene inviato utilizzando questo pulsante.
<code>formenctype</code>	Specifica come devono essere codificati i dati del modulo quando lo si invia al server utilizzando questo pulsante. Può essere utilizzato solo con <code>formmethod="post"</code> .
<code>formmethod</code>	Specifica il metodo HTTP da utilizzare (POST o GET) quando si inviano dati di modulo usando questo pulsante.
<code>formnovalidate</code>	Specifica che i dati del modulo non devono essere convalidati in fase di invio.
<code>formtarget</code>	Specifica dove visualizzare la risposta ricevuta dopo aver inviato il modulo utilizzando questo pulsante.

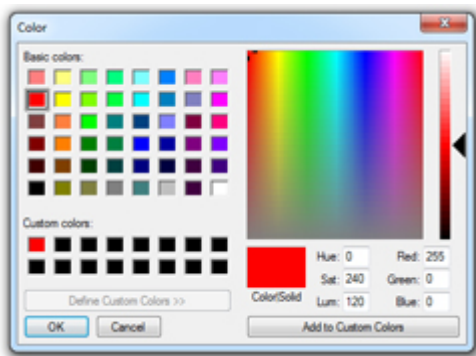
Colore

```
<input type="color" name="favcolor" value="#ff0000">
```

Nel supportare i browser, l'elemento input con un attributo type il cui valore è `color` crea un controllo pulsante, con un colore uguale al valore dell'attributo `color` (il valore predefinito è il nero se il valore non è specificato o è un formato esadecimale non valido).



Facendo clic su questo pulsante si apre il widget dei colori del sistema operativo, che consente all'utente di selezionare un colore.



Il fallback per i browser che non supportano questo tipo di input è un `type=text` [input](#) normale `type=text`.

url

```
<input type="url" name="Homepage">
```

Questo è usato per i campi di input che dovrebbero contenere un indirizzo URL.

A seconda del supporto del browser, il campo `url` può essere automaticamente convalidato quando inviato.

Alcuni smartphone riconoscono il tipo di `url` e aggiunge ".com" alla tastiera per far corrispondere l'input dell'URL.

Data

```
<input type="date" />
```

Sullo schermo comparirà un selezionatore di date per scegliere una data. Questo non è

supportato in Firefox o Internet Explorer.

DateTime-locale

5

```
<input type="datetime-local" />
```

A seconda del supporto del browser, sullo schermo verrà visualizzato un selettore di data e ora che consente di scegliere una data e un'ora.

Immagine

```
<input type="image" src="img.png" alt="image_name" height="50px" width="50px"/>
```

Un'immagine. È necessario utilizzare l'attributo src per definire la fonte dell'immagine e l'attributo alt per definire il testo alternativo. È possibile utilizzare gli attributi di altezza e larghezza per definire la dimensione dell'immagine in pixel.

Gamma

5

```
<input type="range" min="" max="" step="" />
```

Un controllo per inserire un numero il cui valore esatto non è importante.

Attributo	Descrizione	Valore predefinito
min	Valore minimo per intervallo	0
max	Valore massimo per intervallo	100
passo	Quantità da aumentare di ogni incremento.	1

Mese

5

```
<input type="month" />
```

A seconda del supporto del browser, verrà mostrato un controllo per selezionare il mese.

Tempo

5

```
<input type="time" />
```

L'input di `time` contrassegna questo elemento come accettando una stringa che rappresenta un tempo. Il formato è definito in [RFC 3339](#) e dovrebbe essere un tempo parziale come

```
19:04:39  
08:20:39.04
```

Attualmente, tutte le versioni di Edge, Chrome, Opera e Chrome per Android supportano `type = "time"`. Le versioni più recenti di Android Browser, in particolare 4.4 e versioni successive lo supportano. Safari per iOS offre supporto parziale, non supporta gli attributi `min`, `max` e `step`.

Settimana

5

```
<input type="week" />
```

A seconda del supporto del browser, verrà visualizzato un controllo per l'immissione di un numero di settimana e un numero di settimana senza fuso orario.

Testo

Il tipo di input più semplice e l'input predefinito se non viene specificato alcun `type`. Questo tipo di input definisce un campo di testo a riga singola con interruzioni di riga automaticamente rimosse dal valore di input. Tutti gli altri personaggi possono essere inseriti in questo. `<input>` elementi `<input>` vengono utilizzati all'interno di un elemento `<form>` per dichiarare controlli di input che consentono agli utenti di immettere dati.

Sintassi

```
<input type="text">
```

oppure (senza specificare un `type`, utilizzando l'attributo predefinito):

```
<input>
```

La larghezza predefinita di un campo di testo è di 20 caratteri. Questo può essere modificato specificando un valore per l'attributo `size` come questo:

```
<input type="text" size="50">
```

L'attributo `size` è nettamente diverso rispetto all'impostazione di una larghezza con CSS. L'uso di una larghezza definisce un valore specifico (in numero di pixel, percentuale dell'elemento genitore, ecc.) Che l'input deve essere sempre ampio. L'utilizzo della `size` calcola la quantità di larghezza da allocare in base al carattere utilizzato e alla larghezza dei caratteri.

Nota: l'uso dell'attributo `size` non limita intrinsecamente il numero di caratteri che possono essere inseriti nella casella, ma solo quanto è ampia la casella. Per limitare la lunghezza, vedere [Input Validation](#).

Un campo di input consente solo una riga di testo. Se hai bisogno di un input di testo su più righe per una notevole quantità di testo, usa invece [un elemento `<textarea>`](#).

Ricerca

5

La ricerca del tipo di input viene utilizzata per la ricerca testuale. Sulla barra verrà aggiunto il simbolo della lente accanto allo spazio per la maggior parte dei browser

```
<input type="search" name="googlesearch">
```

DateTime (globale)

L'elemento input con un attributo `type` il cui valore è "**datetime**" rappresenta un controllo per l'impostazione del valore dell'elemento su una stringa che rappresenta una **data e un'ora globali (con informazioni sul fuso orario)**.

```
<fieldset>
  <p><label>Meeting time: <input type=datetime name="meeting.start"></label>
</fieldset>
```

Attributi consentiti:

- attributi globali
- nome
- Disabilitato
- modulo
- genere
- completamento automatico
- messa a fuoco automatica
- elenco
- minimo Massimo
- passo (float)
- sola lettura
- valore richiesto

Leggi [Input Control Elements](https://riptutorial.com/it/html/topic/277/input-control-elements) online: <https://riptutorial.com/it/html/topic/277/input-control-elements>

Capitolo 30: Le forme

introduzione

Per raggruppare elementi di input e inviare dati, HTML utilizza un elemento modulo per incapsulare gli elementi di input e di invio. Queste forme gestiscono l'invio dei dati nel metodo specificato a una pagina gestita da un server o gestore. Questo argomento spiega e dimostra l'utilizzo di moduli HTML nella raccolta e nell'invio di dati di input.

Sintassi

- `<form method="post|get" action="somePage.php" target="_blank|_self|_parent|_top|frameName">`

Parametri

Attributo	Descrizione
accept-charset	Specifica le codifiche dei caratteri che devono essere utilizzate per l'invio del modulo.
action	Specifica dove inviare i dati del modulo quando viene inviato un modulo.
autocomplete	Specifica se un modulo deve avere il completamento automatico attivato o disattivato.
enctype	Specifica come devono essere codificati i dati del modulo quando lo si invia al server (solo per method = "post").
method	Specifica il metodo HTTP da utilizzare quando si inviano i dati del modulo (POST o GET).
name	Specifica il nome di un modulo.
novalidate	Specifica che il modulo non deve essere convalidato quando inviato.
target	Specifica dove visualizzare la risposta ricevuta dopo aver inviato il modulo.

Osservazioni

L'elemento `<form>` rappresenta una sezione che contiene elementi associati al modulo (ad esempio `<button>` `<fieldset>` `<input>` `<label>` `<output>` `<select>` `<textarea>`) che invia informazioni a un server. Sono necessari entrambi i `<form>` inizio (`<form>`) e finale (`</form>`).

Examples

L'attributo di azione

L'attributo `action` definisce l'azione da eseguire quando viene inviato il modulo, che di solito conduce a uno script che raccoglie le informazioni inviate e funziona con esso. se lo lasci vuoto, lo invierà allo stesso file

```
<form action="action.php">
```

L'attributo del metodo

L'attributo `method` viene utilizzato per definire il metodo HTTP del modulo che è GET o POST.

```
<form action="action.php" method="get">  
<form action="action.php" method="post">
```

Il metodo GET è principalmente utilizzato per *ottenere* dati, ad esempio per ricevere un post tramite il suo ID o nome, o per inviare una query di ricerca. Il metodo GET aggiungerà i dati del modulo all'URL specificato nell'attributo `action`.

```
www.example.com/action.php?firstname=Mickey&lastname=Mouse
```

Il metodo POST viene utilizzato quando si inviano dati a uno script. Il metodo POST non aggiunge i dati del modulo all'URL dell'azione ma invia utilizzando il corpo della richiesta.

Per inviare correttamente i dati dal modulo, è necessario specificare un nome di attributo nome. Ad esempio, inviamo il valore del campo e impostiamo il suo nome sul *cognome* :

```
<input type="text" name="lastname" value="Mouse">
```

Più attributi

```
<form action="action.php" method="post" target="_blank" accept-charset="UTF-8"  
enctype="application/x-www-form-urlencoded" autocomplete="off" novalidate>
```

```
<!-- form elements -->
```

```
</form>
```

Attributo di destinazione nel tag modulo

L'attributo `target` specifica un nome o una parola chiave che indica dove visualizzare la risposta ricevuta dopo aver inviato il modulo.

L'attributo `target` definisce un nome o una parola chiave per un contesto di esplorazione (ad

esempio tab, finestra o frame incorporato).

Da tag con un attributo target:

```
<form target="_blank">
```

Valori di attributo

Valore	Descrizione
_blank	La risposta viene visualizzata in una nuova finestra o scheda
_se stesso	La risposta viene visualizzata nello stesso frame (questa è l'impostazione predefinita)
_genitore	La risposta è visualizzata nel frame principale
_superiore	La risposta viene visualizzata nel corpo completo della finestra
framename	La risposta viene visualizzata in un iframe con nome

Nota: l'attributo target è stato **deprecato** in **HTML 4.01** . L'attributo target è **supportato** in **HTML5** .

I frame e i set di frame non sono supportati in **HTML5** , quindi i **valori** **_parent**, **_top** e **framename** **vengono ora utilizzati principalmente con iframe** .

Caricamento di file

Immagini e file possono essere caricati / inviati al server impostando `enctype` attributo `enctype` del tag `form` su `multipart/form-data` . `enctype` specifica come i dati del modulo verrebbero codificati durante l'invio al server.

Esempio

```
<form method="post" enctype="multipart/form-data" action="upload.php">
  <input type="file" name="pic" />
  <input type="submit" value="Upload" />
</form>
```

Raggruppamento di alcuni campi di input

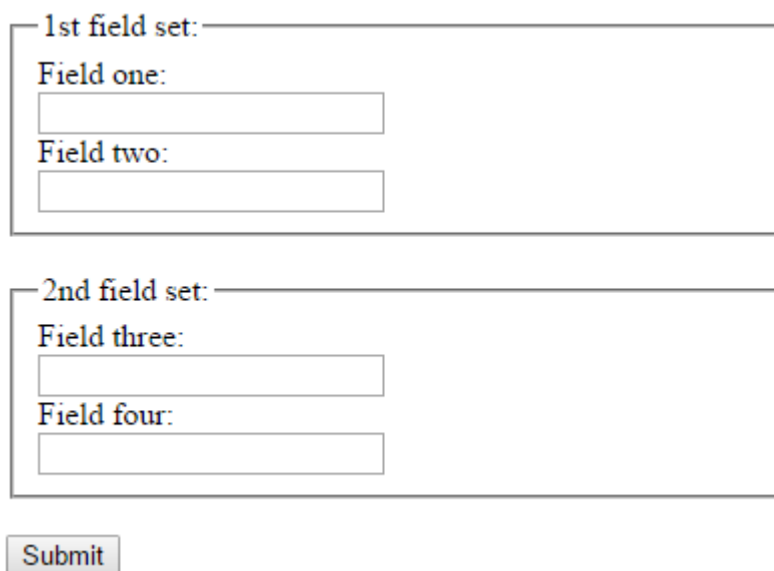
Durante la progettazione di un modulo, è possibile raggruppare alcuni campi di input in un gruppo per organizzare il layout del modulo. Questo può essere fatto usando il tag. Ecco un esempio per utilizzarlo.

Per ogni fieldset, puoi impostare una legenda per il set usando il tag `LEGEND` `TEXT`

Esempio

```
<form>
  <fieldset>
    <legend>1st field set:</legend>
    Field one:<br>
    <input type="text"><br>
    Field two:<br>
    <input type="text"><br>
  </fieldset><br>
  <fieldset>
    <legend>2nd field set:</legend>
    Field three:<br>
    <input type="text"><br>
    Field four:<br>
    <input type="text"><br>
  </fieldset><br>
  <input type="submit" value="Submit">
</form>
```

Risultato



The screenshot shows the rendered HTML form. It consists of two fieldsets, each with a legend and two text input fields. The first fieldset is titled "1st field set:" and contains "Field one:" and "Field two:". The second fieldset is titled "2nd field set:" and contains "Field three:" and "Field four:". Below the fieldsets is a "Submit" button.

Supporto del browser

Chrome, IE, Edge, FireFox, Safari e le ultime versioni di Opera supportano anche il tag

Leggi Le forme online: <https://riptutorial.com/it/html/topic/1160/le-forme>

Capitolo 31: Lingue di contenuto

Sintassi

- `<element lang="language_code">` <! - Il codice della lingua deve essere nel formato [\[ISO 639-1\]](#) (https://en.wikipedia.org/wiki/ISO_639-1) ->

Osservazioni

Il valore dell'attributo `lang` deve essere un **tag di lingua BCP 47** valido o **una stringa vuota** (se la lingua è sconosciuta).

I tag della lingua [BCP 47](#) sono elencati nel [registro sottotag della lingua IANA](#) .

Accessibilità

I criteri di successo WCAG 2.0 rilevanti sono:

- [3.1.1 Lingua della pagina](#)
- [3.1.2 Lingua delle parti](#)

Le relative tecniche WCAG 2.0 sono:

- [H57: utilizzo degli attributi della lingua sull'elemento html](#)
- [H58: utilizzo degli attributi del linguaggio per identificare i cambiamenti nella lingua umana](#)

Examples

Element Language

L'attributo `lang` viene utilizzato per specificare la lingua del contenuto dell'elemento e i valori del testo dell'attributo:

```
<p lang="en">The content of this element is in English.</p>
```

```
<p lang="en" title="The value of this attribute is also in English.">The content of this element is in English.</p>
```

La dichiarazione della lingua viene ereditata:

```
<div lang="en">
  <p>This element contains English content.</p>
  <p title="This attribute, too.">Same with this element.</p>
</div>
```


Elementi con più lingue

È possibile "sovrascrivere" una dichiarazione di lingua:

```
<p lang="en">This English sentence contains the German word <span lang="de">Hallo</span>.</p>
```

Gestione degli attributi con lingue diverse

È possibile "sovrascrivere" la dichiarazione della lingua di un elemento genitore introducendo un elemento diverso da `applet`, `base`, `basefont`, `br`, `frame`, `frameset`, `hr`, `iframe`, `meta`, `param`, `script` (di HTML 4.0) con un proprio attributo `lang`:

```
<p lang="en" title="An English paragraph">  
  <span lang="de" title="A German sentence">Hallo Welt!</span>  
</p>
```

Base Document Language

È buona norma dichiarare la lingua principale del documento nell'elemento `html`:

```
<html lang="en">
```

Se nessun altro attributo `lang` è specificato nel documento, significa che *tutto* (cioè il contenuto dell'elemento e i valori del testo dell'attributo) è in quella lingua.

Se il documento contiene parti in altre lingue, queste parti dovrebbero ottenere i propri attributi `lang` per "sovrascrivere" la dichiarazione della lingua.

URL regionali

È possibile aggiungere l'attributo `hreflang` agli elementi `<a>` e `<area>` che creano collegamenti ipertestuali. Tale specifica la lingua della risorsa collegata. La lingua definita deve essere un tag di lingua [BCP 47](#) ^[1] valido.

```
<p>  
  <a href="example.org" hreflang="en">example.org</a> is one of IANA's example domains.  
</p>
```

1. ↑ Gruppo di lavoro sulla rete IETF: [tag RFC 5646 per l'identificazione delle lingue](#), IETF, settembre 2009

Leggi Lingue di contenuto online: <https://riptutorial.com/it/html/topic/737/lingue-di-contenuto>

Capitolo 32: Mappe di immagini

Sintassi

- `<img usemap="#[map-name]">`
- `<map name="[map-name]"></map>`
- `<area>`

Parametri

Tag / Attributo	Valore
	Di seguito sono riportati gli attributi specifici della mappa immagine da utilizzare con <code></code> . Si applicano gli attributi <code></code> regolari.
usemap	Il <code>name</code> della mappa con un simbolo di hash è stato aggiunto ad esso. Ad esempio, per una mappa con <code>name="map"</code> , l'immagine dovrebbe avere <code>usemap="#map"</code> .
<map>	
name	Il nome della mappa per identificarlo. Da utilizzare con l'attributo <code>usemap</code> dell'immagine.
<area>	Di seguito sono riportati gli attributi specifici <code><area></code> . Quando <code>href</code> è specificato, rendendo <code><area></code> un link, <code><area></code> supporta anche tutti gli attributi del tag <code>anchor</code> (<code><a></code>) tranne <code>ping</code> . Guardali ai documenti MDN .
alt	Il testo alternativo da visualizzare se le immagini non sono supportate. Questo è necessario solo se <code>href</code> è impostato anche su <code><area></code> .
coords	Le coordinate che delineano l'area selezionabile. Quando <code>shape="polygon"</code> , questo dovrebbe essere impostato su un elenco di coppie "x, y" separate da virgole (cioè, <code>shape="polygon" coords="x1, y1, x2, y2, x3, y3, ..."</code>). Quando <code>shape="rectangle"</code> , questo deve essere impostato a <code>left, top, right, bottom</code> . Quando <code>shape="circle"</code> , questo deve essere impostato su <code>centerX, centerY, radius</code> .
href	L'URL del collegamento ipertestuale, se specificato. Se viene omissso, allora <code><area></code> non rappresenterà un collegamento ipertestuale.
shape	La forma di <code><area></code> . Può essere impostato <code>default</code> per selezionare l'intera immagine (non <code>coords</code> attributo necessario), <code>circle</code> o <code>circ</code> per un cerchio, <code>rectangle</code> o <code>rect</code> per un rettangolo e <code>polygon</code> o <code>poly</code> per un'area poligonale

Tag / Attributo	Valore
specificata da punti d'angolo.	

Osservazioni

- L'elenco dei parametri sopra è modificato dai documenti MDN: `<map>` e `<area>` .
- È possibile creare le coordinate di una mappa immagine per un'immagine con forme più semplici (come nell'esempio introduttivo sopra) con un editor di immagini che mostra le coordinate (come GIMP). Tuttavia, potrebbe essere più semplice in generale utilizzare un generatore di mappe di immagini, come [questo](#) .

Examples

Introduzione alle mappe di immagini

Descrizione

Una mappa immagine è un'immagine con aree selezionabili che di solito fungono da collegamenti ipertestuali.

L'immagine è definita dal tag `<img>` e la mappa è definita da un tag `<map>` con tag `<area>` per indicare ciascuna area cliccabile. Utilizza la `usemap` del `usemap` e gli attributi del `name` per associare l'immagine e la mappa.

Esempio di base

Per creare una mappa immagine in modo che ciascuna delle forme nell'immagine sottostante sia selezionabile:



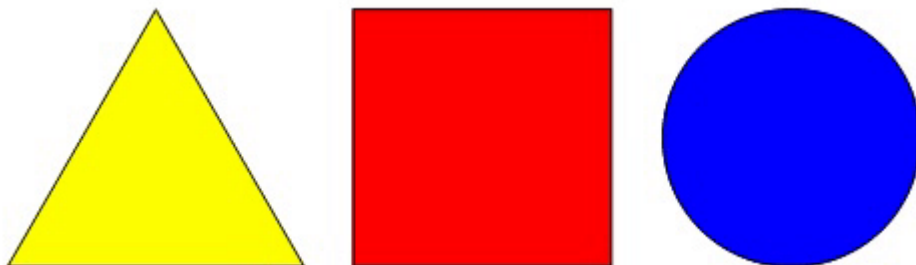
Il codice sarebbe il seguente:

```

<map name="shapes">
```

```
<area shape="polygon" coords="79,6,5,134,153,134">  
<area shape="rectangle" coords="177,6,306,134">  
<area shape="circle" coords="397,71,65">  
</map>
```

Dovresti vedere che il browser riconosce le aree quando il cursore diventa un puntatore. Guarda una [demo dal vivo](#) su JSFiddle o guarda una dimostrazione qui sotto:



Leggi Mappe di immagini online: <https://riptutorial.com/it/html/topic/3819/mappe-di-immagini>

Capitolo 33: Marcatura del codice del computer

Sintassi

- `<pre>Formatted text</pre>`
- `<code>Inline Code</code>`

Osservazioni

L'elemento di `code` dovrebbe essere utilizzato per qualsiasi tipo di "stringa che un computer riconoscerebbe" ([HTML5](#)), ad esempio:

- codice sorgente
- termini da markup / linguaggi di programmazione (nomi di elementi, nomi di funzioni, ecc.)
- nomi di file

Elementi correlati

Per le variabili, è possibile utilizzare l' [elemento](#) `var` .

Per l'uscita del computer, è possibile utilizzare l' [elemento](#) `samp` .

Per l'input dell'utente, è possibile utilizzare l' [elemento](#) `kbd` .

Examples

In linea con

Se una frase contiene codice del computer (ad esempio, il nome di un elemento HTML), utilizzare l'elemento del `code` per contrassegnarlo:

```
<p>The <a> element creates a hyperlink.</p>
```

Blocca con

```
e
```

Se la formattazione (spazio bianco, nuove linee, indentazione) del codice è importante, utilizzare l'elemento `pre` in combinazione con l'elemento `code` :

```
<pre>
  <code>
    x = 42
```

```
if x == 42:
    print "x is ...      ... 42"
</code>
</pre>
```

Devi ancora sfuggire ai caratteri con un significato speciale in HTML (come < con `<`), quindi per *visualizzare* un blocco di codice HTML (`<p>This is a paragraph.</p>`), potrebbe apparire come questo:

```
<pre>
  <code>
    &lt;p>This is a paragraph.&lt;/p>
  </code>
</pre>
```

Leggi [Marcatura del codice del computer](https://riptutorial.com/it/html/topic/1836/marcatura-del-codice-del-computer) online:

<https://riptutorial.com/it/html/topic/1836/marcatura-del-codice-del-computer>

Capitolo 34: Meta informazioni

introduzione

I meta tag nei documenti HTML forniscono informazioni utili sul documento, tra cui una descrizione, parole chiave, autore, date delle modifiche e circa 90 altri campi. Questo argomento riguarda l'utilizzo e lo scopo di questi tag.

Sintassi

- `<meta name="metadata name" content="value">`
- `<meta http-equiv="pragma directive" content="value">`
- `<meta charset="encoding label">`

Osservazioni

Il meta tag è un tag HTML utilizzato per impostare i metadati del documento HTML. I meta tag devono essere inseriti nell'elemento head. Una pagina può avere un numero qualsiasi di meta tag.

Le **keywords** meta tag non sono in genere utilizzate dai robot. La maggior parte dei motori di ricerca determina quali parole chiave si adattano al contenuto delle pagine web. Detto questo, nulla dice che non dovresti più includere il meta tag keywords.

I metadati di una pagina sono utilizzati principalmente dal browser (come il ridimensionamento di un documento) e dagli spider per la scansione del Web utilizzati dai motori di ricerca (Google, Yahoo !, Bing).

La specifica fornisce un numero di [nomi di metadati standardizzati](#) da utilizzare con `<meta name>` e [le direttive standardizzate sui pradma dei metadati](#) da utilizzare con `<meta http-equiv>`. Tuttavia, molti servizi su Internet (web crawler, strumenti di authoring, servizi di condivisione sociale, ecc.) Utilizzano il modulo `<meta name>` come punto di estensione generico per i metadati. Alcuni di questi sono elencati [nella pagina wiki della specifica](#).

Examples

Codifica dei caratteri

L'attributo `charset` specifica la codifica dei caratteri per il documento HTML e deve essere una codifica caratteri valida (gli esempi includono `windows-1252`, `ISO-8859-2`, `Shift_JIS` e `UTF-8`). `UTF-8` (Unicode) è il più usato e dovrebbe essere usato per qualsiasi nuovo progetto.

5

```
<meta charset="UTF-8">
```

```
<meta charset="ISO-8859-1">
```

Tutti i browser hanno sempre riconosciuto il modulo `<meta charset>` , ma se per qualche motivo hai bisogno che la tua pagina sia valida in HTML 4.01, puoi invece utilizzare quanto segue:

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

```
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
```

Vedi anche lo [standard di codifica](#) , per visualizzare tutte le etichette di codifica dei caratteri disponibili che i browser riconoscono.

Aggiornamento automatico

Per aggiornare la pagina ogni cinque secondi, aggiungi questo `meta` elemento nell'elemento `head` :

```
<meta http-equiv="refresh" content="5">
```

ATTENZIONE! Sebbene si tratti di un comando valido, si consiglia di non utilizzarlo a causa dei suoi effetti negativi sull'esperienza utente. Rinfrescare troppo spesso la pagina può far sì che non risponda e spesso scorre verso l'alto della pagina. Se alcune informazioni sulla pagina devono essere aggiornate continuamente, ci sono molti modi migliori per farlo semplicemente aggiornando una parte di una pagina.

Controllo del layout mobile

I siti comuni ottimizzati per dispositivi mobili utilizzano il `<meta name="viewport">` come questo:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

L'elemento `viewport` fornisce al browser le istruzioni su come controllare le dimensioni e il ridimensionamento della pagina in base al dispositivo che si sta utilizzando.

Nell'esempio di cui sopra, `content="width=device-width"` significa che il browser renderà la larghezza della pagina alla larghezza del proprio schermo. Quindi, se questo schermo è `480px wide` , la finestra del browser sarà `480px wide` . `initial-scale=1` indica che lo zoom iniziale (che in questo caso è 1, significa che non esegue lo zoom).

Di seguito sono riportati gli attributi supportati da questo tag:

Attributo	Descrizione
<code>width</code>	La larghezza della finestra virtuale del dispositivo. Valori ¹ : <code>device-width</code> o larghezza effettiva in pixel, come <code>480</code>
<code>height</code>	L'altezza della finestra virtuale del dispositivo. Valori ² : <code>device-height</code> o larghezza effettiva in pixel, come <code>600</code>

Attributo	Descrizione
<code>initial-scale</code>	Lo zoom iniziale quando viene caricata la pagina. 1.0 non ingrandisce.
<code>minimum-scale</code>	L'importo minimo che il visitatore può ingrandire sulla pagina. 1.0 non ingrandisce.
<code>maximum-scale</code>	L'importo massimo che il visitatore può ingrandire sulla pagina. 1.0 non ingrandisce.
<code>user-scalable</code>	Consente al dispositivo di ingrandire e rimpicciolire. I valori sono <code>yes</code> o <code>no</code> . Se impostato su <code>no</code> , l'utente non è in grado di ingrandire la pagina web. L'impostazione predefinita è <code>sì</code> . Le impostazioni del browser possono ignorare questa regola.

Gli appunti:

¹ La proprietà `width` può essere specificata in *pixel* (`width=600`) o in base *alla larghezza del dispositivo* (`width=device-width`) che rappresenta la larghezza fisica dello schermo del dispositivo.

² Analogamente, la proprietà `height` può essere specificata in *pixels* (`height=600`) o in base *device-height* (`height=device-height`) che rappresenta l'altezza fisica dello schermo del dispositivo.

Informazioni sulla pagina

`application-name`

Fornire il nome dell'applicazione Web rappresentata dalla pagina.

```
<meta name="application-name" content="OpenStreetMap">
```

Se non è un'applicazione Web, il meta tag `application-name` non deve essere utilizzato.

`author`

Imposta l'autore della pagina:

```
<meta name="author" content="Your Name">
```

Si può dare un solo nome.

`description`

Imposta la descrizione della pagina:

```
<meta name="description" content="Page Description">
```

Il meta tag `description` può essere utilizzato da vari motori di ricerca durante l'indicizzazione della

pagina Web per scopi di ricerca. Di solito, la descrizione contenuta nel meta tag è il breve riassunto che compare sotto la pagina / il titolo principale del sito Web nei risultati del motore di ricerca. Google di solito utilizza solo le prime 20-25 parole della tua descrizione.

generator

```
<meta name="generator" content="HTML Generator 1.42">
```

Identifica uno dei pacchetti software utilizzati per generare il documento. Da utilizzare solo per le pagine in cui il markup viene generato automaticamente.

keywords

Imposta le parole chiave per i motori di ricerca (separati da virgole):

```
<meta name="keywords" content="Keyword1, Keyword2">
```

Il meta tag `keywords` viene talvolta utilizzato dai motori di ricerca per conoscere la query di ricerca pertinente alla tua pagina web.

Come regola generale, è probabilmente una buona idea non aggiungere troppe parole, poiché la maggior parte dei motori di ricerca che utilizzano questo meta tag per l'indicizzazione indicizzerà solo le prime 20 parole. Assicurati di inserire prima le parole chiave più importanti.

robot

L'attributo `robots`, supportato da diversi principali motori di ricerca, controlla se gli spider dei motori di ricerca sono autorizzati a indicizzare una pagina o meno e se devono seguire o meno i link da una pagina.

```
<meta name="robots" content="noindex">
```

Questo esempio indica a tutti i motori di ricerca di non mostrare la pagina nei risultati di ricerca. Altri valori consentiti sono:

Valore / Direttiva	Senso
all	Predefinito. Equivalente <code>index, follow</code> . Vedi la nota sotto.
noindex	Non indicizzare affatto la pagina.
nofollow	Non seguire i link su questa pagina
follow	I collegamenti sulla pagina possono essere seguiti. Vedi la nota sotto.
none	Equivale a <code>noindex, nofollow</code> .
noarchive	Non rendere disponibile una versione cache di questa pagina nei

Valore / Direttiva	Senso
	risultati di ricerca.
<code>nocache</code>	Sinonimo di <code>noarchive</code> utilizzato da alcuni bot come Bing.
<code>nosnippet</code>	Non mostrare uno snippet di questa pagina nei risultati di ricerca.
<code>noodp</code>	Non utilizzare i metadati di questa pagina dal progetto Open Directory per titoli o snippet nei risultati di ricerca.
<code>notranslate</code>	Non offrire traduzioni di questa pagina nei risultati di ricerca.
<code>noimageindex</code>	Non indicizzare le immagini su questa pagina.
<code>unavailable_after [RFC-850 date/time]</code>	Non mostrare questa pagina nei risultati di ricerca dopo la data / ora specificata. La data / ora devono essere specificate nel formato RFC 850 .

Nota: la definizione esplicita `index` e / o del `follow`, mentre i valori validi, non sono necessari in quanto praticamente tutti i motori di ricerca presumono che siano autorizzati a farlo, se non gli viene impedito esplicitamente di farlo. Simile a come funziona il file `robots.txt`, i motori di ricerca in genere cercano solo cose che *non* sono *autorizzati* a fare. Solo l'affermazione di cose che un motore di ricerca non è autorizzato a fare impedisce anche l'affermazione accidentale di opposti (come `index`, ..., `noindex`) che non tutti i motori di ricerca tratteranno nello stesso modo.

Riconoscimento numero telefonico

Le piattaforme mobili come iOS riconoscono automaticamente i numeri di telefono e li trasformano in `tel:` links. Mentre la funzione è molto pratica, il sistema a volte rileva codici ISBN e altri numeri come numeri di telefono.

Per Safari mobile e alcuni altri browser mobili basati su WebKit per disattivare il riconoscimento e la formattazione automatica del numero di telefono, è necessario questo metatag:

```
<meta name="format-detection" content="telephone=no">
```

Social media

Open Graph è uno standard per i metadati che estende le normali informazioni contenute nel codice headup del sito. Ciò consente a siti Web come Facebook di visualizzare informazioni più approfondite e più complete su un sito Web in un formato strutturato. Queste informazioni vengono quindi visualizzate automaticamente quando gli utenti condividono collegamenti a siti Web contenenti metadati OG su Facebook.

Facebook / Apri grafico

```
<meta property="fb:app_id" content="123456789">
```

```
<meta property="og:url" content="https://example.com/page.html">
<meta property="og:type" content="website">
<meta property="og:title" content="Content Title">
<meta property="og:image" content="https://example.com/image.jpg">
<meta property="og:description" content="Description Here">
<meta property="og:site_name" content="Site Name">
<meta property="og:locale" content="en_US">
<meta property="article:author" content="">
<!-- Facebook: https://developers.facebook.com/docs/sharing/webmasters#markup -->
<!-- Open Graph: http://ogp.me/ -->
```

- [Facebook Open Graph Markup](#)
- [Protocollo Open Graph](#)

Facebook / Articoli istantanei

```
<meta charset="utf-8">
<meta property="op:markup_version" content="v1.0">

<!-- The URL of the web version of your article -->
<link rel="canonical" href="http://example.com/article.html">

<!-- The style to be used for this article -->
<meta property="fb:article_style" content="myarticlestyle">
```

- [Articoli istantanei di Facebook: creazione di articoli](#)
- [Articoli istantanei: riferimento al formato](#)

Twitter utilizza il proprio markup per i metadati. Questi metadati vengono utilizzati come informazioni per controllare come vengono visualizzati i tweet quando contengono un collegamento al sito.

cinguettio

```
<meta name="twitter:card" content="summary">
<meta name="twitter:site" content="@site_account">
<meta name="twitter:creator" content="@individual_account">
<meta name="twitter:url" content="https://example.com/page.html">
<meta name="twitter:title" content="Content Title">
<meta name="twitter:description" content="Content description less than 200 characters">
<meta name="twitter:image" content="https://example.com/image.jpg">
```

- [Twitter Cards: Guida introduttiva](#)
- [Twitter Card Validator](#)

Google+ / Schema.org

```
<link href="https://plus.google.com/+YourPage" rel="publisher">
<meta itemprop="name" content="Content Title">
<meta itemprop="description" content="Content description less than 200 characters">
<meta itemprop="image" content="https://example.com/image.jpg">
```

Reindirizzamento automatico

A volte la tua pagina web ha bisogno di un reindirizzamento automatico.

Ad esempio, per reindirizzare a `example.com` dopo 5 secondi:

```
<meta http-equiv="refresh" content="5;url=https://www.example.com/" />
```

Questa linea ti invierà al sito web designato (in questo `example.com` dopo 5 secondi).

Se è necessario modificare il ritardo prima di un reindirizzamento, è sufficiente modificare il numero subito prima del tuo `;url=` modificherà il tempo di ritardo.

App Web

Puoi configurare la tua app web o il tuo sito web per aggiungere un'icona di collegamento dell'applicazione alla schermata iniziale del dispositivo e avviare l'app in modalità "app" a schermo intero utilizzando la voce di menu **"Aggiungi a homescreen"** di Chrome per Android.

Sotto i meta tag si aprirà l'app Web in modalità a schermo intero (senza barra degli indirizzi).

Android Chrome

```
<meta name="mobile-web-app-capable" content="yes">
```

IOS

```
<meta name="apple-mobile-web-app-capable" content="yes">
```

Puoi anche impostare il colore per la barra di stato e la barra degli indirizzi nel meta tag.

Android Chrome

```
<meta name="theme-color" content="black">
```

IOS

```
<meta name="apple-mobile-web-app-status-bar-style" content="black">
```

Leggi Meta informazioni online: <https://riptutorial.com/it/html/topic/1264/meta-informazioni>

Capitolo 35: Quotazioni di marcatura

Osservazioni

`cite` e `blockquote` elementi non dovrebbero essere usati allo scopo di rappresentare una conversazione, trascrizioni di conversazioni, dialoghi in script, registrazioni di messaggi istantanei e altre situazioni in cui i diversi attori si alternano nel discorso.

Examples

In linea con

L' **elemento** `q` può essere utilizzato per una citazione che fa parte di una frase:

```
<p>She wrote <q>The answer is 42.</q> and everyone agreed.</p>
```

Virgolette

4.01

Le virgolette non dovrebbero essere aggiunte. I programmi utente dovrebbero (in HTML 4.01) risp. deve (in HTML 4.0) renderli automaticamente.

5

Le virgolette non devono essere aggiunte. I programmi utente li renderanno automaticamente.

URL di origine (attributo di `cite`)

L' **attributo** `cite` può essere utilizzato per fare riferimento all'URL della fonte citata:

```
<p>She wrote <q cite="http://example.com/blog/hello-world">The answer is 42.</q> and everyone agreed.</p>
```

Nota che i browser in genere non mostrano questo URL, quindi se la fonte è pertinente, dovresti aggiungere anche un collegamento ipertestuale (`a` elemento).

Blocca con

L' **elemento** `blockquote` può essere utilizzato per una citazione (a livello di blocco):

```
<blockquote>
  <p>The answer is 42.</p>
</blockquote>
```

URL di origine (attributo di `cite`)

L' **attributo** `cite` può essere utilizzato per fare riferimento all'URL della fonte citata:

```
<blockquote cite="http://example.com/blog/hello-world">
  <p>The answer is 42.</p>
</blockquote>
```

Nota che i browser in genere non mostrano questo URL, quindi se la fonte è pertinente, dovresti aggiungere un collegamento ipertestuale (`a` elemento) in aggiunta (consulta la sezione *Citazione / Attribuzione* su dove collocare questo link).

Citation / Attribuzione

4.01

La citazione / attribuzione non dovrebbe essere parte dell'elemento `blockquote` :

```
<blockquote cite="http://example.com/blog/hello-world">
  <p>The answer is 42.</p>
</blockquote>
<p>Source: <cite><a href="http://example.com/blog/hello-world" rel="external">Hello
World</a></cite></p>
```

È possibile aggiungere un elemento `div` per raggruppare la citazione e la citazione, ma non esiste un modo per associarli semanticamente.

L' **elemento** `cite` può essere utilizzato come riferimento della fonte citata (ma non per il nome dell'autore).

5

La citazione / attribuzione (ad esempio, il collegamento ipertestuale che fornisce l'URL di origine) può essere all'interno del `blockquote` , ma in tal caso deve essere all'interno di un elemento `cite` (per attribuzioni nel testo) o di un elemento del `footer` :

```
<blockquote cite="http://example.com/blog/hello-world">
  <p>The answer is 42.</p>
  <footer>
    <p>Source: <cite><a href="http://example.com/blog/hello-world" rel="external">Hello
World</a></cite></p>
  </footer>
</blockquote>
```

L' **elemento** `cite` può essere utilizzato come riferimento della fonte citata o per il nome dell'autore della citazione.

Leggi Quotazioni di marcatura online: <https://riptutorial.com/it/html/topic/2943/quotazioni-di-marcatura>

Capitolo 36: SVG

introduzione

SVG è l'acronimo di Scalable Vector Graphics. SVG è usato per definire la grafica per il Web

L'elemento HTML `<svg>` è un contenitore per la grafica SVG.

SVG ha diversi metodi per disegnare percorsi, scatole, cerchi, testo e immagini grafiche.

Osservazioni

SVG è un linguaggio basato su XML per la creazione di immagini vettoriali scalabili. Può essere scritto direttamente in un documento HTML o incorporato da file SVG esterni. Inline SVG può essere ridisegnato e modificato utilizzando rispettivamente CSS e JavaScript.

Il supporto del browser per SVG varia, ma può essere accertato [qui](#).

Per informazioni più complete, consultare la [documentazione SVG](#).

Examples

Incorporamento di file SVG esterni in HTML

È possibile utilizzare gli elementi `<img>` o `<object>` per incorporare elementi SVG esterni. Impostare l'altezza e la larghezza è facoltativo ma è altamente raccomandato.

Utilizzando l'elemento immagine

```

```

L'uso di `<img>` non ti consente di definire lo SVG usando CSS o manipolarlo usando JavaScript.

Utilizzando l'elemento dell'oggetto

```
<object type="image/svg+xml" data="attention.svg" width="50" height="50">
```

A differenza di `<img>`, `<object>` importa direttamente l'SVG nel documento e quindi può essere manipolato usando Javascript e CSS.

Inline SVG

SVG può essere scritto direttamente in un documento HTML. Inline SVG può essere disegnato e manipolato usando CSS e JavaScript.


```
<body>
  <svg class="attention" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 0 1000 1000" >
    <path id="attention" d="m571,767l0,-106q0,-8,-5,-13t-12,-5l-108,0q-7,0,-12,5t-
5,13l0,106q0,8,5,13t12,6l108,0q7,0,12,-6t5,-13zm-1,-208l10,-257q0,-6,-5,-10q-7,-6,-14,-6l-
122,0q-7,0,-14,6q-5,4,-5,12l9,255q0,5,6,9t13,3l103,0q8,0,13,-3t6,-9zm-7,-522l428,786q20,35,-
1,70q-10,17,-26,26t-35,10l-858,0q-18,0,-35,-10t-26,-26q-21,-35,-1,-70l429,-786q9,-17,26,-
27t36,-10t36,10t27,27z" />
  </svg>
</body>
```

L'SVG inline sopra può quindi essere stilizzato usando la corrispondente classe CSS:

```
.attention {
  fill: red;
  width: 50px;
  height: 50px;
}
```

Il risultato è simile a questo:



Incorporare SVG usando CSS

Puoi aggiungere file SVG esterni usando la proprietà `background-image`, proprio come faresti con qualsiasi altra immagine.

HTML:

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

CSS:

```
.attention {
  background-image: url(attention.svg);
  background-size: 100% 100%;
  width: 50px;
  height: 50px;
}
```

Puoi anche incorporare l'immagine direttamente in un file css utilizzando un url di dati:

```
background-image:
url(data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsvg%22%20xmlns%3Axlink%3D%
106q0%2C-8%2C-5%2C-13t-12%2C-5l-108%2C0q-7%2C0%2C-12%2C5t-
5%2C13l0%2C106q0%2C8%2C5%2C13t12%2C6l108%2C0q7%2C0%2C12%2C-6t5%2C-13zm-1%2C-208l10%2C-
257q0%2C-6%2C-5%2C-10q-7%2C-6%2C-14%2C-6l-122%2C0q-7%2C0%2C-14%2C6q-5%2C4%2C-
5%2C12l9%2C255q0%2C5%2C6%2C9t13%2C3l103%2C0q8%2C0%2C13%2C-3t6%2C-9zm-7%2C-
522l428%2C786q20%2C35%2C-1%2C70q-10%2C17%2C-26%2C26t-35%2C10l-858%2C0q-18%2C0%2C-35%2C-10t-
26%2C-26q-21%2C-35%2C-1%2C-70l429%2C-786q9%2C-17%2C26%2C-27t36%2C-
```

10t36%2C10t27%2C27Z%22%20%2F%3E%0D%0A%3C%2Fsvg%3E) ;

Leggi SVG online: <https://riptutorial.com/it/html/topic/1183/svg>

Capitolo 37: tabelle

introduzione

L'elemento `<table>` HTML consente agli autori Web di visualizzare dati tabulari (come testo, immagini, collegamenti, altre tabelle, ecc.) In una tabella bidimensionale con righe e colonne di celle.

Sintassi

- `<table></table>`
- `<thead></thead>`
- `<tbody></tbody>`
- `<tfoot></tfoot>`
- `<tr></tr>`
- `<th></th>`
- `<td></td>`

Osservazioni

I vari elementi della tabella e i relativi attributi del contenuto definiscono insieme il modello della tabella. L'elemento `<table>` è l'elemento contenitore per i modelli tabella / dati tabulari. Le tabelle hanno righe, colonne e celle dati dai loro discendenti. Le righe e le colonne formano una griglia; le celle di un tavolo devono coprire completamente quella griglia senza sovrapposizioni. L'elenco seguente descrive i vari elementi nel modello di tabella:

- `<table>` - L'elemento contenitore per i modelli tabella / dati tabulari. `<table>` rappresenta i dati con più di una dimensione sotto forma di tabella.
- `<caption>` - Didascalia o titolo della tabella (come una `figcaption` in `figure`)
- `<col>` - Una colonna (un elemento senza contenuto)
- `<colgroup>` - Un raggruppamento di colonne
- `<thead>` - Intestazione tabella (solo una)
- `<tbody>` - Corpo della tabella / contenuto (più sono ok)
- `<tfoot>` - Table footer (solo uno)
- `<tr>` - Riga della tabella
- `<th>` - Cella di intestazione della tabella
- `<td>` - Cella dati della tabella

Semanticamente, le tabelle sono pensate per contenere dati tabulari. Puoi considerarlo come un modo per visualizzare e descrivere i dati che avrebbero senso in un foglio di calcolo (colonne e righe).

L'uso di tabelle per il layout non è raccomandato. Invece, usa le regole CSS per il layout e la formattazione, tra cui `display: table` .

Un'eccezione degna di nota visualizzata in genere nell'industria per quanto riguarda l'utilizzo del

layout `<table>` riguarda l'e-mail HTML: alcuni client di posta elettronica, incluso Outlook, tornano ai vecchi motori di rendering dopo che Microsoft ha perso il proprio caso di monopolio contro l'UE. Per fare in modo che Microsoft non faccia parte del sistema operativo, IE ha semplicemente ripristinato il motore di rendering di Outlook con una versione precedente di Trident. Questo rollback semplicemente non supporta le moderne tecnologie web, quindi l'utilizzo dei layout basati su `<table>` per le email HTML è l'unico modo per garantire la compatibilità cross-browser / piattaforma / client.

Examples

Tavolo semplice

```
<table>
  <tr>
    <th>Heading 1/Column 1</th>
    <th>Heading 2/Column 2</th>
  </tr>
  <tr>
    <td>Row 1 Data Column 1</td>
    <td>Row 1 Data Column 2</td>
  </tr>
  <tr>
    <td>Row 2 Data Column 1</td>
    <td>Row 2 Data Column 2</td>
  </tr>
</table>
```

Ciò renderà una `<table>` composta da tre righe totali (`<tr>`): una riga di celle di intestazione (`<th>`) e due righe di celle di contenuto (`<td>`). `<th>` elementi sono *intestazioni tabulari* e `<td>` elementi sono *dati tabulari* . Puoi mettere tutto ciò che vuoi all'interno di `<td>` o `<th>` .

Titolo 1 / colonna 1	Titolo 2 / colonna 2
Riga 1 Colonna dati 1	Colonna dati riga 1 2
Riga 2 Colonna dati 1	Riga 2 Colonna dati 2

Spanning colonne o righe

Le celle di tabella possono estendersi su più colonne o righe utilizzando gli attributi `colspan` e `rowspan` . Questi attributi possono essere applicati agli elementi `<th>` e `<td>` .

```
<table>
  <tr>
    <td>row 1 col 1</td>
    <td>row 1 col 2</td>
    <td>row 1 col 3</td>
  </tr>
  <tr>
    <td colspan="3">This second row spans all three columns</td>
  </tr>
```

```

<tr>
  <td rowspan="2">This cell spans two rows</td>
  <td>row 3 col 2</td>
  <td>row 3 col 3</td>
</tr>
<tr>
  <td>row 4 col 2</td>
  <td>row 4 col 3</td>
</tr>
</table>

```

Risulterà in

row 1 col 1	row 1 col 2	row 1 col 3
This second row spans all three columns		
This cell spans two rows	row 3 col 2	row 3 col 3
	row 4 col 2	row 4 col 3

Tieni presente che non dovresti progettare una tabella in cui sia le righe che le colonne si sovrappongono in quanto HTML non valido e il risultato viene gestito in modo diverso da browser Web diversi.

`rowspan` = Un numero intero non negativo che specifica il numero di righe occupate da una cella. Il valore predefinito di questo attributo è uno (1). Un valore pari a zero (0) significa che la cella si estenderà dalla riga corrente fino all'ultima riga della tabella (<thead> , <tbody> 0 <tfoot>).

`colspan` = Un numero intero non negativo che specifica il numero di colonne espanse dalla cella corrente. Il valore predefinito di questo attributo è uno (1). Un valore pari a zero (0) significa che la cella si estenderà dalla corrente all'ultima colonna del gruppo di colonne <colgroup> in cui è definita la cella.

Tavolo con `thead`, `tbody`, `tfoot` e `caption`

HTML fornisce anche le tabelle con gli <thead> , <tbody> , <tfoot> e <caption> . Questi elementi aggiuntivi sono utili per aggiungere valore semantico alle tabelle e per fornire un posto per lo stile CSS separato.

Quando stampi un tavolo che non si adatta a una pagina (di carta), la maggior parte dei browser ripete il contenuto di <thead> su ogni pagina.

C'è un ordine specifico che deve essere rispettato e dovremmo essere consapevoli del fatto che non tutti gli elementi vanno a posto come ci si aspetterebbe. L'esempio seguente dimostra come devono essere posizionati i nostri 4 elementi.

```

<table>
  <caption>Table Title</caption> <!--| caption is the first child of table |-->
  <thead> <!--=====| head is after caption |-->
    <tr>
      <th>Header content 1</th>

```

```

        <th>Header content 2</th>
    </tr>
</thead>

<tbody> <!--=====| tbody is after thead |-->
    <tr>
        <td>Body content 1</td>
        <td>Body content 2</td>
    </tr>
</tbody>

<tfoot><!--| tfoot can be placed before or after tbody, but not in a group of tbody. |-->
<!--| Regardless where tfoot is in markup, it's rendered at the bottom. |-->

    <tr>
        <td>Footer content 1</td>
        <td>Footer content 2</td>
    </tr>
</tfoot>

</table>

```

I seguenti esempi di risultati sono dimostrati due volte: la prima tabella non ha stili, la seconda tabella ha alcune proprietà CSS applicate: `background-color` , `color` e `border` *. Gli stili sono forniti come guida visiva e non sono un aspetto essenziale dell'argomento in questione.

Table Title	
Header content 1	Header content 2
Body content 1	Body content 2
Footer content 1	Footer content 2

Table Title	
Header content 1	Header content 2
Body content 1	Body content 2
Footer content 1	Footer content 2

Elemento	Gli stili si applicano
<caption>	Testo giallo su sfondo nero.
<thead>	Testo in grassetto su sfondo viola.
<tbody>	Testo su sfondo blu
<tfoot>	Testo su sfondo verde
<th>	Bordi arancioni
<td>	Bordi rossi

Gruppi di colonne

A volte potresti voler applicare lo stile a una colonna o a un gruppo di colonne. O per scopi semantici, potresti voler raggruppare le colonne. Per fare ciò, usa gli elementi `<colgroup>` e `<col>`.

Il tag `<colgroup>` facoltativo consente di raggruppare le colonne. `<colgroup>` elementi `<colgroup>` devono essere elementi figli di `<table>` e devono venire dopo ogni elemento `<caption>` e prima di qualsiasi contenuto di tabella (ad es. `<tr>`, `<thead>`, `<tbody>`, ecc.).

```
<table>
  <colgroup span="2"></colgroup>
  <colgroup span="2"></colgroup>
  ...
</table>
```

Il tag `<col>` facoltativo consente di fare riferimento a singole colonne o a un intervallo di colonne senza applicare un raggruppamento logico. `<col>` elementi `<col>` sono facoltativi, ma se presenti, devono essere all'interno di un elemento `<colgroup>`.

```
<table>
  <colgroup>
    <col id="MySpecialColumn" />
    <col />
  </colgroup>
  <colgroup>
    <col class="CoolColumn" />
    <col class="NeatColumn" span="2" />
  </colgroup>
  ...
</table>
```

I seguenti stili CSS possono essere applicati agli elementi `<colgroup>` e `<col>`:

- `border`
- `background`
- `width`
- `visibility`
- `display` (come in `display: none`)
 - `display: none`; rimuoverà effettivamente le colonne dal display, facendo sì che la tabella esegua il rendering come se tali celle non esistessero

Per ulteriori informazioni, vedere [Dati tabulari HTML5](#).

Ambito di titolo

`th` elementi `th` sono usati molto comunemente per indicare le intestazioni di righe e colonne di tabelle, in questo modo:

```

<table>
  <thead>
    <tr>
      <td></td>
      <th>Column Heading 1</th>
      <th>Column Heading 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>Row Heading 1</th>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <th>Row Heading 2</th>
      <td></td>
      <td></td>
    </tr>
  </tbody>
</table>

```

Questo può essere migliorato per l'accessibilità mediante l'uso dell'attributo `scope`. L'esempio precedente sarà modificato come segue:

```

<table>
  <thead>
    <tr>
      <td></td>
      <th scope="col">Column Heading 1</th>
      <th scope="col">Column Heading 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">Row Heading 1</th>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <th scope="row">Row Heading 1</th>
      <td></td>
      <td></td>
    </tr>
  </tbody>
</table>

```

`scope` è conosciuto come un *attributo enumerato*, nel senso che può avere un valore da un insieme specifico di valori possibili. Questo set include:

- `col`
- `row`
- `colgroup`
- `rowgroup`

Riferimenti:

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/th#attr-scope>

- <https://www.w3.org/TR/WCAG20-TECHS/H63.html>

Leggi tabelle online: <https://riptutorial.com/it/html/topic/274/tabelle>

Capitolo 38: tabindex

Parametri

Valore	Senso
negativo	l'elemento sarà focalizzabile, ma non dovrebbe essere raggiungibile tramite la navigazione sequenziale della tastiera
0	l'elemento sarà focalizzabile e raggiungibile tramite la navigazione sequenziale della tastiera, ma l'ordine relativo è definito dalla convenzione della piattaforma
positivo	l'elemento deve essere focalizzabile e accessibile tramite la navigazione sequenziale della tastiera; il suo ordine relativo sarà definito dal valore dell'attributo: il sequenziale segue il numero crescente del <code>tabindex</code>

Osservazioni

Il valore massimo per `tabindex` non deve superare 32767 come da sezione W1C 17.11.1 A meno che il valore predefinito specificato sia -1

Un elemento con valore 0, un valore non valido o nessun valore `tabindex` deve essere posizionato dopo gli elementi con un indice positivo nell'ordine sequenziale della navigazione da tastiera.

Examples

Aggiungi un elemento all'ordine di tabulazione

```
<div tabindex="0">Some button</div>
```

Nota: Provare a utilizzare un codice HTML nativo `button` o un `a` tag, se del caso.

Rimuovi un elemento dall'ordine di tabulazione

```
<button tabindex="-1">This button will not be reachable by tab</button>
```

L'elemento verrà rimosso dall'ordine di tabulazione ma sarà ancora selezionabile.

Definire un ordine di tabulazione personalizzato (non consigliato)

```
<div tabindex="2">Second</div>
<div tabindex="1">First</div>
```

I valori positivi inseriranno l'elemento nella posizione dell'ordine di tabulazione del rispettivo valore. Gli elementi senza preferenze (es. `tabindex="0"` o elementi nativi come `button` e `a`) saranno aggiunti dopo quelli con preferenza.

I valori positivi **non** sono **raccomandati** in quanto interrompono il comportamento previsto del tabbing e potrebbero confondere le persone che si affidano agli screen reader. Prova a creare un ordine naturale riorganizzando la tua struttura DOM.

Leggi `tabindex` online: <https://riptutorial.com/it/html/topic/2594/tabindex>

Capitolo 39: Tela

Parametri

Attributo	Descrizione
altezza	Specifica l'altezza della tela
larghezza	Specifica la larghezza della tela

Osservazioni

- Questo tag non è compatibile con le versioni di Internet Explorer meno di 9. Controlla caniuse.com per la compatibilità del browser.
- `canvas` è solo un contenitore per la grafica e il disegno reale della grafica è fatto da JavaScript.

Examples

Esempio di base

L'elemento `canvas` stato introdotto in HTML5 per disegnare la grafica.

```
<canvas id="myCanvas">
  Cannot display graphic. Canvas is not supported by your browser (IE<9)
</canvas>
```

Quanto sopra creerà un elemento `<canvas>` HTML trasparente di dimensioni 300 × 150 px.

Puoi usare l'elemento **canvas** per disegnare cose incredibili come forme, grafici, manipolare immagini, creare giochi coinvolgenti ecc. Con **JavaScript**.

L'oggetto superficie del `CanvasRenderingContext2D` *disegnabile* 2D della `canvas` è denominato `CanvasRenderingContext2D`; o da un `HTMLCanvasElement` usando il `.getContext("2d")`:

```
var ctx = document.getElementById("myCanvas").getContext("2d");
// now we can refer to the canvas's 2D layer context using `ctx`

ctx.fillStyle = "#f00";
ctx.fillRect(0, 0, ctx.canvas.width, ctx.canvas.height); // x, y, width, height

ctx.fillStyle = "#000";
ctx.fillText("My red canvas with some black text", 24, 32); // text, x, y
```

[esempio jsFiddle](#)

Disegnare due rettangoli su a

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Draw two rectangles on the canvas</title>
  <style>
    canvas{
      border:1px solid gray;
    }
  </style>
  <script async>
    window.onload = init; // call init() once the window is completely loaded
    function init(){
      // #1 - get reference to <canvas> element
      var canvas = document.querySelector('canvas');

      // #2 - get reference to the drawing context and drawing API
      var ctx = canvas.getContext('2d');

      // #3 - all fill operations are now in red
      ctx.fillStyle = 'red';

      // #4 - fill a 100x100 rectangle at x=0,y=0
      ctx.fillRect(0,0,100,100);

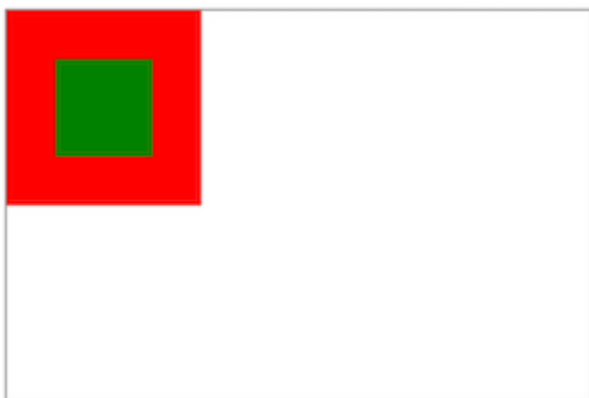
      // #5 - all fill operations are now in green
      ctx.fillStyle = 'green';

      // #6 - fill a 50x50 rectangle at x=25,y=25
      ctx.fillRect(25,25,50,50);

    }
  </script>
</head>
<body>
  <canvas width=300 height=200>Your browser does not support canvas.</canvas>
</body>
</html>

```

Questo esempio ha il seguente aspetto:



Leggi Tela online: <https://riptutorial.com/it/html/topic/1162/tela>

Capitolo 40: Titoli

introduzione

HTML fornisce non solo tag di paragrafo semplici, ma sei tag di intestazione separati per indicare intestazioni di varie dimensioni e spessori. Enumerato come da titolo 1 a titolo 6, l'intestazione 1 presenta il testo più grande e più spesso mentre l'intestazione 6 è il più piccolo e il più sottile, fino al livello di paragrafo. Questo argomento descrive il corretto utilizzo di questi tag.

Sintassi

- `<h1>...</h1>`
- `<h2>...</h2>`
- `<h3>...</h3>`
- `<h4>...</h4>`
- `<h5>...</h5>`
- `<h6>...</h6>`

Osservazioni

- Un elemento `h1` - `h6` deve avere sia un tag di inizio che un tag di fine. ¹
- `h1` elementi `h1` - `h6` sono elementi a livello di blocco per impostazione predefinita (stile CSS: `display: block`). ²
- `h1` elementi `h1` - `h6` non devono essere confusi con l' [elemento section](#)
- I tag di intestazione (`h1` - `h6`) non sono correlati al tag `head` .
- Contenuto consentito: [contenuto di frasi](#)
- I diversi stili CSS per le intestazioni differiscono solitamente in `font-size` e `margin` . Le seguenti impostazioni CSS per gli elementi `h1` - `h6` possono servire come orientamento (caratterizzato dal 'informativo' dal [W3C](#))
- Gli spider dei motori di ricerca (il codice che aggiunge una pagina a un motore di ricerca) presta automaticamente maggiore attenzione a una maggiore importanza (`h1` ha più, `h2` ha meno, `h3` ha ancora meno, ...) titoli per capire di cosa tratta una pagina.

Examples

Usando le intestazioni

Le intestazioni possono essere utilizzate per descrivere l'argomento che precedono e sono definite con i `<h1>` a `<h6>` . Le intestazioni supportano tutti gli [attributi globali](#) .

- `<h1>` definisce l'intestazione più importante.
- `<h6>` definisce l'intestazione meno importante.

Definire un'intestazione:

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

La struttura corretta conta

I **motori di ricerca** e altri programmi **utente di** solito indicizzano il contenuto della pagina in base agli elementi dell'intestazione, ad esempio per creare un sommario, quindi è importante utilizzare la struttura corretta per le intestazioni.

In generale, un articolo dovrebbe avere un elemento `h1` per il titolo principale seguito dai sottotitoli `h2` - scendendo di un livello se necessario. Se ci sono elementi `h1` su un livello più alto, essi non possono essere usati per descrivere alcun contenuto di livello inferiore.

Documento di esempio (intendimento extra per illustrare la gerarchia):

```
<h1>Main title</h1>
<p>Introduction</p>

  <h2>Reasons</h2>

    <h3>Reason 1</h3>
    <p>Paragraph</p>

    <h3>Reason 2</h3>
    <p>Paragraph</p>

  <h2>In conclusion</h2>
  <p>Paragraph</p>
```

Leggi Titoli online: <https://riptutorial.com/it/html/topic/226/titoli>

Capitolo 41: Usare l'HTML con i CSS

introduzione

Il **CSS** fornisce gli stili agli elementi HTML sulla pagina. Lo stile in linea comporta l'utilizzo dell'attributo `style` nei tag ed è altamente sconsigliato. I fogli di stile interni usano il tag `<style>` e sono usati per dichiarare le regole per le porzioni dirette della pagina. I fogli di stile esterni possono essere utilizzati tramite un tag `<link>` che accetta un file esterno di CSS e applica le regole al documento. Questo argomento riguarda l'utilizzo di tutti e tre i metodi di collegamento.

Sintassi

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

Examples

Uso del foglio di stile esterno

Usa l'attributo `link` nella `head` del documento:

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

È inoltre possibile utilizzare i fogli di stile forniti dai siti Web tramite una rete di distribuzione dei contenuti o in breve CDN. (ad esempio, Bootstrap):

```
<head>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiisSIFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
</head>
```

In generale, è possibile trovare il supporto CDN per un framework sul proprio sito Web.

Foglio di stile interno

Puoi anche includere elementi CSS internamente usando il tag `<style>` :

```
<head>
  <style type="text/css">
    body {
      background-color: gray;
    }
  </style>
```



```
</head>
```

È possibile includere più fogli di stile interni in un programma.

```
<head>
  <style type="text/css">
    body {
      background-color: gray;
    }
  </style>

  <style type="text/css">
    p {
      background-color: blue;
    }
  </style>
</head>
```

Stile in linea

Puoi dare uno stile a un elemento specifico usando l'attributo `style` :

```
<span style="color: red">This text will appear in red.</span>
```

Nota: cerca di evitare questo: il punto del CSS è separare il contenuto dalla presentazione.

Fogli di stile multipli

È possibile caricare più fogli di stile:

```
<head>
  <link rel="stylesheet" type="text/css" href="general.css">
  <link rel="stylesheet" type="text/css" href="specific.css">
</head>
```

Nota che i **file e le dichiarazioni successivi sovrascriveranno quelli precedenti** . Quindi se `general.css` contiene:

```
body {
  background-color: red;
}
```

e `specific.css` contiene:

```
body {
  background-color: blue;
}
```

se entrambi vengono utilizzati, lo sfondo del documento sarà blu.

Leggi Usare l'HTML con i CSS online: <https://riptutorial.com/it/html/topic/4536/usare-l-html-con-i-css>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con HTML	4444 , Abhishek Pandey , aea2002 , ahmednawazbutt , Alexander Wigmore , Alexandre N. , Amanda Ahn , amflare , Amitay Stern , animuson , Anthony Pham , Boris , bwegs , Callan Heard , ChrisD , CocoaBean , Community , Dave Everitt , Dinidu , dippas , dtyler , duskwuff , Eric Dobbs , Firix , FlyingPiMonster , geek1011 , George Bailey , Gerold Broser , H Mirza , H. Pauwelyn , Harish Gyanani , Hemant Kumar , hillary.fraley , Hudson Taylor , ihavemorealts , intboolstring , Isak Combrinck , Jeffrey Lin , JHS , jmmarco , joe_young , John Slegers , Jon Chan , JonasCz , JPB , kelvinelove , Krii , Kurniawantaari , Lahiru Ashan , Lambda Ninja , Léo Martin , Leonidas Menendez , Malcolm , Matt , Matt , MC93 , Michael Moriarty , mnoronha , Muntasir , Nishchay , Ortomala Lokni , Persijn , Prateek , Pyloid , Ryan Hilbert , Shannon Young , sideshowbarker , stark , Stelian Matei , Sunny R Gupta , the12 , tmg , unor , user3130333 , Valor Naram , Willi , Wolfgang , Zaz , zygimantus , Zze
2	Ancore e collegamenti ipertestuali	Al.G. , animuson , Anselm Urban , Anthony Pham , ban17 , DawnPaladin , Emil , FlyingPiMonster , insertusernamehere , J F , JHS , joe_young , Jojodmo , Jones Joseph , Lambda Ninja , Matas Vaitkevicius , Nathan Tuggy , Pranav , Prateek , Raystafarian , Robert Columbia , Squidward , Steyn van Esveld , Thomas Gerot , unor , Wolfgang
3	ARIA	Bhavya Singh , Paul Sweatte , Shannon Young , Travis , unor , user30796
4	Attributi dei dati	animuson , Community , Faegy , Infuzed guy , James Donnelly , Manish , Nathan Tuggy , Nhan , Racil Hilan , rajarshig , swatchai , unor , Yasir T
5	Attributi evento HTML	Paresh Maghodiya
6	Attributi globali	animuson , Becca , unor , Zange-chan
7	Barre di navigazione	Community
8	Classi e ID	Angelos Chalaris , animuson , brandaemon , Caleb Kleveter , Community , Duh-Wayne-101 , Emil , Epodax , Evan , GoatsWearHats , Ingrid Stevens , jhnance , JHS , John Slegers ,

		lexith , Luca Putzu , Michael_B , Natalie , Nhan , Richard Hamilton , Simone Carletti , Thomas Gerot , Timothy , Tyler Zika , unor , Wolfgang , xims
9	Collegamento di risorse	AA2992 , animuson , Anselm Urban , Aravind Suresh , Callan Heard , Chris Rutherford , cone56 , DawnPaladin , Domenic , feeela , Henrique Barcelos , Infuzed guy , JHS , Lambda Ninja , Matas Vaitkevicius , Nhan , Thomas Gerot , unor , V4karian , vladdobra
10	Commenti	Ani Menon , animuson , Ashwin Ramaswami , bdkopen , ChrisD , Epodax , JHS , jkdev , RamenChef , Robert Grant , Soaring Code , Squazz , Thomas Gerot , Ulrich Schwarz , Wolfgang
11	Controlli del menu di selezione	Ali Almoullim , amflare , animuson , GentlePurpleRain , Ilyas karim , Mosh Feu , Tot Zam
12	doctype	Akshay Anand , Al.G. , Angelos Chalaris , Ani Menon , animuson , Chris , Content Solutions , heerfk , mnoronha , pinjasaur , Right leg , Sumner Evans , Thomas Gerot , tmg , unor
13	Elementi di sezionamento	Andrew Brooke , Anil , animuson , Hanif Formoly , nalply , Shannon Young , SuperBiasedMan , SuperStormer , Yossi Aharon
14	Elementi multimediali	feeela , Isak Combrinck , LisaMM , Shiva , Yossi Aharon
15	Elementi vuoti	4444 , ChrisD , Thomas Gerot , unor
16	Elemento di progresso	animuson , Content Solutions , Richard Hamilton
17	Elemento di uscita	J F , Stephen Leppik , zer00ne
18	Elemento div	animuson , Araknid , Black Mamba , Chris , Content Solutions , D M , Faust , feeela , Gal Ratzkin , JHS , MySpeed , S.L. Barth , SuperStormer , Thomas Gerot
19	Elemento etichetta	Anthony Pham , fredden , Josiah Keller , m_callens , Roko C. Buljan
20	elenchi	animuson , BiscuitBaker , Daniel Käfer , Grace Note , H. Pauwelyn , Jon Ericson , kcpike , Marvin , Matas Vaitkevicius , platy11 , Prateek , Pseudonym Patel , Richard Hamilton , Right leg , Sayakiss , Stewartside , Thomas Gerot , tmg , Tom Johnson , unor , Zack
21	Entità personaggio	animuson , MervS , stack-learner
22	Formattazione del testo	animuson , Ben Rhys-Lewis , Emil , gustavohenke , J F , Matas Vaitkevicius , Peter L. , Raystafarian , Stephen Leppik , Thomas

		Gerot, unor, Wolfgang
23	HTML 5 Cache	Farhad, TricksfortheWeb, Valor Naram
24	I paragrafi	Abrar Jahin, Thomas Gerot, Valor Naram
25	IFrame	Adjit, Alexandre N., animuson, ChrisD, dorukayhan, Duh-Wayne-101, Emanuel Vintilă, J F, Ojen, Wojciech Kazior
26	immagini	Alex, Alexandre N., andreaem, animuson, Boysenb3rry, Caleb Kleveter, Gabriel Chi Hong Lee, Infuzed guy, ivn, Kake_Fisk, Maximillian Laumeister, Mohd Samir Khan, Mr Lister, Nhan, Shivangi Chaurasia, Stephen Leppik, Wolfgang
27	Includi codice JavaScript in HTML	Alexandre N., andreaem, animuson, Anselm Urban, Charles, Marjorie Pickard, MervS, Roko C. Buljan, Sildoreth, SuperStormer
28	Incorporare	Alexandre N.
29	Input Control Elements	Abhishek Pandey, Al.G., Alohci, amflare, Amitay Stern, Angelos Chalaris, Ani Menon, animuson, bhansa, Bob, Charlie H, Christophe Strobbe, CN, Community, cone56, Daniel, DawnPaladin, Dipen Shah, Domenic, Druzion, Edwin Tenovimas, Epodax, Franck Dernoncourt, gabe3886, geeksal, H. Pauwelyn, Henrique Barcelos, Huy Nguyen, J F, John Slegers, Kashyap Jha, Lahiru Ashan, Lankymart, Magisch, Marvin, Matas Vaitkevicius, Matt, Maximillian Laumeister, Mike McCaughan, morewry, Mosh Feu, Nathan Arthur, Nil Llisterri, Nishchay, niyasc, NoobCoder, Optimiser, Ortomala Lokni, Pi Programs, Pimgd, Prateek, Prav, Praveen Kumar, Psaniko, QoP, RamenChef, Ranjit Singh, Richard Hamilton, Robert Columbia, Roko C. Buljan, SeinopSys, Sharavnan Kv, Shivangi Chaurasia, SJDS, Squazz, Stephen Leppik, Stewartside, Sunny R Gupta, sv3k, the12, think123, Thomas Gerot, Timon, tmg, Tot Zam, trungk18, Undo, vladdobra, zzzzBov
30	Le forme	Ali Almoullim, Ani Menon, animuson, Aown Muhammad, Chris Rutherford, Cullub, Gabriel Chi Hong Lee, Greg T, j08691, Kimmax, Luca langella, Niek Brouwer, Thomas Gerot
31	Lingue di contenuto	animuson, FelipeAls, Gerold Broser, Isak Combrinck, Muntasir, Shannon Young, unor
32	Mappe di immagini	animuson, Lambda Ninja, RamenChef
33	Marcatura del codice del computer	4444, Naveen Gogineni, Shannon Young, SuperStormer, Tot Zam, unor

34	Meta informazioni	Abhishek Pandey , Akshit Soota , Alexander Wigmore , Angelos Chalaris , Ani Menon , animuson , Anselm Urban , Bálint , bdkopen , Bookeater , Boris , coliff , Domenic , geek1011 , Habel Philip , Hafidz Ilham Aji Permana , Himanshu Vaghela , insertusernamehere , jhoanna , JHS , kelvinelove , m_callens , Matt S , Michael Moriarty , Mr. Alien , Nishchay , Ortomala Lokni , Peter O. , Safoor Safdar , Senjuti Mahapatra , Shannon Young , Stas Christiansen , Stephen Leppik , Ted Goas , Thomas Gerot , timmyRS , tmg , unor , VatsalSura , xims
35	Quotazioni di marcatura	Content Solutions , mnoronha , unor
36	SVG	andreas , Black Mamba , ChrisD , HerrSerker , Patrickdev , Timothy Miller , w5m
37	tabelle	albert , Alexandre N. , animuson , Cedric Zoppolo , Eduardo Molteni , Grant Palin , J F , j08691 , JHS , joe_young , Lambda Ninja , Mottie , Mr Lister , Nijin22 , Prateek , PrAtik Lochawala , Praveen Kumar , Sildoreth , svarog , Ted Goas , tehciolo , Thomas Landauer , zer00ne
38	tabindex	Content Solutions , Psaniko
39	Tela	cone56 , Richard Hamilton , Roko C. Buljan , tonethar , Trevor Clarke , user4040648
40	Titoli	Ani Menon , animuson , dippas , Evan , Infuzed guy , joe_young , MervS , Nathan Arthur , Pseudonym Patel , sasha , Thomas Gerot , unor , V-Kopio
41	Usare l'HTML con i CSS	animuson , bdkopen , Christian Ternus , Community , Euan Williams , feeela , Jones Joseph , Michael Moriarty , Thomas Gerot , thouusten