eBook Gratuit

APPRENEZ HTML

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#html

Table des matières

À propos

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.

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

Chapitre 1: Démarrer avec HTML

Remarques

HTML **(H** yper **t** ext **M** arkup **L** angue) est un XML système de documents compatibles avec la norme annoter avec des 'balises'. Il est utilisé spécifiquement pour créer du contenu pour les pages Web et les applications Web, qui peuvent ensuite être partagées sur un réseau.

Outre le texte, la version actuelle de HTML prend en charge de nombreux types de supports , y compris des images et des vidéos.

Versions

Version	spécification	Date de sortie
1.0	N / A	1994-01-01
2.0	RFC 1866	1995-11-24
3.2	W3C: Spécification HTML 3.2	1997-01-14
4.0	W3C: Spécification HTML 4.0	1998-04-24
4.01	W3C: spécification HTML 4.01	1999-12-24
5	WHATWG: Norme de vie HTML	2014-10-28
5.1	W3C: Spécification HTML 5.1	2016-11-01

Examples

Bonjour le monde

---

introduction

HTML **(H** yper **t** ext **M** arkup **L** anguage) utilise un système de marquage composé d'éléments qui représentent un contenu spécifique. *Le balisage* signifie qu'avec HTML, vous déclarez *ce qui* est présenté à l'utilisateur et non *comment* il est présenté. Les représentations visuelles sont définies par les feuilles de style en cascade (CSS) et réalisées par les navigateurs. Les éléments existants qui permettent une telle utilisation , comme la `font` , par exemple, "sont complètement obsolètes et ne doivent pas être utilisés par les auteurs" [1] .

Le HTML est parfois appelé un langage de programmation mais il n'a pas de logique, de même

qu'un **langage de balisage** . Les balises HTML fournissent une signification sémantique et une lisibilité automatique au contenu de la page.

Un élément se compose généralement d'une balise d'ouverture (`<element_name>`), d'une balise de fermeture (`</element_name>`), qui contient le nom de l'élément entre crochets et du contenu: `<element_name>...content...</element_name>`

Certains éléments HTML n'ont pas de balise de fermeture ni de contenu. Ce sont les éléments vides . Les éléments annulés incluent `<img>` , `<meta>` , `<link>` et `<input>` .

Les noms d'élément peuvent être considérés comme des mots-clés descriptifs du contenu qu'ils contiennent, tels que la `video` , l' `audio` , la `table` , le `footer` .

Une page HTML peut comporter des centaines d'éléments qui sont ensuite lus par un navigateur Web, interprétés et rendus sous forme de contenu lisible ou audible sur l'écran.

Pour ce document, il est important de noter la différence entre les éléments et les balises:

Éléments: `video` , `audio` , `table` , `footer`

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

Élément de l'élément

Décomposons un tag ...

La `<p>` représente un paragraphe commun.

Les éléments ont généralement une balise d'ouverture et une balise de fermeture. La balise d'ouverture contient le nom de l'élément entre crochets (`<p>`). La balise de fermeture est identique à la balise d'ouverture avec l'ajout d'une barre oblique (`/`) entre le crochet ouvrant et le nom de l'élément (`</p>`).

Le contenu peut alors passer entre ces deux balises: `<p>This is a simple paragraph.</p>` .

Créer une page simple

L'exemple HTML suivant crée une page Web simple "Hello World" .

Les fichiers HTML peuvent être créés à l'aide de n'importe quel éditeur de texte . Les fichiers doivent être enregistrés avec une extension `.html` ou `.htm` [2] pour être reconnus en tant que fichiers HTML.

Une fois créé, ce fichier peut être ouvert dans n'importe quel navigateur 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>
```

Page simple décomposer

Ce sont les balises utilisées dans l'exemple:

Marque	Sens
`<!DOCTYPE>`	Définit la version HTML utilisée dans le document. Dans ce cas, c'est HTML5. Voir la rubrique doctypes pour plus d'informations.
`<html>`	Ouvre la page. Aucun balisage ne devrait venir après la balise de fermeture (`</html>`). L'attribut `lang` déclare la langue principale de la page en utilisant les codes de langue ISO (`en` anglais). Consultez la rubrique Langage de contenu pour plus d'informations.
`<head>`	Ouvre la section head, qui n'apparaît pas dans la fenêtre principale du navigateur, mais contient principalement des informations *sur* le document HTML, appelées *métadonnées* . Il peut également contenir des importations à partir de feuilles de style et de scripts externes. La balise de fermeture est `</head>` .
`<meta>`	Donne au navigateur des métadonnées sur le document. L'attribut `charset` déclare le codage de caractères . Les documents HTML modernes doivent toujours utiliser UTF-8 , même s'il ne s'agit pas d'une exigence. En HTML, la `<meta>` ne nécessite pas de balise de fermeture. Consultez la rubrique Meta pour plus d'informations.
`<title>`	Le titre de la page. Le texte écrit entre cette ouverture et la balise de fermeture (`</title>`) sera affiché dans l'onglet de la page ou dans la barre de titre du

Marque	Sens
	navigateur.
`<body>`	Ouvre la partie du document affichée aux utilisateurs, c'est-à-dire tout le contenu visible ou audible d'une page. Aucun contenu ne doit être ajouté après la balise de fermeture `</body>` .
`<h1>`	Un niveau 1 pour la page. Voir les titres pour plus d'informations.
`<p>`	Représente un paragraphe commun de texte.

---

1. ↑ HTML5, 11.2 Fonctions non conformes
2. ↑ `.htm` est hérité de la limite d'extension du fichier DOS à trois caractères.

Lire Démarrer avec HTML en ligne: https://riptutorial.com/fr/html/topic/217/demarrer-avec-html

Chapitre 2: Ancres et Hyperliens

Introduction

Les balises d'ancrage sont couramment utilisées pour lier des pages Web distinctes, mais elles peuvent également être utilisées pour créer des liens entre différents endroits d'un même document, souvent dans la table des matières ou même pour lancer des applications externes. Cette rubrique explique l'implémentation et l'application des balises d'ancrage HTML dans divers rôles.

Syntaxe

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

Paramètres

Paramètre	Détails
`href`	Spécifie l'adresse de destination. Il peut s'agir d'une URL absolue ou relative ou du `name` d'une ancre. Une URL absolue est l'URL complète d'un site Web tel que http://example.com/ . Une URL relative pointe vers un autre répertoire et / ou document du même site Web, par exemple `/about-us/` pointe vers le répertoire «about-us» dans le répertoire racine (`/`). Lorsque vous pointez vers un autre répertoire sans spécifier explicitement le document, les serveurs Web retournent généralement le document «index.html» dans ce répertoire.
`hreflang`	Spécifie la langue de la ressource liée par l'attribut `href` (qui doit être présent avec celui-ci). Utilisez les valeurs linguistiques de BCP 47 pour HTML5 et RFC 1766 pour HTML 4.
`rel`	Spécifie la relation entre le document actuel et le document lié. Pour HTML5, les valeurs doivent être définies dans la spécification ou enregistrées dans le wiki Microformats .
`target`	Spécifie où ouvrir le lien, par exemple dans un nouvel onglet ou une nouvelle fenêtre. Les valeurs possibles sont `_blank` , `_self` , `_parent` , `_top` et `framename` (dépréciée). Forcer un tel comportement n'est pas recommandé car il viole le contrôle de l'utilisateur sur un site Web.
`title`	Spécifie des informations supplémentaires sur un lien. L'information est le plus souvent affichée sous forme de texte d'info-bulle lorsque le curseur se déplace sur le lien. Cet attribut n'est pas limité aux liens, il peut être utilisé sur presque toutes les balises HTML.
`download`	Indique que la cible sera téléchargée lorsqu'un utilisateur clique sur le lien

Paramètre	Détails
	hypertexte. La valeur de l'attribut sera le nom du fichier téléchargé. Il n'y a pas de restrictions sur les valeurs autorisées et le navigateur détecte automatiquement l'extension de fichier correcte et l'ajoute au fichier (.img, .pdf, etc.). Si la valeur est omise, le nom de fichier d'origine est utilisé.

Examples

Lien vers un autre site

Ceci est l'utilisation de base de l' élément `<a>` (**un** élément nchor) :

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

Il crée un lien hypertexte vers l'URL `http://example.com/` comme spécifié par l' `href` (référence hypertexte), avec le texte d'ancrage "Lien vers exemple.com". Cela ressemblerait à quelque chose comme ceci:

> Lien vers example.com

Pour indiquer que ce lien mène à un site Web externe, vous pouvez utiliser le type de lien `external` :

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

Vous pouvez créer un lien vers un site qui utilise un protocole autre que HTTP. Par exemple, pour créer un lien vers un site FTP, vous pouvez le faire,

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

Dans ce cas, la différence réside dans le fait que cette balise d'ancrage demande au navigateur de l'utilisateur de se connecter à `example.com` à l'aide du protocole FTP (File Transfer Protocol) plutôt qu'au protocole HTTP (Hypertext Transfer Protocol).

Cela pourrait être un lien vers un site FTP

Ouvrir le lien dans un nouvel onglet / une nouvelle fenêtre

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

L'attribut `target` spécifie où ouvrir le lien. En le définissant sur `_blank` , vous indiquez au navigateur de l'ouvrir dans un nouvel onglet ou une nouvelle fenêtre (selon les préférences de l'utilisateur).

> **VULNÉRABILITÉ DE SÉCURITÉ AVERTISSEMENT!**

Utiliser `target="_blank"` donne au site d'ouverture un accès partiel à l'objet `window.opener` via JavaScript, ce qui permet à cette page d'accéder et de modifier le `window.opener.location` de *votre* page et de rediriger les utilisateurs vers des sites malveillants ou des sites de phishing.

Chaque fois que vous utilisez ceci pour les pages que vous ne contrôlez pas, ajoutez `rel="noopener"` à votre lien pour empêcher l'objet `window.opener` d'être envoyé avec la requête.

Actuellement, Firefox ne supporte pas `noopener` , vous devrez donc utiliser `rel="noopener noreferrer"` pour un effet maximum.

Lien vers une ancre

Les ancres peuvent être utilisées pour accéder à des balises spécifiques sur une page HTML. La balise `<a>` peut pointer sur tout élément ayant un attribut `id` . Pour en savoir plus sur les identifiants, consultez la documentation sur les classes et les identifiants . Les ancres sont principalement utilisées pour accéder à une sous-section d'une page et sont utilisées conjointement avec des balises d'en-tête.

Supposons que vous ayez créé une page (`page1.html`) sur de nombreux sujets:

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

Une fois que vous avez plusieurs sections, vous pouvez créer une table des matières en haut de la page avec des liens rapides (ou des signets) vers des sections spécifiques.

Si vous avez attribué un attribut `id` à vos sujets, vous pouvez alors y accéder

```
<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>
```

Maintenant, vous pouvez utiliser l'ancre dans votre table des matières:

```
<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>
```

Ces ancres sont également attachées à la page Web sur laquelle elles se trouvent (`page1.html`). Vous pouvez donc vous connecter sur le site d'une page à l'autre en référençant le nom de la page *et de l'* ancre.

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

Lien qui exécute JavaScript

Il suffit d'utiliser le protocole `javascript:` pour exécuter le texte en JavaScript au lieu de l'ouvrir comme un lien normal:

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

Vous pouvez également obtenir la même chose en utilisant l'attribut `onclick` :

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

Le `return false;` est nécessaire pour empêcher votre page de défiler vers le haut lorsque le lien vers `#` est cliqué. Assurez-vous d'inclure tout le code que vous souhaitez exécuter avant, car le retour arrêtera l'exécution de code supplémentaire.

À noter également que vous pouvez inclure un point d'exclamation `!` après le hashtag afin d'empêcher la page de défiler vers le haut. Cela fonctionne car tout slug invalide ne fera pas défiler le lien *n'importe où* sur la page, car il ne pouvait pas localiser l'élément auquel il fait référence (un élément avec `id="!"`). Vous pouvez également utiliser un slug non valide (tel que `#scrollsNowhere`) pour obtenir le même effet. Dans ce cas, `return false;` n'est pas requis:

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

> **Devriez-vous utiliser l'un de ces éléments?**
>
> La réponse est presque certainement ***non*** . L'exécution de JavaScript en ligne avec l'élément comme celui-ci est une mauvaise pratique. Envisagez d'utiliser des solutions JavaScript pures qui recherchent l'élément dans la page et y associent une fonction. Écouter un événement
>
> Pensez également à savoir si cet élément est réellement un *bouton* plutôt qu'un *lien* . Si oui, vous devriez utiliser `<button>` .

Lien vers une page sur le même site

Vous pouvez utiliser un chemin relatif pour créer un lien vers des pages du même site Web.

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

L'exemple ci-dessus irait à l' `example` fichier dans le répertoire racine (`/`) du serveur.

Si ce lien se trouvait sur http://exemple.com , les deux liens suivants amèneraient l'utilisateur au même emplacement

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

Les deux ci-dessus iraient au fichier de la `page` dans le répertoire racine de `example.com` .

Lien qui exécute le client de messagerie

Utilisation de base

Si la valeur de l' `href` commence par `mailto:` il essaiera d'ouvrir un client de messagerie sur le clic:

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

Cela mettra l'adresse e-mail `example@example.com` comme destinataire du nouvel e-mail créé.

Cc et Bcc

Vous pouvez également ajouter des adresses pour les destinataires cc ou bcc en utilisant la syntaxe suivante:

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

Sujet et texte du corps

Vous pouvez également renseigner le sujet et le corps du nouvel email:

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

Ces valeurs doivent être encodées en URL .

En cliquant sur un lien avec `mailto:` essaiera d'ouvrir le client de messagerie par défaut spécifié par votre système d'exploitation ou il vous demandera de choisir quel client vous souhaitez utiliser. Toutes les options spécifiées après l'adresse du destinataire ne sont pas prises en charge par tous les clients de messagerie.

Lien qui compose un numéro

Si la valeur de l' `href` commence par `tel:` :, votre appareil composera le numéro lorsque vous cliquez dessus. Cela fonctionne sur les appareils mobiles ou sur les ordinateurs / tablettes exécutant des logiciels - tels que Skype ou FaceTime - qui peuvent faire des appels téléphoniques.

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

La plupart des appareils et des programmes inviteront l'utilisateur à confirmer le numéro qu'il va composer.

Lire Ancres et Hyperliens en ligne: https://riptutorial.com/fr/html/topic/254/ancres-et-hyperliens

Chapitre 3: ARIA

Syntaxe

- aria-live
- aria-pertinent
- aria-autocomplete
- aria-vérifié
- aria-handicapé
- aria-expand
- aria-haspopup
- aria-caché
- aria-invalide
- label aria
- niveau aria
- aria-multiline
- aria-multiselectable
- orientation aria
- aria-pressé
- aria-readonly
- aria-required
- aria-sélectionné
- aria-tri
- aria-valuemax
- aria-valuemin
- aria-valuenow
- aria-valuetext
- aria-atomic
- aria-occupé
- aria-dropeffect
- aria-traîné
- aria-activedescendant
- aria-contrôles
- aria-décrit par
- aria-flowto
- aria-labelledby
- Aria-possède
- aria-posinset
- aria-setsize

Remarques

ARIA est une spécification pour la description sémantique des applications Web riches. Suivre les normes ARIA peut améliorer l'accessibilité pour ceux qui utilisent des technologies d'assistance

(telles qu'un lecteur d'écran) pour accéder à votre contenu.

Examples

role = "alerte"

Un message contenant des informations importantes et généralement sensibles au temps.

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

> Notez que j'ai inclus à la fois `role="alert"` et `aria-live="assertive"` en même temps. Ce sont des attributs synonymes, mais certains lecteurs d'écran ne supportent que l'un ou l'autre. En utilisant les deux simultanément, nous maximisons donc les chances que la région en direct fonctionne comme prévu.
>
> Source - Heydon Pickering 'Quelques exemples pratiques d'ARIA'

role = "alertdialog"

Un type de boîte de dialogue contenant un message d'alerte, où le focus initial est dirigé vers un élément de la boîte de dialogue.

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

role = "application"

Une région déclarée en tant qu'application Web, par opposition à un document Web. Dans cet exemple, l'application est une calculatrice simple qui peut ajouter deux nombres ensemble.

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

role = "article"

Section d'une page constituée d'une composition qui constitue une partie indépendante d'un document, d'une page ou d'un site.

> La définition d'un attribut ARIA et / ou d'un attribut aria- * correspondant à la sémantique implicite ARIA implicite est inutile et n'est pas recommandée car ces propriétés sont déjà définies par le navigateur.

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

Vous utiliseriez `role=article` sur des éléments non sémantiques (non recommandé, non valide)

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

Entrée W3C pour le `role=article`

role = "banner"

Une région qui contient principalement du contenu orienté site, plutôt que du contenu spécifique à la page.

```
<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 = "bouton"

Une entrée permettant des actions déclenchées par l'utilisateur lorsque l'utilisateur clique ou appuie dessus.

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

role = "cell"

Une cellule dans un conteneur tabulaire.

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

role = "checkbox"

Une entrée vérifiable qui a trois valeurs possibles: true, false ou mixed.

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

role = "en-tête de colonne"

Une cellule contenant des informations d'en-tête pour une colonne.

```
<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 = "combobox"

Une présentation d'une sélection; généralement similaire à une zone de texte où les utilisateurs peuvent taper en avant pour sélectionner une option, ou taper pour entrer du texte arbitraire en tant que nouvel élément dans la liste.

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

En règle générale, vous utiliseriez JavaScript pour créer le reste de la fonctionnalité de sélection de tête ou de liste.

role = "complémentaire"

Une section de support du document, conçue pour compléter le contenu principal à un niveau similaire dans la hiérarchie DOM, mais reste significative lorsqu'elle est séparée du contenu principal.

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

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

role = "contentinfo"

Une grande région perceptible contenant des informations sur le document parent.

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

role = "definition"

Une définition d'un terme ou d'un concept.

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

role = "dialog"

Une boîte de dialogue est une fenêtre d'application conçue pour interrompre le traitement en cours d'une application afin d'inviter l'utilisateur à saisir des informations ou à demander une réponse.

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

role = "répertoire"

Une liste de références aux membres d'un groupe, telle qu'une table des matières statique.

```
<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 = "document"

Une région contenant des informations connexes déclarées en tant que contenu de document, par opposition à une application Web.

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

role = "form"

Une région de repère qui contient une collection d'éléments et d'objets qui, dans leur ensemble, se combinent pour créer un formulaire.

L'utilisation de l'élément HTML sémantiquement correct `<form>` implique la sémantique ARIA par défaut, ce qui signifie que `role=form` n'est pas requis car vous ne devez pas appliquer de rôle contrasté à un élément déjà sémantique, car l'ajout d'un rôle remplace la sémantique native d'un élément.

> La définition d'un attribut ARIA et / ou d'un attribut aria- * correspondant à la sémantique implicite ARIA implicite est inutile et n'est pas recommandée car ces propriétés sont déjà définies par le navigateur.

```
<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>
```

Vous utiliseriez `role=form` sur des éléments non sémantiques (non recommandé, non valide)

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

role = "grid"

Une grille est un contrôle interactif qui contient des cellules de données tabulaires disposées en lignes et en colonnes, comme une table.

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

role = "gridcell"

Une cellule dans une grille ou une grille.

```
<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 = "group"

Un ensemble d'objets d'interface utilisateur qui ne sont pas destinés à être inclus dans un résumé de page ou une table des matières par des technologies d'assistance.

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

role = "en-tête"

Un en-tête pour une section de la page.

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

role = "img"

Un conteneur pour une collection d'éléments qui forment une image.

```
<figure role="img">
  <img alt="A cute cat." src="albert.jpg">
  <figcaption>This is my cat, Albert.</figcaption>
<figure>
```

role = "link"

Une référence interactive à une ressource interne ou externe qui, lorsqu'elle est activée, oblige l'agent utilisateur à accéder à cette ressource.

> Dans la majorité des cas, il est inutile de définir un attribut ARIA et / ou un attribut aria-* correspondant à la sémantique implicite ARIA implicite, ce qui n'est pas recommandé car ces propriétés sont déjà définies par le navigateur.
>
> Source - https://www.w3.org/TR/html5/dom.html#aria-usage-note

role = "list"

Un groupe d'éléments de liste non interactifs.

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

role = "listbox"

Un widget qui permet à l'utilisateur de sélectionner un ou plusieurs éléments dans une liste de choix.

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

En règle générale, vous utiliseriez JavaScript pour créer la fonctionnalité de sélection multiple.

role = "listitem"

Un seul élément dans une liste ou un répertoire.

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

role = "log"

Un type de région en direct dans lequel de nouvelles informations sont ajoutées dans un ordre significatif et les anciennes informations peuvent disparaître.

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

role = "main"

Le contenu principal d'un document.

```
<!-- header & nav here -->
<div role="main">
```

```
  <p>Lorem ipsum...</p>
</div>
<!-- footer here -->
```

role = "chapiteau"

Un type de région en direct où les informations non essentielles changent fréquemment.

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

role = "math"

Contenu qui représente une expression mathématique.

```
<img role="math" alt="y=mx+b" src="slope.png">
```

role = "menu"

Un type de widget qui offre une liste de choix à l'utilisateur.

```
<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 = "menubar"

Une présentation de menu qui reste généralement visible et est généralement présentée horizontalement.

```
<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"

Une option dans un groupe de choix contenus dans un menu ou une barre de menus.

```
<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 menuitem vérifiable qui a trois valeurs possibles: true, false ou mixed.

```
<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 menuitem vérifiable dans un groupe de rôles menuitemradio, dont un seul peut être vérifié à la fois.

```
<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 = "navigation"

Une collection d'éléments de navigation (généralement des liens) pour naviguer dans le document ou les documents associés.

```
<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"

Une section dont le contenu est parenthétique ou accessoire au contenu principal de la ressource.

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

role = "option"

Un élément sélectionnable dans une liste de sélection.

```
<ul role="listbox">
```

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

role = "presentation"

Un élément dont la sémantique implicite du rôle natif ne sera pas mappée à l'API d'accessibilité.

```
<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 élément qui affiche l'état d'avancement pour les tâches qui prennent beaucoup de temps.

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

role = "radio"

Une entrée vérifiable dans un groupe de rôles radio, dont un seul peut être vérifié à la fois.

```
<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 = "region"

Une grande partie perceptible d'une page Web ou d'un document que l'auteur juge suffisamment importante pour être incluse dans un résumé de page ou une table des matières, par exemple une zone de la page contenant des statistiques d'événements sportifs en direct.

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

role = "radiogroup"

Un groupe de boutons 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 = "row"

Une rangée de cellules dans un conteneur tabulaire.

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

role = "rowgroup"

Un groupe contenant un ou plusieurs éléments de ligne dans une grille.

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

role = "rowheader"

Une cellule contenant des informations d'en-tête pour une ligne dans une grille.

```
<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 = "scrollbar"

Objet graphique qui contrôle le défilement du contenu dans une zone de visualisation, que le contenu soit entièrement affiché ou non dans la zone de visualisation.

```
<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"

Une région de repère qui contient une collection d'éléments et d'objets qui, dans leur ensemble, se combinent pour créer une fonction de recherche.

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

role = "searchbox"

Un type de zone de texte destiné à spécifier des critères de recherche.

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

role = "separator"

Un séparateur qui sépare et distingue des sections de contenu ou des groupes de menuitems.

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

role = "curseur"

Une entrée utilisateur où l'utilisateur sélectionne une valeur dans une plage donnée.

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

role = "spinbutton"

Une forme de plage qui attend de l'utilisateur qu'il choisisse parmi des choix discrets.

```
<input
  role="spinbutton"
  aria-valuemax="100"
  aria-valuemin="0"
  aria-valuenow="25"
  type="number"
  value="25">
```

role = "status"

Un conteneur dont le contenu est une information consultative pour l'utilisateur mais n'est pas assez important pour justifier une alerte, souvent mais pas nécessairement présenté comme une barre d'état.

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

role = "switch"

Type de case à cocher représentant les valeurs d'activation / désactivation, par opposition aux valeurs cochées / non cochées.

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

role = "tab"

Une étiquette de regroupement fournissant un mécanisme pour sélectionner le contenu de l'onglet à rendre à l'utilisateur.

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

role = "table"

Une section contenant des données organisées en lignes et en colonnes. Le rôle de table est destiné aux conteneurs tabulaires qui ne sont pas interactifs.

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

```
  </tbody>
</table>
```

role = "tablist"

Une liste d'éléments d'onglet, qui sont des références à des éléments 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 conteneur pour les ressources associées à un onglet, où chaque onglet est contenu dans une tablist.

```
<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 = "textbox"

Entrée qui autorise le texte libre comme valeur.

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

role = "timer"

Type de région active contenant un compteur numérique indiquant la durée écoulée depuis un point de départ ou la durée restante jusqu'à un point final.

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

role = "barre d'outils"

Une collection de boutons de fonctions couramment utilisés représentés sous forme visuelle compacte.

```
<ul role="toolbar">
  <li><img alt="New" src="new.png"></li>
```

```
  <li><img alt="Open" src="open.png"></li>
  <li><img alt="Save" src="save.png"></li>
  <li><img alt="Close" src="close.png"></li>
</ul>
```

role = "tooltip"

Une fenêtre contextuelle qui affiche une description pour un élément.

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

En règle générale, l'info-bulle est masquée. En utilisant JavaScript, l'info-bulle sera affichée après un délai lorsque l'utilisateur survole l'élément décrit.

role = "tree"

Type de liste pouvant contenir des groupes imbriqués de sous-niveau pouvant être réduits et développés.

```
<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"

Une grille dont les lignes peuvent être étendues et réduites de la même manière que pour un arbre.

role = "treeitem"

Un élément d'option d'un arbre. C'est un élément dans une arborescence qui peut être développé ou réduit s'il contient un groupe de sous-niveaux d'arborescence.

```
<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>
```

Lire ARIA en ligne: https://riptutorial.com/fr/html/topic/2734/aria

Chapitre 4: Attributs d'événement HTML

Examples

Événements de formulaire HTML

Événements déclenchés par des actions à l'intérieur d'un formulaire HTML (s'applique à presque tous les éléments HTML, mais est le plus utilisé dans les éléments de formulaire):

Attribut	La description
le flou	Tire le moment où l'élément perd le focus
sur le changement	Déclenche le moment où la valeur de l'élément est modifiée
oncontextmenu	Script à exécuter lorsqu'un menu contextuel est déclenché
onfocus	Tire le moment où l'élément se concentre
oninput	Script à exécuter lorsqu'un élément obtient une entrée utilisateur
surinvalide	Script à exécuter lorsqu'un élément n'est pas valide
onreset	Se déclenche lorsque le bouton Réinitialiser dans un formulaire est cliqué
sur la recherche	Se déclenche lorsque l'utilisateur écrit quelque chose dans un champ de recherche (pour <input = "search">)
onselect	Se déclenche après qu'un texte a été sélectionné dans un élément
onsubmit	Se déclenche lorsqu'un formulaire est soumis

Événements clavier

Attribut	La description
onkeydown	Se déclenche lorsqu'un utilisateur appuie sur une touche
onkeypress	Se déclenche lorsqu'un utilisateur appuie sur une touche
onkeyup	Se déclenche lorsqu'un utilisateur libère une clé

Lire Attributs d'événement HTML en ligne: https://riptutorial.com/fr/html/topic/10924/attributs-d-evenement-html

Chapitre 5: Attributs de données

Syntaxe

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

Paramètres

Valeur	La description
somevalue	Spécifie la valeur de l'attribut (sous forme de chaîne)

Examples

Utilisation des attributs de données

`data-*` attributs de `data-*` HTML5 `data-*` constituent un moyen pratique de stocker des données dans des éléments HTML. Les données stockées peuvent être lues ou modifiées en utilisant JavaScript

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

- La structure de l'attribut de `data-*` est `data-*` , c'est-à-dire que le nom de l'attribut de données vient après la partie de `data-` . En utilisant ce nom, l'attribut est accessible.
- Les données au format chaîne (y compris `json`) peuvent être stockées en utilisant l'attribut `data-*` .

Les anciens navigateurs prennent en charge

Les attributs de données ont été introduits dans HTML5, qui est pris en charge par tous les navigateurs modernes, mais les anciens navigateurs avant HTML5 ne reconnaissent pas les attributs de données.

Toutefois, dans les spécifications HTML, les attributs qui ne sont pas reconnus par le navigateur doivent être laissés seuls et le navigateur les ignore simplement lors du rendu de la page.

Les développeurs Web ont utilisé ce fait pour créer des attributs non standard qui sont des attributs qui ne font pas partie des spécifications HTML. Par exemple, l'attribut `value` dans la ligne ci-dessous est considéré comme un attribut non standard car les spécifications de la `<img>` n'ont pas d'attribut `value` et ce n'est pas un attribut global:

```
<img src="sample.jpg" value="test" />
```

Cela signifie que même si les attributs de données ne sont pas pris en charge dans les navigateurs plus anciens, ils travaillent encore et vous pouvez définir et récupérer les utiliser les mêmes génériques JavaScript `setAttribute` et `getAttribute` méthodes, mais vous ne pouvez pas utiliser le nouveau `dataset` de `dataset` propriété qui est uniquement pris en charge dans les navigateurs modernes.

Lire Attributs de données en ligne: https://riptutorial.com/fr/html/topic/1182/attributs-de-donnees

Chapitre 6: Attributs globaux

Paramètres

Attribut	La description
`class`	Définit un ou plusieurs noms de classe pour un élément. Voir Classes et ID .
`contenteditable`	Définit si le contenu d'un élément peut être modifié.
`contextmenu`	Définit un menu contextuel affiché lorsqu'un utilisateur clique avec le bouton droit sur un élément.
`dir`	Définit la direction du texte pour le texte dans un élément.
`draggable`	Définit si un élément peut être déplacé.
`hidden`	Masque un élément qui n'est pas actuellement utilisé sur la page.
`id`	Définit un identifiant unique pour un élément. Voir Classes et ID .
`lang`	Définit la langue du contenu d'un élément et ses valeurs d'attribut de texte. Voir Langues de contenu .
`spellcheck`	Définit s'il faut orthographier / vérifier la grammaire du contenu d'un élément.
`style`	Définit un ensemble de styles CSS intégrés pour un élément.
`tabindex`	Définit l'ordre dans lequel les éléments d'une page sont parcourus par le raccourci clavier de l'onglet.
`title`	Définit des informations supplémentaires sur un élément, généralement sous la forme d'un texte d'info-bulle au survol de la souris.
`translate`	Définit s'il faut traduire le contenu d'un élément.

Remarques

Les attributs globaux sont simplement attribués et peuvent être appliqués à tout élément du document entier.

Examples

Attribut Contenteditable

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

En cliquant sur le paragraphe, son contenu peut être modifié de la même manière qu'un champ de saisie de texte.

Lorsque l'attribut `contenteditable` n'est pas défini sur un élément, l'élément l'héritera de son parent. Ainsi, tout le texte enfant d'un élément modifiable par le contenu sera également modifiable, mais vous *pouvez* le désactiver pour un texte spécifique, comme ceci:

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

Notez qu'un élément de texte non éditable à l'intérieur d'un élément éditable aura toujours un curseur de texte hérité de son parent.

Lire Attributs globaux en ligne: https://riptutorial.com/fr/html/topic/2811/attributs-globaux

Chapitre 7: Barres de navigation

Examples

Barre de navigation de base

Les barres de navigation sont essentiellement une liste de liens, de sorte que les éléments ul et li sont utilisés pour encapsuler les liens de navigation.

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

HTML5 Barre de navigation

Pour créer une barre de navigation à l'aide de l'élément de navigation HTML5, intégrez les liens dans la balise de navigation.

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

Lire Barres de navigation en ligne: https://riptutorial.com/fr/html/topic/10725/barres-de-navigation

Chapitre 8: Cartes d'image

Syntaxe

- `<img usemap="#[map-name]">`
- `<map name="[map-name]"></map>`
- `<area>`

Paramètres

Tag / Attribut	Valeur
`<img>`	Vous trouverez ci-dessous les attributs spécifiques à la carte image à utiliser avec `<img>` . Les attributs `<img>` réguliers s'appliquent.
`usemap`	Le `name` de la carte avec un symbole de hachage qui lui a été ajouté. Par exemple, pour une carte avec `name="map"` , l'image doit avoir `usemap="#map"` .
`<map>`	
`name`	Le nom de la carte pour l'identifier. À utiliser avec l'attribut `usemap` l'image.
`<area>`	Vous trouverez ci-dessous des attributs spécifiques à `<area>` . Lorsque `href` est spécifié, ce qui rend la `<area>` un lien, `<area>` prend également en charge tous les attributs de la balise d'ancrage (`<a>`) à l' exception `ping` . Voyez-les dans les documents MDN .
`alt`	Le texte de remplacement à afficher si les images ne sont pas prises en charge. Ceci n'est nécessaire que si `href` est également défini sur la `<area>` .
`coords`	Les coordonnées indiquant la zone sélectionnable. Lorsque `shape="polygon"` , il convient de définir une liste de paires "x, y" séparées par des virgules (ie `shape="polygon" coords="x1, y1, x2, y2, x3, y3, ..."`). Lorsque `shape="rectangle"` , cela devrait être mis à `left, top, right, bottom` . Lorsque `shape="circle"` , cela doit être réglé sur `centerX, centerY, radius` .
`href`	L'URL du lien hypertexte, si spécifié. S'il est omis, la `<area>` ne représentera pas un lien hypertexte.
`shape`	La forme de la `<area>` . Peut être réglé à `default` pour sélectionner l'image entière (aucun `coords` attribut nécessaire), `circle` ou `circ` pour un cercle, `rectangle` ou `rect` d'un rectangle, et un `polygon` ou `poly` pour une région polygonale spécifiée par des points d'angle.

Remarques

- La liste de paramètres ci-dessus est modifiée à partir des documents MDN: `<map>` et `<area>` .
- Il est possible de créer les coordonnées d'une carte d'image pour une image avec des formes plus simples (comme dans l'exemple d'introduction ci-dessus) avec un éditeur d'image qui affiche les coordonnées (comme GIMP). Cependant, il peut être plus facile en général d'utiliser un générateur de carte d'image, tel que celui-ci .

Examples

Introduction aux cartes d'image

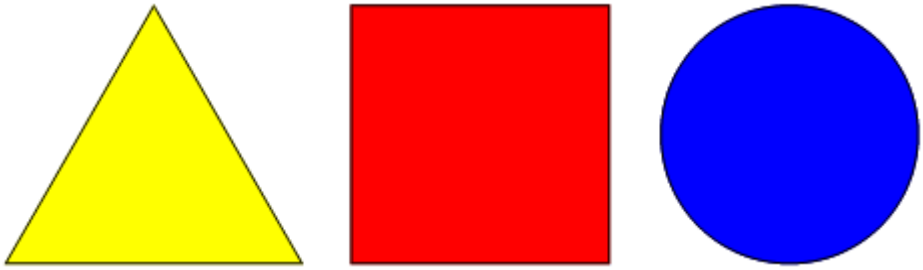
La description

Une image cartographie est une image avec des zones cliquables qui agissent généralement comme des hyperliens.

L'image est définie par la `<img>` et la carte est définie par une `<map>` avec des balises `<area>` pour désigner chaque zone cliquable. Utilisez les `usemap` et `name` pour lier l'image et la carte.

Exemple de base

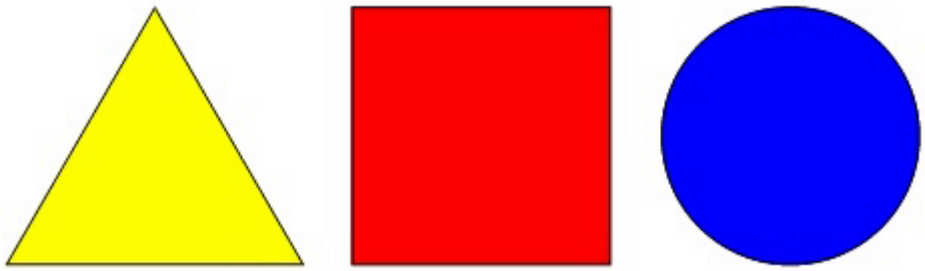
Pour créer une carte-image afin de pouvoir cliquer sur chacune des formes de l'image ci-dessous:



Le code serait comme suit:

```
<img src="http://jaced.com/blogpix/2007/trisquarecircle/002.gif" usemap="#shapes">
<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>
```

Vous devriez voir que le navigateur reconnaît les zones lorsque le curseur devient un pointeur. Voir une démonstration en direct sur JSFiddle, ou voir une démonstration ci-dessous:



Lire Cartes d'image en ligne: https://riptutorial.com/fr/html/topic/3819/cartes-d-image

Chapitre 9: Citations de marquage

Remarques

`cite` éléments `cite` et `blockquote` ne doivent pas être utilisés dans le but de représenter une conversation, des transcriptions de conversations, des dialogues dans des scripts, des enregistrements de messages instantanés et d'autres situations dans lesquelles différents acteurs interviennent à tour de rôle.

Examples

En ligne avec

L' **élément** `q` peut être utilisé pour une citation qui fait partie d'une phrase:

```
<p>She wrote <q>The answer is 42.</q> and everyone agreed.</p>
```

Guillemets

4.01

Les guillemets ne doivent pas être ajoutés. Les agents utilisateurs devraient (en HTML 4.01) resp. must (en HTML 4.0) les rendre automatiquement.

5

Les guillemets ne doivent pas être ajoutés. Les agents utilisateurs les rendront automatiquement.

URL source (attribut `cite`)

L' **attribut** `cite` peut être utilisé pour référencer l'URL de la source citée:

```
<p>She wrote <q cite="http://example.com/blog/hello-world">The answer is 42.</q> and everyone
agreed.</p>
```

Notez que les navigateurs n'affichent généralement pas cette URL. Par conséquent, si la source est pertinente, vous devez ajouter un lien hypertexte (`a` élément).

Bloquer avec

L' **élément** `blockquote` peut être utilisé pour un devis (au niveau du bloc):

```
<blockquote>
  <p>The answer is 42.</p>
</blockquote>
```

URL source (attribut `cite`)

L' **attribut `cite`** peut être utilisé pour référencer l'URL de la source citée:

```
<blockquote cite="http://example.com/blog/hello-world">
  <p>The answer is 42.</p>
</blockquote>
```

Notez que les navigateurs ne montrent généralement pas cette URL, donc si la source est pertinente, vous devez ajouter un lien hypertexte (`a` élément) en plus (voir la section *Citation / Attribution* concernant l'emplacement de ce lien).

Citation / Attribution

4.01

La citation / attribution ne doit pas faire partie de l'élément `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>
```

Vous pouvez ajouter un élément `div` pour regrouper la citation et la citation, mais il n'existe aucun moyen de les associer sémantiquement.

L' **élément `cite`** peut être utilisé pour la référence de la source citée (mais pas pour le nom de l'auteur).

5

La citation / attribution (par exemple, le lien hypertexte donnant l'URL source) peut se trouver à l'intérieur du `blockquote` , mais dans ce cas, il doit être dans un élément `cite` (pour les attributions de texte) ou un élément `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' **élément `cite`** peut être utilisé pour la référence de la source citée ou pour le nom de l'auteur de la citation.

Lire Citations de marquage en ligne: https://riptutorial.com/fr/html/topic/2943/citations-de-marquage

Chapitre 10: Classes et identifiants

Introduction

Les classes et les identifiants facilitent le référencement des éléments HTML à partir des scripts et des feuilles de style. L'attribut class peut être utilisé sur une ou plusieurs balises et est utilisé par CSS pour le style. Les identifiants sont cependant destinés à faire référence à un seul élément, ce qui signifie que le même identifiant ne doit jamais être utilisé deux fois. Les identifiants sont généralement utilisés avec JavaScript et les liens internes aux documents et sont déconseillés dans les CSS. Cette rubrique contient des explications utiles et des exemples d'utilisation correcte des attributs class et ID en HTML.

Syntaxe

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

Paramètres

Paramètre	Détails
classe	Indique la classe de l'élément (non unique)
id	Indique l'ID de l'élément (unique dans le même contexte)

Remarques

- Les `class` et les `id` sont tous deux des attributs globaux et peuvent donc être affectés à n'importe quel élément HTML.
- Les noms de classe doivent commencer par une lettre (AZ ou az) et peuvent être suivis de lettres, de chiffres, de tirets et de traits de soulignement.
- En `HTML5` , les attributs class et id peuvent être utilisés sur n'importe quel élément. Dans HTML 4.0.1, ils étaient interdits aux balises `<base>` , `<head>` , `<html>` , `<meta>` , `<param>` , `<script>` , `<style>` et `<title>` .
- Un élément peut avoir une ou plusieurs classes. Les classes sont séparées par des espaces et ne peuvent pas contenir d'espaces eux-mêmes.
- Un élément ne peut avoir qu'un seul identifiant et doit être unique dans son contexte (c'est-à-dire une page Web). Les identifiants ne peuvent pas non plus contenir d'espaces eux-mêmes.

Examples

Donner un élément à une classe

Les classes sont des identificateurs pour les éléments auxquels elles sont affectées. Utilisez l'attribut `class` pour affecter une classe à un élément.

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

Pour affecter plusieurs classes à un élément, séparez les noms de classe par des espaces.

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

Utiliser des classes en CSS

Les classes peuvent être utilisées pour styliser certains éléments sans changer tous les éléments de ce type. Par exemple, ces deux `span` éléments peuvent avoir Stylings complètement différents:

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

Des classes du même nom peuvent être attribuées à un nombre quelconque d'éléments sur une page et elles recevront toutes le style associé à cette classe. Cela sera toujours vrai sauf si vous spécifiez l'élément dans le CSS.

Par exemple, nous avons deux éléments, les deux avec la classe `highlight` :

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

Si notre CSS est comme ci-dessous, alors la couleur verte sera appliquée au texte dans les deux éléments:

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

Cependant, si nous voulons seulement cibler les `div` avec le `highlight` la classe, nous pouvons ajouter une spécificité comme ci-dessous:

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

Néanmoins, lors du stylisme avec CSS, il est généralement recommandé de n’utiliser que des classes (par exemple, `.highlight`) plutôt que des éléments avec des classes (par exemple, `div.highlight`).

Comme pour tout autre sélecteur, les classes peuvent être imbriquées:

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

Vous pouvez également enchaîner le sélecteur de classe pour sélectionner uniquement les éléments qui combinent plusieurs classes. Par exemple, s'il s'agit de notre HTML:

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

Et nous voulons colorer cette partie spécifique du texte en rose, nous pouvons faire ce qui suit dans notre CSS:

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

Donner un identifiant à un élément

L'attribut ID d'un élément est un identifiant qui doit être unique dans tout le document. Son but est d'identifier de manière unique l'élément lors de la liaison (à l'aide d'une ancre), du script ou du styling (avec CSS).

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

Vous ne devriez pas avoir deux éléments avec le même ID dans le même document, même si les attributs sont attachés à deux types d'éléments différents. Par exemple, le code suivant est incorrect:

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

Les navigateurs feront de leur mieux pour rendre ce code, mais un comportement inattendu peut se produire lors de la création de styles avec CSS ou de l'ajout de fonctionnalités avec JavaScript.

Pour référencer des éléments par leur ID en CSS, préfixez l'ID avec `#` .

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

Pour accéder à un élément avec un ID sur une page donnée, ajoutez `#` avec le nom de l'élément dans l'URL.

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

Cette fonctionnalité est prise en charge dans la plupart des navigateurs et ne nécessite pas JavaScript ou CSS supplémentaire pour fonctionner.

Problèmes liés aux ID en double

Avoir plus d'un élément avec le même ID est un problème difficile à résoudre. L'analyseur HTML tentera généralement de rendre la page dans tous les cas. Généralement aucune erreur ne se produit. Mais le rythme pourrait se retrouver dans une page Web qui se comporte mal.

Dans cet exemple:

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

Les sélecteurs CSS fonctionnent toujours

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

Mais JavaScript ne gère pas les deux éléments:

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

Dans ce cas, la variable `html` ne porte que le premier contenu `div ("a")` .

Valeurs acceptables

Pour un identifiant

5

Les seules restrictions sur la valeur d'un `id` sont:

1. il doit être unique dans le document
2. il ne doit contenir aucun espace
3. il doit contenir au moins un caractère

Donc, la valeur peut être tous les chiffres, un seul chiffre, juste des caractères de ponctuation, inclure des caractères spéciaux, peu importe. Juste pas de blanc.

Donc ce sont valables:

```
<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>
```

Ceci est invalide:

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

Ceci est également invalide, lorsqu'il est inclus dans le même document:

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

4.01

Une valeur d' `id` doit commencer par une lettre, qui ne peut alors être suivie que par:

- lettres (AZ / az)
- chiffres (0-9)
- traits d'union ("-")
- traits de soulignement ("_")
- colons (":")
- périodes (".")

En vous référant au premier groupe d'exemples dans la section HTML5 ci-dessus, un seul est valide:

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

Ce sont également valables:

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

Encore une fois, s'il ne commence pas par une lettre (majuscule ou minuscule), ce n'est pas valide.

Pour une classe

Les règles pour les classes sont essentiellement les mêmes que pour un `id` . La différence est que `class` valeurs de `class` *n'ont pas* besoin d'être uniques dans le document.

En vous référant aux exemples ci-dessus, bien que cela ne soit pas valable dans le même document:

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

Ceci est parfaitement correct:

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

Remarque importante: Comment les valeurs d'ID et de classe sont traitées en dehors du code HTML

Gardez à l'esprit que les règles et les exemples ci-dessus s'appliquent dans le contexte du HTML.

L'utilisation de nombres, de ponctuation ou de caractères spéciaux dans la valeur d'un `id` ou d'une `class` peut causer des problèmes dans d'autres contextes, tels que CSS, JavaScript et les expressions régulières.

Par exemple, bien que l' `id` suivant soit valide en HTML5:

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

... il est invalide en CSS:

4.1.3 Caractères et cas

> En CSS, les *identificateurs* (y compris les noms d'éléments, les classes et les identifiants dans les sélecteurs) ne peuvent contenir que les caractères [a-zA-Z0-9] et les caractères ISO 10646 U + 00A0 et supérieur, plus le tiret _); ***ils ne peuvent pas commencer par un chiffre, deux tirets ou un tiret suivi d'un chiffre*** . (soulignement ajouté)

Dans la plupart des cas, vous pouvez échapper à des caractères dans des contextes où ils ont des restrictions ou une signification particulière.

Références W3C

- 3.2.5.1 L'attribut `id`
- 3.2.5.7 L'attribut de `class`
- 6.2 Types de base SGML

Lire Classes et identifiants en ligne: https://riptutorial.com/fr/html/topic/586/classes-et-identifiants

Chapitre 11: Commandes du menu de sélection

Syntaxe

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

Examples

Sélectionnez le menu

L'élément `<select>` génère un menu déroulant à partir duquel l'utilisateur peut choisir une option.

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

Changer la taille

Vous pouvez modifier la taille du menu de sélection avec l'attribut `size` . Une taille de 0 ou 1 affiche le menu de style déroulant standard. Une taille supérieure à 1 convertira la liste déroulante en une boîte affichant autant de lignes, avec une option par ligne et une barre de défilement afin de faire défiler les options disponibles.

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

Menus de sélection multi-options

Par défaut, les utilisateurs ne peuvent sélectionner qu'une seule option. L'ajout de l'attribut `multiple` permet aux utilisateurs de sélectionner plusieurs options à la fois et de soumettre toutes les options sélectionnées au formulaire. L'utilisation de l'attribut `multiple` convertit automatiquement le menu déroulant en une zone comme si sa taille était définie. La taille par défaut lorsque cela se produit est déterminée par le navigateur que vous utilisez, et il est impossible de la rétablir dans un menu de style déroulant tout en autorisant plusieurs sélections.

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

Lorsque vous utilisez l'attribut `multiple` , il existe une différence entre 0 et 1 pour la taille, alors

qu'aucune différence n'existe lorsque vous n'utilisez pas l'attribut. Si vous utilisez 0, le navigateur se comportera de la manière programmée par défaut. L'utilisation de 1 définit explicitement la taille de la boîte résultante sur une seule ligne de haut.

Groupes d'options

Vous pouvez facilement regrouper vos options dans un menu de sélection afin de fournir une mise en page plus structurée dans une longue liste d'options à l'aide de l'élément `<optgroup>` .

La syntaxe est très simple, en utilisant simplement l'élément avec un attribut `label` pour identifier le titre du groupe et en contenant zéro ou plusieurs options qui doivent figurer dans ce groupe.

```
<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>
```

Lorsque vous utilisez des groupes d'options, toutes les options ne doivent pas être contenues dans un groupe. De même, la désactivation d'un groupe d'options désactivera toutes les options du groupe et il n'est pas possible de réactiver manuellement une seule option dans un groupe désactivé.

Les options

Les options à l'intérieur d'un menu de sélection correspondent à la sélection de l'utilisateur. La syntaxe normale d'une option est la suivante:

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

Cependant, il est important de noter que le texte à l'intérieur de l'élément `<option>` lui-même n'est pas toujours utilisé et devient essentiellement la valeur par défaut des attributs qui ne sont pas spécifiés.

Les attributs qui contrôlent l'apparence et la fonction de l'option sont `value` et `label` . L'étiquette représente le texte qui sera affiché dans le menu déroulant (ce que vous regardez et cliquez sur pour le sélectionner). La valeur représente le texte qui sera envoyé avec l'envoi du formulaire. Si l'une de ces valeurs est omise, elle utilise le texte à l'intérieur de l'élément à la place. Ainsi, l'exemple que nous avons donné ci-dessus pourrait être "élargi" à ceci:

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

Notez l'omission du texte interne et de la balise de fin, qui ne sont pas nécessaires pour créer une

option dans le menu. S'ils étaient inclus, le texte intérieur serait ignoré car les deux attributs sont déjà spécifiés et le texte n'est pas nécessaire. Cependant, vous ne verrez probablement pas beaucoup de gens les écrire de cette façon. La façon la plus courante d'écrire est d'utiliser une valeur qui sera envoyée au serveur, avec le texte intérieur qui deviendra éventuellement l'attribut label, comme ceci:

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

Sélection d'une option par défaut

Vous pouvez également spécifier une certaine option à sélectionner dans le menu par défaut en lui associant l'attribut `selected` . Par défaut, si aucune option n'est spécifiée dans le menu, la première option du menu sera sélectionnée lors du rendu. Si l'attribut `selected` associé à plusieurs options, la dernière option présente dans le menu avec l'attribut sera celle sélectionnée par défaut.

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

Si vous utilisez l'attribut dans un menu de sélection multi-options, toutes les options avec l'attribut seront sélectionnées par défaut et aucune ne sera sélectionnée si aucune option ne possède l'attribut.

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

Datalist

La `<datalist>` spécifie une liste d'options prédéfinies pour un élément `<input>` . Il fournit une fonctionnalité "autocomplete" sur `<input>` éléments `<input>` . Les utilisateurs verront une liste déroulante d'options à mesure qu'ils écrivent.

```
<input list="Languages">

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

Prise en charge du navigateur

Chrome	Bord	Mozilla	Safari	Opéra
20.0	10.0	4.0	Non supporté	9.0

Lire Commandes du menu de sélection en ligne:
https://riptutorial.com/fr/html/topic/722/commandes-du-menu-de-selection

Chapitre 12: commentaires

Introduction

Semblables aux autres langages de programmation, de balisage et de démarquage, les commentaires en HTML fournissent aux autres développeurs des informations spécifiques au développement sans affecter l'interface utilisateur. Contrairement aux autres langages, les commentaires HTML peuvent être utilisés pour spécifier des éléments HTML uniquement pour Internet Explorer. Cette rubrique explique comment écrire des commentaires HTML et leurs applications fonctionnelles.

Syntaxe

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

Remarques

Tout ce qui commence par `<!--` et se termine par `-->` est un commentaire. Les commentaires ne peuvent pas contenir deux tirets adjacents (`--`) et doivent se terminer par deux tirets exactement (c.-à-d. `--->` n'est pas correct).

Les commentaires ne sont pas visibles sur une page Web et ne peuvent pas être stylés avec CSS. Ils peuvent être utilisés par le développeur de la page pour prendre des notes dans le code HTML ou pour masquer certains contenus lors du développement.

Pour les pages dynamiques ou interactives, masquer et afficher du contenu se fait avec JavaScript et CSS plutôt qu'avec des commentaires HTML.

JavaScript peut être utilisé pour obtenir le contenu des nœuds de commentaire HTML et ces nœuds peuvent être créés, ajoutés et supprimés dynamiquement du document, mais cela n'affecte pas l'affichage de la page.

Les commentaires HTML faisant partie du code source de la page, ils sont téléchargés dans le navigateur avec le reste de la page. Le code source peut généralement être visualisé à l'aide de l'option de menu du navigateur Web: "Afficher la source" ou "Afficher la source de la page".

Examples

Créer des commentaires

Les commentaires HTML peuvent être utilisés pour laisser des notes à vous-même ou à d'autres développeurs sur un point spécifique du code. Ils peuvent être initiés avec `<!--` et conclus avec `-->` , comme ceci:

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

Ils peuvent être intégrés en ligne dans d'autres contenus:

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

Ils peuvent également couvrir plusieurs lignes pour fournir plus d'informations:

```
<!-- 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.
-->
```

Cependant, ils **ne peuvent pas** apparaître dans une autre balise HTML, comme ceci:

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

Ce produit HTML invalide l'ensemble `<h1 <!-- testAttribute="something" -->` bloc serait considéré comme une seule balise de début `h1` avec d'autres informations non valides qu'il contient, suivi d'un seul `>` support de fermeture qui ne fait rien.

Pour des raisons de compatibilité avec les outils qui tentent d'analyser HTML au format XML ou SGML, le corps de votre commentaire ne doit pas contenir deux tirets `--` .

Commentaires conditionnels pour Internet Explorer

Les commentaires conditionnels peuvent être utilisés pour personnaliser le code de différentes versions de Microsoft Internet Explorer. Par exemple, différentes classes HTML, balises de script ou feuilles de style peuvent être fournies. Les commentaires conditionnels sont pris en charge dans les versions 5 à 9 d'Internet Explorer. Les versions Internet Explorer plus anciennes et plus récentes, ainsi que tous les navigateurs autres qu'IE, sont considérés comme "de bas niveau" et traitent les commentaires conditionnels comme des commentaires HTML ordinaires.

Caché de bas

Les commentaires cachés au niveau inférieur fonctionnent en encapsulant l'intégralité du contenu dans ce qui semble être un commentaire HTML normal. Seuls les IE 5 à 9 le liront toujours comme un commentaire conditionnel et ils masqueront ou afficheront le contenu en conséquence. Dans d'autres navigateurs, le contenu sera caché.

```
<!--[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.
-->
```

Niveau inférieur révélé

Celles-ci sont légèrement différentes des commentaires cachés au niveau inférieur: seul le commentaire conditionnel lui-même est contenu dans la syntaxe de commentaire normale. Les navigateurs qui ne prennent pas en charge les commentaires conditionnels les ignorent simplement et affichent le reste du contenu entre eux.

```
<!--[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]-->
```

Commenter les espaces entre les éléments en ligne

Les éléments d'affichage en ligne, généralement tels que `span` ou `a` , incluront jusqu'à un caractère blanc avant et après le document. Pour éviter les lignes très longues dans le balisage (difficiles à lire) et les espaces blancs involontaires (qui affectent le formatage), l'espace blanc peut être mis en commentaire.

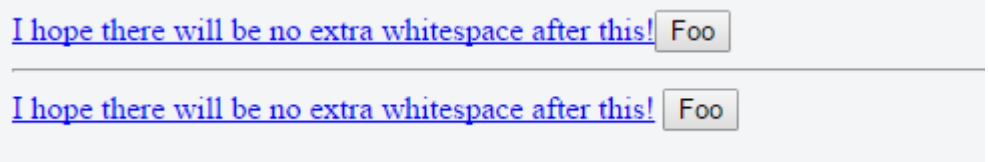
```
<!-- Use an HTML comment to nullify the newline character below: -->
<a href="#">I hope there will be no extra whitespace after this!</a><!--
--><button>Foo</button>
```

Essayez-le sans commentaire entre les éléments en ligne, et il y aura un espace entre eux. Parfois, ramasser le caractère d'espace est désiré.

Exemple de code:

```
<!-- 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>
```

Sortie:



Lire commentaires en ligne: https://riptutorial.com/fr/html/topic/468/commentaires

Chapitre 13: Des listes

Introduction

HTML offre trois méthodes pour spécifier des listes: les listes ordonnées, les listes non ordonnées et les listes de description. Les listes ordonnées utilisent des séquences ordinales pour indiquer l'ordre des éléments de liste, les listes non ordonnées utilisent un symbole défini tel qu'une puce pour répertorier les éléments dans aucun ordre désigné et les listes de description utilisent des indentations pour répertorier les éléments avec leurs enfants. Cette rubrique explique l'implémentation et la combinaison de ces listes dans le balisage HTML.

Syntaxe

- `<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>`

Remarques

Voir également

Vous pouvez ajouter une propriété CSS de type liste-style à une `<ul>` afin de modifier le type d'icône utilisé pour marquer chaque élément de liste, par exemple `<ul style="list-style-type:disc">` . Les types de liste suivants sont autorisés:

- "list-style-type: disc" est le point par défaut.
- "list-style-type: circle" est un cercle non rempli.
- "list-style-type: square" est un carré rempli.
- "list-style-type: none" n'utilise aucune marque.

Vous pouvez également ajouter un attribut type à une `<ol>` afin de modifier la numérotation, par exemple `<ol type="1">` . Les types suivants sont autorisés:

- type = "1" est le formulaire par défaut.
- type = "A" utilise des lettres majuscules dans l'ordre alphabétique
- type = "a" utilise des lettres minuscules dans l'ordre alphabétique
- type = "I" utilise des chiffres romains avec des lettres majuscules
- type = "i" utilise des chiffres romains avec des lettres minuscules

Examples

Liste non ordonnée

Une liste non ordonnée peut être créée avec la `<ul>` et chaque élément de la liste peut être créé avec la `<li>` comme le montre l'exemple ci-dessous:

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

Cela produira une liste à puces (qui est le style par défaut):

- Article
- Un autre article
- Encore un autre article

Vous devez utiliser `ul` pour afficher une liste d'éléments, où l'ordre des éléments n'est pas important. Si la modification de l'ordre des éléments rend la liste incorrecte, vous devez utiliser `<ol>` .

Liste ordonnée

Une liste ordonnée peut être créée avec la `<ol>` et chaque élément de la liste peut être créé avec la `<li>` comme dans l'exemple ci-dessous:

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

Cela produira une liste numérotée (qui est le style par défaut):

1. Article
2. Un autre article
3. Encore un autre article

Changer manuellement les numéros

Il y a plusieurs façons de jouer avec les numéros qui apparaissent sur les éléments d'une liste ordonnée. La première consiste à définir un numéro de départ en utilisant l'attribut `start` . La liste commencera à ce numéro défini et continuera à incrémenter de un comme d'habitude.

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

Cela produira une liste numérotée (qui est le style par défaut):

3. Article
4. Un autre article
5. Encore un autre article

Vous pouvez également définir explicitement un certain élément de liste sur un numéro spécifique. Les autres éléments de liste après l'un avec une valeur spécifiée continueront à incrémenter d'une valeur de cet élément de la liste, en ignorant où se trouvait la liste parente.

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

Il est également intéressant de noter qu'en utilisant l'attribut `value` directement sur un élément de la liste, vous pouvez remplacer le système de numérotation existant d'une liste ordonnée en redémarrant la numérotation à une valeur inférieure. Donc, si la liste parente était déjà à la valeur 7 et rencontrait un élément de liste à la valeur 4, cet élément de liste afficherait toujours 4 et continuerait à compter à partir de ce point.

```
<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>
```

Ainsi, l'exemple ci-dessus produira une liste qui suit le schéma de numérotation de 5, 6, 4, 5, 6 - recommençant à un nombre inférieur au précédent et dupliquant le nombre 6 dans la liste.

Remarque: les attributs de `start` et de `value` n'acceptent qu'un nombre - même si la liste ordonnée est définie pour s'afficher sous forme de chiffres romains ou de lettres.

5

Vous pouvez inverser la numérotation en ajoutant `reversed` dans votre élément `ol` :

```
<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 numérotation inversée est utile si vous ajoutez continuellement à une liste, par exemple avec de nouveaux épisodes ou présentations de podcast, et que vous souhaitez que les éléments les plus récents apparaissent en premier.

Changer le type de chiffre

Vous pouvez facilement changer le type de chiffre indiqué dans le marqueur d'élément de liste en

utilisant l'attribut `type`

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

Type	La description	Exemples
`1`	Valeur par défaut - Nombre décimal	1,2,3,4
`a`	Ordonné alphabétiquement (minuscule)	a B c d
`A`	Ordonné alphabétiquement (majuscule)	A B C D
`i`	Chiffres romains (minuscules)	i, ii, iii, iv
`I`	Chiffres romains (majuscules)	I, II, III, IV

> Vous devez utiliser `ol` pour afficher une liste d'éléments, où les articles ont été intentionnellement ordonnés et la commande doit être soulignée. Si la modification de l'ordre des éléments ne rend pas la liste incorrecte, vous devez utiliser `<ul>` .

Liste de description

Une liste de description (ou *une liste de définitions* , comme on l'appelait avant HTML5) peut être créée avec l'élément `dl` . Il se compose de groupes nom-valeur, où le nom est donné dans l'élément `dt` et la valeur est donnée dans l'élément `dd` .

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

Démo en direct

Un groupe nom-valeur peut avoir plusieurs noms et / ou plusieurs valeurs (qui représentent des alternatives):

```
<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>
```

Démo en direct

Listes imbriquées

Vous pouvez imbriquer des listes pour représenter des sous-éléments d'un élément de liste.

```
<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>
```

- objet 1
- article 2
    - sous-point 2.1
    - sous-point 2.2
- article 3

La liste imbriquée doit être un enfant de l'élément `li` .

Vous pouvez également imbriquer différents types de liste:

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

Lire Des listes en ligne: https://riptutorial.com/fr/html/topic/393/des-listes

Chapitre 14: Doctypes

Introduction

Doctypes - abréviation de «type de document» - aide les navigateurs à comprendre la version de HTML dans laquelle le document est écrit pour une meilleure interprétation. Les déclarations Doctype ne sont pas des balises HTML et se trouvent tout en haut d'un document. Cette rubrique explique la structure et la déclaration de divers types de code en HTML.

Syntaxe

- <! DOCTYPE [chaîne spécifique à la version]>

Remarques

La déclaration `<!DOCTYPE>` n'est pas une balise HTML. Il est utilisé pour spécifier la version de HTML utilisée par le document. C'est ce que l'on appelle la déclaration de type de document (DTD).

La déclaration `<!DOCTYPE>` n'est PAS sensible à la casse. Pour vérifier si le code HTML de vos pages Web est valide, accédez au service de validation du W3C .

- Certaines anciennes versions d'IE ne prennent pas en charge certaines balises HTML, à moins qu'un doctype approprié ne soit disponible.
- Il est *essentiel* qu'un type de document soit déclaré pour s'assurer que le navigateur n'utilise pas le mode quirks. Plus d'infos sur MDN.

Examples

Ajouter le Doctype

La déclaration `<!DOCTYPE>` doit toujours figurer en haut du document HTML, avant la `<html>` .

5

Voir HTML 5 Doctype pour plus de détails sur le HTML 5 Doctype.

```
<!DOCTYPE html>
```

4.01

Voir HTML 4.01 Doctypes pour plus de détails sur la différence entre ces types.

Strict

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

De transition

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

Jeu de cadres

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

HTML 4.01 Doctypes

La spécification HTML 4.01 fournit plusieurs types de doctypes permettant de spécifier différents types d'éléments dans le document.

HTML 4.01 Strict

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

Inclut tous les éléments et attributs HTML, mais **n'inclut pas les éléments de présentation ou obsolètes** et les **jeux de cadres ne sont pas autorisés** .

HTML 4.01 Transitional

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

Inclut tous les éléments et attributs HTML et éléments de présentation et obsolètes, mais les **jeux de cadres ne sont pas autorisés** .

Jeu de cadres HTML 4.01

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

Inclut tous les éléments et attributs HTML, les éléments de présentation et les éléments obsolètes. Les jeux de cadres sont autorisés.

HTML 5 Doctype

HTML5 n'est pas basé sur SGML et ne nécessite donc pas de référence à une DTD.

HTML 5 Déclaration Doctype:

```
<!DOCTYPE html>
```

Insensibilité à la casse

Par le W3.org HTML 5 DOCTYPE Spec :

> Un DOCTYPE doit comprendre les composants suivants, dans cet ordre:
>
> 1. Chaîne correspondant à une chaîne ASCII **insensible** à la **casse** pour la chaîne `"<!DOCTYPE"`.

par conséquent, les DOCTYPE suivants sont également valides:

```
<!doctype html>
<!dOCtyPe html>
<!DocTYpe html>
```

Cet article SO aborde le sujet en profondeur: Doctype majuscule ou minuscule?

Vieux Doctypes

HTML 3.2

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

HTML 3.2 est bien supporté par la plupart des navigateurs utilisés. Cependant, HTML 3.2 prend en charge les feuilles de style et ne prend pas en charge les fonctionnalités HTML 4 telles que les cadres et l'internationalisation.

HTML 2.0

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

HTML 2.0 est largement pris en charge par les navigateurs mais ne prend pas en charge les tableaux, les cadres et l'internationalisation, ainsi que de nombreux éléments et attributs de présentation couramment utilisés.

Lire Doctypes en ligne: https://riptutorial.com/fr/html/topic/806/doctypes

Chapitre 15: Élément d'étiquette

Syntaxe

- `<label>Example <input type="radio" name="r"></label>` // Envelopper un élément de contrôle
- `<label for="rad1">Example</label> <input id="rad1" type="radio" name="r">` // Utilisation `for` attribut

Paramètres

Les attributs	La description
pour	Référence à l'élément d'identification cible. Ie: `for="surname"`
forme	**HTML5** , **[Obsolète]** Référence au formulaire contenant l'élément cible. Les éléments d'étiquette sont attendus dans un élément `<form>` . Si le `form="someFormId"` est fourni, cela vous permet de placer l'étiquette n'importe où dans le document.

Examples

Utilisation de base

Formulaire simple avec étiquettes ...

```
<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 propos de l'étiquette

L'élément `<label>` est utilisé pour référencer un élément d'action de formulaire.
Dans le cadre de **l'interface utilisateur,** il est utilisé pour faciliter la sélection des éléments tels que Type `radio` ou `checkbox` .

`<label>` comme wrapper

Il peut contenir l'élément d'action souhaité

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

(En cliquant sur le texte, l' `input` cible basculera son état / valeur)

`<label>` comme référence

En utilisant l'attribut `for` vous n'avez pas besoin de placer l'élément de contrôle en tant que descendant de `label` - mais la valeur `for` doit correspondre à son identifiant

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

Remarque
N'utilisez pas plus d'un élément de contrôle dans un élément `<label>`

Lire Élément d'étiquette en ligne: https://riptutorial.com/fr/html/topic/1704/element-d-etiquette

Chapitre 16: Élément de progrès

Paramètres

Paramètre	Valeur
max	Combien de travail la tâche exige au total
valeur	Combien de travail a déjà été accompli
position	Cet attribut renvoie la position actuelle de l'élément `<progress>`
Étiquettes	Cet attribut renvoie une liste d'étiquettes d'élément `<progress>` (le cas échéant)

Remarques

L'élément `<progress>` n'est pas pris en charge dans les versions d'Internet Explorer inférieures à 10

L'élément `<progress>` est le mauvais élément à utiliser pour quelque chose qui n'est qu'une jauge, plutôt que la progression de la tâche. Par exemple, l'affichage de l'utilisation de l'espace disque à l'aide de l'élément `<progress>` est inapproprié. Au lieu de cela, l'élément `<meter>` est disponible pour ce type de cas d'utilisation.

Examples

Le progrès

L'élément `<progress>` est nouveau en HTML5 et sert à représenter la progression d'une tâche

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

Cela crée une barre remplie 22%

Changer la couleur d'une barre de progression

Les barres de progression peuvent être stylées avec le sélecteur de `progress[value]` .

Cet exemple donne une barre de progression d'une largeur de `250px` et d'une hauteur de `20px`

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

Les barres de progression peuvent être particulièrement difficiles à mettre en forme.

Chrome / Safari / Opera

Ces navigateurs utilisent le sélecteur d' `-webkit-appearance` pour personnaliser la balise de progression. Pour annuler cela, nous pouvons réinitialiser l'apparence.

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

Maintenant, nous pouvons styliser le conteneur lui-même

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

Firefox

Firefox classe la barre de progression un peu différemment. Nous devons utiliser ces styles

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

Internet Explorer

Internet Explorer 10+ prend en charge l'élément de `progress` . Cependant, il ne prend pas en charge la propriété `background-color` . Vous devrez utiliser la propriété color à la place.

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

  border: none;                        /* Remove border from Firefox */

  width: 250px;
  height: 20px;

  color: blue;
}
```

HTML de repli

Pour les navigateurs qui ne prennent pas en charge l'élément de `progress` , vous pouvez l'utiliser comme solution de contournement.

```
<progress max="100" value="20">
```

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

Les navigateurs qui prennent en charge la balise de progression ignorent le div imbriqué à l'intérieur. Les navigateurs existants qui ne peuvent pas identifier la balise de progression afficheront le div à la place.

Lire Élément de progrès en ligne: https://riptutorial.com/fr/html/topic/5055/element-de-progres

Chapitre 17: Élément de sortie

Paramètres

Attribut	La description
Global	Attributs disponibles pour tout élément HTML5. Pour une documentation complète de ces attributs, voir: Attributs globaux MDN
prénom	Une chaîne représentant le nom d'une sortie. En tant qu'élément de formulaire, la sortie peut être référencée par son nom à l'aide de la propriété `document.forms` . Cet attribut est également utilisé pour collecter des valeurs sur une soumission de formulaire.
pour	Une liste d'espaces d'élément de formulaire séparés par des espaces (par exemple, `<inputs id="inp1"> for value is "inp1"`) que la sortie est censée afficher.
forme	Une chaîne représentant le `<form>` associé à la sortie. Si la sortie est en fait en dehors de `<form>` , cet attribut garantira que la sortie appartient toujours au `<form>` et est soumise aux collectes et aux soumissions dudit `<form>` .

Examples

Elément de sortie utilisant les attributs For et Form

La démonstration suivante utilise un élément `<output>` utilisant les attributs `[for]` et `[form]` . Gardez à l'esprit que `<output>` **besoin de JavaScript** pour fonctionner. Le code JavaScript intégré est couramment utilisé dans les formulaires, comme le montre cet exemple. Bien que les éléments `<input>` soient `type="number"` , leur `value` s ne sont pas des nombres, mais du texte. Donc, si vous voulez que la `value` s soit calculée, vous devez convertir chaque `value` en un nombre en utilisant des méthodes telles que: `parseInt()` , `parseFloat()` , `Number()` , etc.

Démo en direct

```
    <!--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>
```

Elément de sortie avec attributs

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

Lire Élément de sortie en ligne: https://riptutorial.com/fr/html/topic/723/element-de-sortie

Chapitre 18: Élément div

Introduction

L'élément div dans HTML est un élément conteneur qui encapsule d'autres éléments et peut être utilisé pour regrouper et séparer des parties d'une page Web. Une div par elle-même ne représente en soi rien mais est un outil puissant dans la conception de sites Web. Ce sujet couvre le but et les applications de l'élément div.

Syntaxe

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

Examples

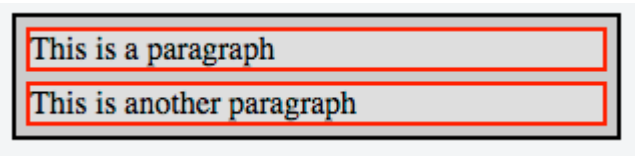
Nidification

Il est courant de placer plusieurs `<div>` dans un autre `<div>` . Ceci est généralement appelé éléments "d'imbrication" et permet de diviser davantage les éléments en sous-sections ou d'aider les développeurs avec un style CSS.

Le `<div class="outer-div">` est utilisé pour regrouper deux éléments `<div class="inner-div">` ; chacun contenant un élément `<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>
```

Cela donnera le résultat suivant (styles CSS appliqués pour plus de clarté):



Imbrication d'éléments en ligne et de blocs Lors de l'imbrication d'éléments, vous devez garder à l'esprit qu'il existe des éléments en ligne et des blocs. tandis que les éléments de bloc "ajoutent un saut de ligne en arrière-plan", ce qui signifie que les autres éléments imbriqués sont affichés automatiquement dans la ligne suivante, les éléments en ligne peuvent être positionnés les uns à côté des autres par défaut

Évitez les imbrications `<div>` profondes

Un format de conteneur imbriqué profond et souvent utilisé montre un style de codage incorrect.

Les coins arrondis ou certaines fonctions similaires créent souvent un tel code HTML. Pour la plupart des navigateurs de dernière génération, il existe des contreparties CSS3. Essayez d'utiliser le moins possible d'éléments HTML pour augmenter le rapport entre le contenu et les balises et réduire le chargement de la page, ce qui se traduit par un meilleur classement dans les moteurs de recherche.

`div` section L'élément ne doit pas être plus imbriqué que 6 couches.

Utilisation de base

L'élément `<div>` n'a généralement aucune signification sémantique spécifique, représentant simplement une division, et est généralement utilisé pour regrouper et encapsuler d'autres éléments dans un document HTML et pour séparer ceux des autres groupes de contenu. En tant que tel, chaque `<div>` est mieux décrit par son contenu.

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

L'élément `div` est généralement un élément de niveau bloc , ce qui signifie qu'il sépare un bloc d'un document HTML et occupe la largeur maximale de la page. Les navigateurs ont généralement la règle CSS par défaut suivante:

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

> **Le Consortium World Wide Web (W3C)** encourage vivement l'élément div comme élément de dernier recours, lorsque aucun autre élément ne convient. L'utilisation d'éléments plus appropriés au lieu de l'élément div entraîne une meilleure accessibilité pour les lecteurs et une facilité de maintenance pour les auteurs.

Par exemple, un article de blog serait marqué en utilisant `<article>` , un chapitre utilisant `<section>` , des aides à la navigation d'une page utilisant `<nav>` et un groupe de contrôles de formulaire utilisant `<fieldset>` .

Les éléments div peuvent être utiles à des fins stylistiques ou pour envelopper plusieurs paragraphes dans une section qui doivent tous être annotés de manière similaire.

Lire Élément div en ligne: https://riptutorial.com/fr/html/topic/1468/element-div

Chapitre 19: Éléments de contrôle d'entrée

Introduction

Composant essentiel des systèmes Web interactifs, les balises d'entrée sont des éléments HTML conçus pour prendre en charge une forme d'information spécifique des utilisateurs. Différents types d'éléments de saisie peuvent réguler les données saisies pour s'adapter à un format spécifié et assurer la sécurité de la saisie du mot de passe.

Syntaxe

- `<input type="" name="" value="">`

Paramètres

Paramètre	Détails
classe	Indique la classe de l'entrée
id	Indique l'ID de l'entrée
type	Identifie le type de contrôle d'entrée à afficher. Les valeurs acceptables sont `hidden` , `text` , `tel` , `url` , `email` , `password` , `date` , `time` , `number` , `range` , `color` , `checkbox` , `radio` , `file` , `submit` , `image` , `reset` et `button` . Par défaut, le `text` n'est pas spécifié, si la valeur n'est pas valide ou si le navigateur ne prend pas en charge le type spécifié.
prénom	Indique le nom de l'entrée
désactivée	Valeur booléenne indiquant que l'entrée doit être désactivée. Les contrôles désactivés ne peuvent pas être modifiés, ne sont pas envoyés lors de l'envoi du formulaire et ne peuvent pas recevoir le focus.
vérifié	Lorsque la valeur de l'attribut type est radio ou case à cocher, la présence de cet attribut booléen indique que le contrôle est sélectionné par défaut. sinon il est ignoré.
plusieurs	**HTML5** Indique que plusieurs fichiers ou valeurs peuvent être transmis (s'applique uniquement aux entrées de type `file` et `email`)
espace réservé	**HTML5** Un indice pour l'utilisateur de ce qui peut être saisi dans le contrôle. Le texte d'espace réservé ne doit pas contenir de retour chariot ou de saut de ligne
autocomplete	**HTML5** Indique si la valeur du contrôle peut être complétée automatiquement par le navigateur.

Paramètre	Détails
lecture seulement	Valeur booléenne indiquant que l'entrée n'est pas modifiable. Les commandes en lecture seule sont toujours envoyées lors de la soumission du formulaire, mais ne reçoivent pas le focus. **HTML5:** cet attribut est ignoré lorsque la valeur de l'attribut `type` est définie sur `hidden` , `range` , `color` , `checkbox` , `radio` , `file` ou `button` .
Champs obligatoires	**HTML5** Indique qu'une valeur doit être présente ou que l'élément doit être vérifié pour que le formulaire soit soumis
alt	Un texte alternatif pour les images, au cas où elles ne seraient pas affichées.
autofocus	L'élément `<input>` devrait avoir le focus lors du chargement de la page.
valeur	Spécifie la valeur de l'élément `<input>` .
étape	L'attribut `step` spécifie les intervalles de numéros légaux. Il fonctionne avec les types d'entrées suivants: `number` , `range` , `date` , `date` - `date datetime-local` , `month` , `time` et `week` .

Remarques

Comme pour les autres éléments vides de HTML5, `<input>` se ferme automatiquement et peut être écrit `<input />` . HTML5 ne nécessite pas cette barre oblique.

Les types d'entrées valides suivants sont en HTML:

- `button`
- `checkbox`
- `file`
- `hidden`
- `image`
- `password`
- `radio`
- `reset`
- `submit`
- `text` (valeur par défaut)

5

Les suivants sont des types d'entrée nouvellement introduits dans le standard HTML 5. Certains de ces types ne sont pas pris en charge par tous les navigateurs Web. Dans le cas où un type n'est pas pris en charge, l'élément d'entrée sera par défaut le type de `text` .

- `color`
- `date`
- `datetime` / heure (obsolète et obsolète)
- `datetime-local`
- `email`

- month
- number
- range
- search
- tel
- time
- url
- week

Pour vérifier quels navigateurs prennent en charge quels types, vous pouvez aller sur caniuse.com .

Examples

Case à cocher et boutons radio

Vue d'ensemble

Les cases à cocher et les boutons radio sont écrits avec la balise HTML `<input>` et leur comportement est défini dans la spécification HTML .

La case à cocher ou le bouton radio le plus simple est un élément `<input>` avec un attribut de `type` respectivement `checkbox` ou `radio` :

```
<input type="checkbox">
<input type="radio">
```

Un seul élément de case à cocher autonome est utilisé pour une seule option binaire telle qu'une question oui ou non. Les cases à cocher sont indépendantes, ce qui signifie que l'utilisateur peut sélectionner autant de choix qu'il le souhaite dans un groupe de cases à cocher. En d'autres termes, cocher une case ne désélectionne *pas* les autres cases à cocher du groupe de cases à cocher.

Les boutons radio viennent généralement en groupes (si elle n'est pas groupée avec un autre bouton radio, vous vouliez probablement utiliser une case à cocher) identifiée en utilisant le même attribut de `name` sur tous les boutons de ce groupe. La sélection des boutons radio est *mutuellement exclusive* , ce qui signifie que l'utilisateur ne peut sélectionner qu'un seul choix parmi un groupe de boutons radio. Lorsqu'un bouton radio est coché, tout autre bouton radio portant le même `name` que précédemment coché devient décoché.

Exemple:

```
<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">
```

Lorsqu'ils sont affichés, les boutons radio apparaissent sous la forme d'un cercle (non coché) ou d'un cercle rempli (coché). Les cases à cocher apparaissent sous la forme d’un carré (non coché)

ou d’un carré rempli (coché). Selon le navigateur et le système d'exploitation, le carré a parfois des coins arrondis.

Les attributs

Les cases à cocher et les boutons radio ont plusieurs attributs pour contrôler leur comportement:

`value`

Comme tout autre élément d'entrée, l'attribut `value` spécifie la valeur de chaîne à associer au bouton en cas de soumission de formulaire. Cependant, les cases à cocher et des boutons radio sont spéciaux en ce que lorsque la valeur est omis, par défaut `on` lorsqu'il est soumis, plutôt que d' envoyer une valeur vide. L'attribut de `value` n'est pas reflété dans l'apparence du bouton.

`checked`

L'attribut `checked` spécifie l'état initial d'une case à cocher ou d'un bouton radio. Ceci est un attribut booléen et peut être omis.

Chacune de ces méthodes est valide et équivalente pour définir un bouton radio vérifié:

```
<input checked>
<input checked="">
<input checked="checked">
<input checked="ChEcKeD">
```

L'absence de l'attribut `checked` est la seule syntaxe valide pour un bouton non `checked` :

```
<input type="radio">
<input type="checkbox">
```

Lors de la réinitialisation d'un `<form>` , les cases à cocher et les boutons radio reviennent à l'état de leur attribut `checked` .

Accessibilité

Étiquettes

Pour donner un contexte aux boutons et montrer aux utilisateurs à quoi sert chaque bouton, chacun d’eux doit avoir une étiquette. Cela peut être fait en utilisant un élément `<label>` pour envelopper le bouton. De plus, cela permet de cliquer sur l'étiquette, de sorte que vous sélectionnez le bouton correspondant.

Exemple:

```
<label>
```

```
  <input type="radio" name="color" value="#F00">
  Red
</label>
```

ou avec un élément `<label>` avec un attribut `for` défini sur l'attribut `id` du bouton:

```
<input type="checkbox" name="color" value="#F00" id="red">
<label for="red">Red</label>
```

Groupes de boutons

Étant donné que chaque bouton radio affecte les autres dans le groupe, il est courant de fournir une étiquette ou un contexte pour l'ensemble du groupe de boutons radio.

Pour fournir une étiquette pour le groupe entier, les boutons radio doivent être inclus dans un élément `<fieldset>` avec un élément `<legend>` .

Exemple:

```
<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>
```

Les cases à cocher peuvent également être regroupées de manière similaire, avec un champ et une légende identifiant le groupe de cases à cocher correspondantes. Cependant, gardez à l'esprit que les cases à cocher *ne* doivent *pas* partager le même nom car elles ne s'excluent pas mutuellement. Si cela se produit, le formulaire soumet plusieurs valeurs pour la même clé et tous les langages côté serveur ne gèrent pas cela de la même manière (comportement non défini). Chaque case à cocher doit avoir un nom unique ou utiliser un ensemble de crochets (`[]`) pour indiquer que le formulaire doit soumettre un tableau de valeurs pour cette clé. La méthode que vous choisissez doit dépendre de la manière dont vous prévoyez de gérer les données de formulaire côté client ou côté serveur. Vous devez également garder la légende courte, car certaines combinaisons de navigateurs et de lecteurs d'écran lisent la légende avant chaque champ de saisie dans le jeu de champs.

Caché

```
<input type="hidden" name="inputName" value="inputValue">
```

Une entrée masquée ne sera pas visible pour l'utilisateur, mais sa valeur sera envoyée au serveur lorsque le formulaire est néanmoins soumis.

Mot de passe

```
<input type="password" name="password">
```

L'élément input avec un attribut type dont la valeur est `password` crée un champ de texte à une seule ligne similaire au `type=text` entrée `type=text` , sauf que le texte n'est pas affiché lorsque l'utilisateur le saisit.

```
<input type="password" name="password" placeholder="Password">
```

Le texte d'espace réservé est affiché en texte brut et est automatiquement remplacé lorsqu'un utilisateur commence à taper.

Remarque: Certains navigateurs et systèmes modifient le comportement par défaut du champ de mot de passe pour afficher également le dernier caractère saisi pendant une courte durée, comme ceci:

Soumettre

```
<input type="submit" value="Submit">
```

Une entrée de soumission crée un bouton qui soumet le formulaire dans lequel il se trouve lorsqu'il est cliqué.

Vous pouvez également utiliser l'élément `<button>` si vous avez besoin d'un bouton de soumission qui peut être plus facilement stylé ou contenir d'autres éléments:

```
<button type="submit">
  <img src="submit-icon.jpg" /> Submit
</button>
```

Fichier

```
<input type="file" name="fileSubmission">
```

Les entrées de fichier permettent aux utilisateurs de sélectionner un fichier dans leur système de fichiers local pour l'utiliser avec la page en cours. S'ils sont utilisés avec un élément de `form` , ils peuvent être utilisés pour permettre aux utilisateurs de télécharger des fichiers sur un serveur

(pour plus d'informations, voir Téléchargement de fichiers).

L'exemple suivant permet aux utilisateurs d'utiliser l'entrée de `file` pour sélectionner un fichier dans leur système de fichiers et télécharger ce fichier sur un script sur le serveur nommé `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>
```

Plusieurs fichiers

L'ajout de l'attribut `multiple` permettra à l'utilisateur de sélectionner **plusieurs** fichiers:

```
<input type="file" name="fileSubmission" id="fileSubmission" multiple>
```

Accepter les fichiers

L'attribut d'acceptation spécifie les types de fichiers que l'utilisateur peut sélectionner. Par exemple, `.png` , `.gif` , `.jpeg` .

```
<input type="file" name="fileSubmission" accept="image/x-png,image/gif,image/jpeg" />
```

Validation des entrées

La validation des entrées HTML est effectuée automatiquement par le navigateur en fonction des attributs spéciaux de l'élément d'entrée. Il pourrait remplacer partiellement ou complètement la validation des entrées JavaScript. Ce type de validation peut être contourné par l'utilisateur via des requêtes HTTP spécialement conçues pour ne pas remplacer la validation des entrées côté serveur. La validation ne se produit que lorsque vous tentez de soumettre le formulaire. Par conséquent, toutes les entrées restreintes doivent se trouver dans un formulaire pour que la validation ait lieu (sauf si vous utilisez JavaScript). Gardez à l'esprit que les entrées désactivées ou en lecture seule ne déclencheront pas de validation.

Certains nouveaux types de saisie (tels que la `email` , l' `url` , le `tel` , la `date` et bien d'autres) sont automatiquement validés et ne nécessitent pas vos propres contraintes de validation.

5

Champs obligatoires

Utilisez l'attribut `required` pour indiquer qu'un champ doit être rempli afin de réussir la validation.

```
<input required>
```

Longueur minimum / maximum

Utilisez les `minlength` et `maxlength` pour indiquer les exigences de longueur. La plupart des navigateurs empêcher l'utilisateur de taper plus de caractères *max* dans la boîte, les empêchant de faire leur entrée invalide même avant de tenter la soumission.

```
<input minlength="3">
<input maxlength="15">
<input minlength="3" maxlength="15">
```

Spécifier une plage

Utiliser les attributs `min` et `max` pour restreindre la plage de nombres qu'un utilisateur peut saisir dans une entrée de type `number` ou `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" />
```

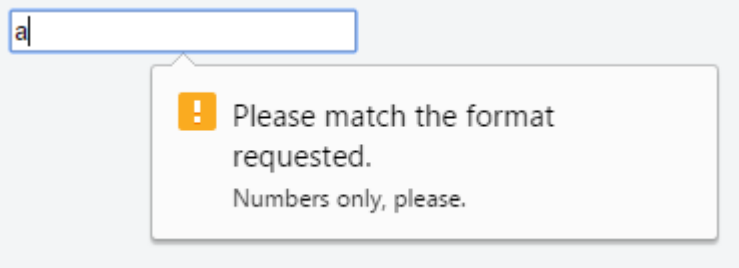
5

Faire correspondre un motif

Pour plus de contrôle, utilisez l'attribut `pattern` pour spécifier toute expression régulière qui doit être mise en correspondance afin de réussir la validation. Vous pouvez également spécifier un `title` , qui est inclus dans le message de validation si le champ ne passe pas.

```
<input pattern="\d*" title="Numbers only, please.">
```

Voici le message affiché dans Google Chrome version 51 lors de la tentative de soumission du formulaire avec une valeur non valide dans ce champ:



Tous les navigateurs n'affichent pas de message pour les modèles non valides, bien qu'il existe un support complet parmi les navigateurs modernes les plus utilisés.

Vérifiez le dernier support sur CanIUse et implémentez-le en conséquence.

5

Accepter le type de fichier

Pour les champs d'entrée de type `file` , il est possible d'accepter uniquement certains types de fichiers, tels que les vidéos, les images, les audios, les extensions de fichiers spécifiques ou certains types de supports . Par exemple:

```
<input type="file" accept="image/*" title="Only images are allowed">
```

Plusieurs valeurs peuvent être spécifiées avec une virgule, par exemple:

```
<input type="file" accept="image/*,.rar,application/zip">
```

Remarque: L' `novalidate` attribut `novalidate` à l'élément de `form` ou à l'attribut `formnovalidate` au bouton d' `formnovalidate` empêche la validation sur les éléments de formulaire. Par exemple:

```
<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>
```

Le formulaire comporte des champs requis pour "publier" le brouillon mais n'est pas requis pour "enregistrer" le brouillon.

Réinitialiser

```
<input type="reset" value="Reset">
```

Une entrée de type `reset` crée un bouton qui, une fois cliqué, réinitialise toutes les entrées sous sa forme d'origine.

- Le texte d'un champ de saisie sera réinitialisé ou sa valeur par défaut (spécifiée à l'aide de l'attribut `value`).
- Toute option dans un menu de sélection sera désélectionnée sauf si elle possède l'attribut `selected` .
- Toutes les cases à cocher et les cases radio seront désélectionnées, sauf si elles possèdent l'attribut `checked` .

Remarque: Un bouton de réinitialisation doit être situé à l'intérieur ou attaché (via l'attribut de `form`) à un élément `<form>` pour avoir un effet. Le bouton ne réinitialisera que les éléments de ce formulaire.

Nombre

5

```
<input type="number" value="0" name="quantity">
```

L'élément Input avec un attribut type dont la valeur est `number` représente un contrôle précis pour définir la valeur de l'élément sur une chaîne représentant un nombre.

Veuillez noter que ce champ ne garantit pas d'avoir un numéro correct. Il ne permet que tous les

symboles pouvant être utilisés dans un nombre réel, par exemple l'utilisateur pourra entrer une valeur comme `e1e-,0` .

Tel

```
<input type="tel" value="+8400000000">
```

L'élément input avec un attribut type dont la valeur est `tel` représente un contrôle d'édition de texte en clair sur une ligne pour la saisie d'un numéro de téléphone.

Email

5

Le `<input type="email">` est utilisé pour les champs de saisie qui doivent contenir une adresse électronique.

```
<form>
  <label>E-mail: <label>
  <input type="email" name="email">
</form>
```

L'adresse e-mail peut être automatiquement validée lorsqu'elle est soumise en fonction de la prise en charge du navigateur.

Bouton

```
<input type="button" value="Button Text">
```

Les boutons peuvent être utilisés pour déclencher des actions sur la page sans soumettre le formulaire. Vous pouvez également utiliser l'élément `<button>` si vous avez besoin d'un bouton pouvant être plus facilement stylé ou contenant d'autres éléments:

```
<button type="button">Button Text</button>
```

Les boutons sont généralement utilisés avec un événement "onclick":

```
<input type="button" onclick="alert('hello world!')" value="Click Me">
```

ou

```
<button type="button" onclick="alert('hello world!')">Click Me</button>
```

Les attributs

[name]

Le `name` du bouton, qui est soumis avec les données du formulaire.

[type]

Le `type` du bouton.

Les valeurs possibles sont:

`submit` : le bouton envoie les données du formulaire au serveur. C'est la valeur par défaut si l'attribut n'est pas spécifié ou si l'attribut est modifié dynamiquement en une valeur vide ou non valide.

`reset` : le bouton réinitialise toutes les commandes à leurs valeurs initiales.

`button` : le bouton n'a pas de comportement par défaut. Des scripts côté client peuvent être associés aux événements de l'élément, qui sont déclenchés lorsque les événements se produisent.

`menu` : Le bouton ouvre un menu contextuel défini via son élément désigné.

[value]

La valeur initiale du bouton.

5

Attributs supplémentaires pour les boutons de soumission

Attribut	La description
`form`	Spécifie l'ID du formulaire auquel le bouton appartient. Si aucun n'est spécifié, il appartiendra à son élément de formulaire ancêtre (s'il en existe un).
`formaction`	Spécifie où envoyer les données de formulaire lorsque le formulaire est soumis en utilisant ce bouton.
`formenctype`	Spécifie comment les données de formulaire doivent être encodées lors de la soumission au serveur en utilisant ce bouton. Ne peut être utilisé qu'avec `formmethod="post"` .
`formmethod`	Spécifie la méthode HTTP à utiliser (POST ou GET) lors de l'envoi de données de formulaire à l'aide de ce bouton.
`formnovalidate`	Indique que les données de formulaire ne doivent pas être validées lors de la soumission.

Attribut	La description
`formtarget`	Spécifie où afficher la réponse reçue après avoir soumis le formulaire en utilisant ce bouton.

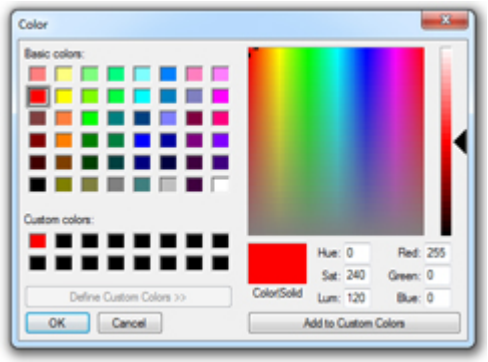
Couleur

5

```
<input type="color" name="favcolor" value="#ff0000">
```

Dans les navigateurs prenant en charge, l'élément input avec un attribut type dont la valeur est `color` crée un contrôle de type bouton, avec une couleur égale à la valeur de l'attribut `color` (par défaut noir si la valeur n'est pas spécifiée ou est un format hexadécimal invalide).

En cliquant sur ce bouton, vous ouvrez le widget couleur du système d'exploitation, qui permet à l'utilisateur de sélectionner une couleur.



Le repli pour les navigateurs qui ne supportent pas ce type d'entrée est un `type=text` entrée standard `type=text` .

URL

5

```
<input type="url" name="Homepage">
```

Ceci est utilisé pour les champs de saisie qui doivent contenir une adresse URL.

Selon la prise en charge du navigateur, le champ `url` peut être automatiquement validé lors de la soumission.

Certains smartphones reconnaissent le type d' `url` et ajoutent ".com" au clavier pour correspondre à l'entrée d'URL.

Rendez-vous amoureux

5

```
<input type="date" />
```

Un sélecteur de date apparaîtra à l'écran pour vous permettre de choisir une date. Ceci n'est pas pris en charge dans Firefox ou Internet Explorer.

DateTime-Local

5

```
<input type="datetime-local" />
```

En fonction de la prise en charge du navigateur, un sélecteur de date et d’heure apparaîtra à l’écran pour que vous choisissiez une date et une heure.

Image

```
<input type="image" src="img.png" alt="image_name" height="50px" width="50px"/>
```

Une image. Vous devez utiliser l'attribut src pour définir la source de l'image et l'attribut alt pour définir un autre texte. Vous pouvez utiliser les attributs height et width pour définir la taille de l'image en pixels.

Gamme

5

```
<input type="range" min="" max="" step="" />
```

Un contrôle pour entrer un nombre dont la valeur exacte n'est pas importante.

Attribut	La description	Valeur par défaut
min	Valeur minimum pour la gamme	0
max	Valeur maximale pour la plage	100
étape	Montant à augmenter de chaque incrément.	1

Mois

5

```
<input type="month" />
```

En fonction de la prise en charge du navigateur, un contrôle apparaîtra pour choisir le mois.

Temps

5

```
<input type="time" />
```

L'entrée de `time` marque cet élément comme acceptant une chaîne représentant une heure. Le format est défini dans la RFC 3339 et devrait être un temps partiel tel que

```
19:04:39
08:20:39.04
```

Actuellement, toutes les versions de Edge, Chrome, Opera et Chrome pour Android prennent en charge le type = "time". Les nouvelles versions du navigateur Android, en particulier 4.4 et plus, le supportent. Safari pour iOS offre une prise en charge partielle, ne prenant pas en charge les attributs min, max et step.

La semaine

5

```
<input type="week" />
```

En fonction de la prise en charge du navigateur, un contrôle affichera pour entrer un numéro de semaine et un numéro de semaine sans fuseau horaire.

Texte

Le type d'entrée le plus basique et l'entrée par défaut si aucun `type` n'est spécifié. Ce type d'entrée définit un champ de texte à une seule ligne avec des sauts de ligne automatiquement supprimés de la valeur d'entrée. Tous les autres caractères peuvent être entrés dans ceci. `<input>` éléments `<input>` sont utilisés dans un élément `<form>` pour déclarer des contrôles d'entrée permettant aux utilisateurs d'entrer des données.

Syntaxe

```
<input type="text">
```

ou (sans spécifier de `type` , en utilisant l'attribut default):

```
<input>
```

La largeur par défaut d'une entrée de champ de texte est de 20 caractères. Cela peut être modifié en spécifiant une valeur pour l'attribut `size` comme ceci:

```
<input type="text" size="50">
```

L'attribut `size` est nettement différent de la définition d'une largeur avec CSS. L'utilisation d'une largeur définit une valeur spécifique (en nombre de pixels, en pourcentage de l'élément parent, etc.) selon laquelle l'entrée doit toujours être large. L'utilisation de la `size` calcule la largeur à allouer en fonction de la police utilisée et de la largeur des caractères.

Remarque: L' utilisation de l'attribut `size` ne limite pas intrinsèquement le nombre de caractères pouvant être saisis dans la boîte, mais uniquement la largeur d'affichage de la boîte. Pour limiter la longueur, voir Validation en entrée .

Un champ de saisie n'autorise qu'une ligne de texte. Si vous avez besoin d'une entrée de texte multiligne pour une quantité substantielle de texte, utilisez plutôt un élément `<textarea>` .

Chercher

5

La recherche de type d'entrée est utilisée pour la recherche textuelle. Il ajoutera un symbole de loupe à côté de l'espace pour le texte sur la plupart des navigateurs.

```
<input type="search" name="googlesearch">
```

DateTime (Global)

L'élément input avec un attribut type dont la valeur est " **datetime** " représente un contrôle permettant de définir la valeur de l'élément sur une chaîne représentant une **date et une heure globales (avec des informations de fuseau horaire).**

```
<fieldset>
   <p><label>Meeting time: <input type=datetime name="meeting.start"></label>
</fieldset>
```

Attributs autorisés:

- attributs globaux
- prénom
- désactivée
- forme
- type
- autocomplete
- autofocus
- liste
- min max
- step (float)
- lecture seulement
- valeur requise

Lire Éléments de contrôle d'entrée en ligne: https://riptutorial.com/fr/html/topic/277/elements-de-controle-d-entree

Chapitre 20: Éléments de coupe

Remarques

Les normes HTML5 ne répertorient pas l'élément principal en tant qu'élément de sectionnement.

Examples

Élément de l'article

L'élément `<article>` contient **un contenu autonome** tel que des articles, des articles de blog, des commentaires d'utilisateurs ou un widget interactif pouvant être distribué en dehors du contexte de la page, par exemple par RSS.

- Lorsque des éléments d'article sont imbriqués, le contenu du nœud d'article interne doit être associé à l'élément d'article externe.

Un blog (`section`) contenant plusieurs articles (`article`) et des commentaires (`article`) pourrait ressembler à ceci.

```
<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>
```

Évitez les utilisations inutiles

Lorsque le contenu principal de la page (à l'exclusion des en-têtes, pieds de page, barres de navigation, etc.) est simplement un groupe d'éléments. Vous pouvez omettre le `<article>` en faveur de l'élément `<main>` .

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

Au lieu de cela, remplacez l'article par un élément `<main>` pour indiquer qu'il s'agit du contenu `main` de cette page.

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

Si vous utilisez un autre élément, assurez-vous de spécifier le rôle `<main>` ARIA pour une interprétation et un rendu corrects sur plusieurs périphériques et navigateurs non HTML5.

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

Remarques:

- `<main>` descendants de l' élément `<main>` ne sont pas autorisés dans un `<article>`

Cliquez ici pour lire la spécification HTML5 officielle de l'élément `<article>`

Élément principal

L'élément `<main>` contient le **contenu principal** de votre page Web. Ce contenu est unique à la page individuelle et ne devrait pas apparaître ailleurs sur le site. Le contenu répétitif comme les en-têtes, les pieds de page, la navigation, les logos, etc., est placé en dehors de l'élément.

- Le `<main>` élément ne doit jamais être utilisé au plus **une fois** sur une seule page.
- L'élément `<main>` ne doit pas être inclus en tant que descendant d'un `article` , d'un élément de `aside` , d'un `footer` , d'un en- `header` ou d'un élément de `nav` .

Dans l'exemple suivant, nous affichons un **seul article de blog** (et des informations connexes telles que des références et des commentaires).

```
<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>
```

- Le billet de blog est contenu dans l'élément `<main>` pour indiquer qu'il s'agit du contenu principal de cette page (et donc unique sur le site Web).
- Les balises `<header>` et `<footer>` sont des *frères* pour l'élément `<main>` .

Remarques:

La spécification HTML5 reconnaît l'élément `<main>` tant qu'élément de **regroupement** et non en tant qu'élément de *sectionnement* .

- Attributs du rôle ARIA : `main` *(par défaut)* , `presentation`

L'ajout d'un attribut de rôle ARIA `role="main"` à d' **autres éléments** destinés à être utilisés comme contenu principal est conseillé pour aider les agents utilisateurs qui ne prennent pas en charge HTML5 et pour fournir davantage de contexte à ceux qui le font.

L'élément `<main>` par défaut a le rôle principal et n'a donc pas besoin d'être fourni.

Cliquez ici pour lire la spécification HTML5 officielle de l'élément `<main>`

Elément de navigation

L'élément `<nav>` est principalement destiné à être utilisé pour des sections contenant **des blocs de navigation principaux** pour le site Web. Cela peut inclure des liens vers d'autres parties de la page Web *(par exemple des ancres pour une table des matières)* ou d'autres pages.

Éléments en ligne

Ce qui suit affichera un ensemble de liens hypertexte en ligne.

```
<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>
```

Utilisez les éléments de la liste si nécessaire

Si le contenu représente une liste d'éléments, utilisez un élément de liste pour l'afficher et améliorer l'expérience utilisateur.

Notez le `role="navigation"` , *plus à ce sujet ci-dessous.*

```
<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>
```

Évitez les utilisations inutiles

`<footer>` éléments `<footer>` peuvent avoir une liste de liens vers d'autres parties du site (FAQ, T & C, etc.). L'élément pied de page seul est suffisant dans ce cas, vous n'avez pas *besoin* d'envelopper vos liens avec un élément `<nav>` dans le `<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>
```

Remarques:

- `<main>` descendants de l' élément `<main>` ne sont pas autorisés dans un `<nav>`

Ajouter un `role="navigation"` ARIA `role="navigation"` à l'élément `<nav>` est conseillé pour aider les agents utilisateurs qui ne prennent pas en charge HTML5 et pour fournir plus de contexte à ceux qui le font.

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

Screen Readers: *(logiciel permettant aux utilisateurs aveugles ou malvoyants de*

naviguer sur le site)

Les agents utilisateurs tels que les lecteurs d'écran interpréteront l'élément `<nav>` différemment en fonction de leurs besoins.

- Cela peut donner à l'élément `<nav>` une priorité plus élevée lors du rendu de la page
- Cela pourrait retarder le rendu de l'élément
- Il pourrait adapter la page de manière spécifique pour répondre aux besoins de l'utilisateur
  Exemple: agrandissez les liens de texte dans les éléments `<nav>` pour une personne ayant une déficience visuelle.

Cliquez ici pour lire la spécification HTML5 officielle de l'élément `<nav>`

Élément de section

L'élément `<section>` représente une section générique pour regrouper par thème le contenu. Chaque section, en général, devrait pouvoir être identifiée avec un élément d'en-tête en tant qu'enfant de la `section` .

- Vous pouvez utiliser l'élément `<section>` dans un `<article>` et vice-versa.
- Chaque section doit avoir un *thème* (un élément de titre identifiant cette région)
- N'utilisez pas l'élément `<section>` tant que "conteneur" de style général.
  Si vous avez besoin d'un conteneur pour appliquer un style, utilisez plutôt un `<div>` .

Dans l'exemple suivant, nous affichons un **seul article de blog** avec plusieurs chapitres. Chaque chapitre est une section *(un ensemble de contenu regroupé par thème, qui peut être identifié par les éléments de titre de chaque section)* .

```
<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>
```

Remarques:

Les développeurs doivent utiliser l'élément **article** lorsqu'il est judicieux de syndiquer le

contenu de l'élément.

Cliquez ici pour lire la spécification HTML5 officielle de l'élément `<main>`

Élément d'en-tête

L'élément `<header>` représente le contenu d'introduction pour le contenu de section ancêtre ou l'élément racine de sectionnement le plus proche. Un `<header>` contient généralement un groupe d'aides d'introduction ou de navigation.

> **Remarque:** L'élément d'en-tête ne divise pas le contenu. il n'introduit pas une nouvelle section.

Exemples:

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

Dans cet exemple, `<article>` a 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>
```

Recommandation proposée par le W3C

Élément de pied de page

L'élément `<footer>` contient la partie pied de page de la page.

Voici un exemple d'élément `<footer>` contenant une balise de paragraphe `p` .

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

Lire Éléments de coupe en ligne: https://riptutorial.com/fr/html/topic/311/elements-de-coupe

Chapitre 21: Éléments multimédias

Paramètres

Attribut	Détails
largeur	Définit la largeur de l'élément en pixels.
la taille	Définit la hauteur de l'élément en pixels.
`<source>`	Définit les ressources des fichiers audio ou vidéo
Piste	Définit la piste de texte pour les éléments multimédias
contrôles	Affiche les commandes
lecture automatique	Commence automatiquement à jouer les médias
boucle	Joue le média dans un cycle répété
en sourdine	Joue les médias sans son
affiche	Assigne une image à afficher jusqu'à ce qu'une vidéo soit chargée

Remarques

Prise en charge dans les navigateurs

Fonctionnalité	Chrome	Firefox (Gecko)	Internet Explorer	Opéra	Safari
Support de base	3.0	3.5 (1.9.1)	9.0	10.50	3.1
`<audio>` : PCM dans WAVE	(Oui)	3.5 (1.9.1)	Pas de support	10.50	3.1
`<audio>` : Vorbis dans WebM	(Oui)	4.0 (2.0)	Pas de support	10.60	3.1
`<audio>` : Streaming Vorbis / Opus dans WebM via MSE	?	36,0	?	?	?
`<audio>` : Vorbis dans Ogg	(Oui)	3.5 (1.9.1)	Pas de support	10.50	Pas de support
`<audio>` : MP3	(Oui)	(Oui)	9.0	(Oui)	3.1

Fonctionnalité	Chrome	Firefox (Gecko)	Internet Explorer	Opéra	Safari
`<audio>` : MP3 en MP4	?	?	?	?	(Oui)
`<audio>` : AAC en MP4	(Oui)	(Oui)	9.0	(Oui)	3.1
`<audio>` : Opus dans Ogg	27,0	15,0 (15,0)	?	?	?
`<video>` : VP8 et Vorbis dans WebM	6,0	4.0 (2.0)	9.0	10.60	3.1
`<video>` : VP9 et Opus dans WebM	29.0	28,0 (28,0)	?	(Oui)	?
`<video>` : Streaming WebM via MSE	?	42,0 (42,0)	?	?	?
`<video>` : Theora et Vorbis dans Ogg	(Oui)	3.5 (1.9.1)	Pas de support	10.50	Pas de support
`<video>` : H.264 et MP3 en MP4	(Oui)	(Oui)	9.0	(Oui)	(Oui)
`<video>` : H.264 et AAC en MP4	(Oui)	(Oui)	9.0	(Oui)	3.1
tout autre format	Pas de support	Pas de support	Pas de support	Pas de support	3.1

Examples

Utiliser ` `et` `élément pour afficher le contenu audio / vidéo

Utilisez l'élément HTML ou `<audio>` pour incorporer du contenu vidéo / audio dans un document. L'élément vidéo / audio contient une ou plusieurs sources vidéo / audio. Pour spécifier une source, utilisez l'attribut src ou l'élément `<source>` . le navigateur choisira le plus approprié.

Exemple de balise 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>
```

Exemple de balise 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>
```

l'audio

HTML5 fournit une nouvelle norme pour intégrer un fichier audio sur une page Web.

Vous pouvez incorporer un fichier audio à une page en utilisant l'élément `<audio>` :

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

Vidéo

Vous pouvez également intégrer une vidéo à une page Web à l'aide de l'élément `<video>` :

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

En-tête vidéo ou arrière-plan

Ajout d'une vidéo qui sera automatiquement lancée sur une boucle et qui n'a ni contrôle ni son. Parfait pour un en-tête vidéo ou un arrière-plan.

```
<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.ogg" type="video/ogg">
</video>
```

Ce CSS fournit une solution de secours si la vidéo ne peut pas être chargée. Notez qu'il est recommandé d'utiliser la première image de la vidéo comme affiche video.jpg.

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

Lire Éléments multimédias en ligne: https://riptutorial.com/fr/html/topic/2111/elements-multimedias

Chapitre 22: Éléments vides

Introduction

Toutes les balises HTML ne sont pas de la même structure. Bien que la plupart des éléments requièrent une balise d'ouverture, une balise de fermeture et un contenu, certains éléments - appelés éléments nuls - ne nécessitent qu'une balise d'ouverture car ils ne contiennent aucun élément. Cette rubrique explique et démontre l'utilisation correcte des éléments vides dans HTML

Remarques

Un élément vide ne peut avoir aucun contenu mais peut avoir des attributs. Les éléments nuls sont auto-fermants, ils ne doivent donc pas avoir de balise de fermeture.

En HTML5 , les éléments suivants sont nuls:

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

Examples

Éléments vides

HTML 4.01 / XHTML 1.0 Strict comprend les éléments suivants:

- `area` - `area` cliquable, définie dans une image
- `base` - spécifie une URL de base à partir de laquelle toutes les bases de liens
- `br` - saut de ligne
- `col` - colonne dans une table [obsolète]
- `hr` - règle horizontale (ligne)
- `img` - image
- `input` - champ où les utilisateurs entrent des données
- `link` - lie une ressource externe au document
- `meta` - fournit des informations sur le document

- `param` - définit les paramètres pour les plugins

Les normes HTML 5 incluent toutes les balises non obsolètes de la liste précédente et

- `command` - représente une commande que les utilisateurs peuvent appeler [obsolète]
- `keygen` - facilite la génération de clés publiques pour les certificats Web [obsolète]
- `source` - spécifie les sources de média pour les éléments `picture` , `audio` et `video`

L'exemple ci-dessous n'inclut **pas** les éléments vides:

```
<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>
```

Notez comment chaque élément a une balise d'ouverture, une balise de fermeture et du texte ou d'autres éléments à l'intérieur des balises d'ouverture et de fermeture. Les balises annulées figurent toutefois dans l'exemple ci-dessous:

```
<img src="https://cdn.sstatic.net/Sites/stackoverflow/company/img/logos/so/so-icon.png" />
<br>
<hr>
<input type="number" placeholder="Enter your favorite number">
```

À l'exception de la balise img, tous ces éléments nuls ne comportent qu'une balise d'ouverture. La balise img, contrairement à toute autre balise, a une fermeture auto `/` avant le signe supérieur de l'étiquette d'ouverture. Il est préférable d'avoir un espace avant la barre oblique.

Lire Éléments vides en ligne: https://riptutorial.com/fr/html/topic/1449/elements-vides

Chapitre 23: Entités de caractère

Examples

Caractères spéciaux communs

Certains caractères peuvent être réservés pour HTML et ne peuvent pas être utilisés directement car ils peuvent obstruer les codes HTML réels. Par exemple, si vous essayez d'afficher les crochets d'angle gauche et droit (`<>`) dans le code source, cela peut entraîner des résultats inattendus dans la sortie. De même, les espaces vides tels qu'ils sont écrits dans le code source peuvent ne pas s'afficher comme prévu dans la sortie HTML. Certains, comme ☎, ne sont pas disponibles dans le jeu de caractères ASCII.

A cette fin, des entités de caractères sont créées. Ce sont de la forme `&entity_name;` ou `&entity_number;` . Voici quelques-unes des entités HTML disponibles.

Personnage	La description	Nom de l'entité	Numéro d'entité
“ ”	Espace non-cassant	` `	` `
“<”	moins que	`<`	`&#60;`
“>”	plus grand que	`>`	`&#62;`
“&”	esperluette	`&`	`&#38;`
“-”	em dash	`&mdash;`	`&#8212;`
“-”	en course	`&ndash;`	`&#8211;`
“©”	droits d'auteur	`&copy;`	`&#169;`
“®”	marque déposée	`&reg;`	`&#174;`
“™”	marque déposée	`&trade;`	`&#8482;`
“☎”	téléphone	`&phone;`	`&#9742;`

Ainsi, écrire

© 2016 Stack Exchange Inc.

le code HTML suivant est utilisé:

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

Entités de caractère en HTML

De nombreux symboles et caractères spéciaux sont requis lors du développement d'une page Web en HTML, mais comme nous le savons, l'utilisation directe de caractères peut parfois interférer avec le code HTML réel qui contient certains caractères et certains caractères non disponibles au clavier. Ainsi, pour éviter le conflit et pouvoir utiliser simultanément des symboles différents dans notre code, w3 org nous fournit des «Entités de caractère».

Les entités de caractères sont prédéfinies avec 'Nom d'entité' - & entity_name; et 'Numéro d'entité' - & entity_number; Nous devons donc utiliser l'un des deux pour que le symbole requis soit affiché sur notre page.

La liste de quelques entités de caractère peut être trouvée à https://dev.w3.org/html5/html-author/charref

Un exemple simple avec l'utilisation de l'entité de caractère pour «loupe»:

```
<input type="text" placeholder="  &#128269; Search"/>
```

qui rend comme

Lire Entités de caractère en ligne: https://riptutorial.com/fr/html/topic/5229/entites-de-caractere

Chapitre 24: Formatage du texte

Introduction

Alors que la plupart des balises HTML sont utilisées pour créer des éléments, HTML fournit également des balises de mise en forme dans le texte pour appliquer des styles de texte spécifiques à des parties de texte. Cette rubrique comprend des exemples de mise en forme de texte HTML, tels que la mise en surbrillance, le gras, le soulignement, l'indice et le texte mis en évidence.

Syntaxe

- `<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

Gras, italique et souligné

Texte en gras

Pour mettre du texte en gras, utilisez les balises `<strong>` ou `<b>` :

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

ou

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

Quelle est la différence? Sémantique. `<strong>` est utilisé pour indiquer que le texte est fondamentalement ou sémantiquement *important* pour le texte environnant, tandis que `<b>` n'indique pas une telle importance et représente simplement un texte qui doit être mis en gras.

Si vous utilisiez `<b>` un programme de synthèse vocale ne dirait pas le (s) mot (s) différemment que les autres mots qui l'entourent - vous attirez simplement l'attention sur eux sans ajouter d'importance supplémentaire. En utilisant `<strong>` , cependant, le même programme voudrait

prononcer ces mots avec un ton différent pour indiquer que le texte est important d'une manière ou d'une autre.

Texte en italique

Pour mettre du texte en italique, utilisez les balises `<em>` ou `<i>` :

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

ou

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

Quelle est la différence? Sémantique. `<em>` est utilisé pour indiquer que le texte doit être accentué de manière à être souligné, tandis que `<i>` représente simplement un texte qui doit être séparé du texte qui l'entoure.

Par exemple, si vous voulez insister sur l'action à l'intérieur d'une phrase, vous pouvez le faire en italique via `<em>` : "Souhaitez-vous simplement *soumettre* l'édition?"

Mais si vous identifiiez un livre ou un journal que vous classeriez normalement en italique, vous utiliseriez simplement `<i>` : "J'ai été obligé de lire *Roméo et Juliette* au lycée.

Texte souligné

Alors que l'élément `<u>` lui-même était déconseillé dans HTMl 4, il a été réintroduit avec une signification sémantique alternative dans HTML 5 - pour représenter une annotation non textuelle et non textuelle. Vous pouvez utiliser un tel rendu pour indiquer un texte mal orthographié sur la page ou pour une marque chinoise.

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

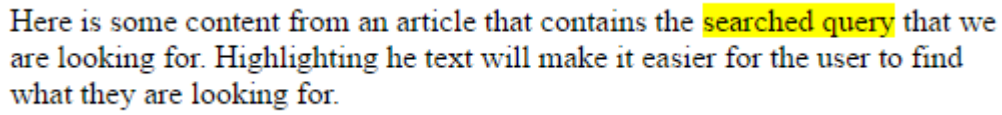
Mise en évidence

L'élément `<mark>` est nouveau en HTML5 et permet de marquer ou de mettre en évidence du texte dans un document "en raison de sa pertinence dans un autre contexte". [1]

L'exemple le plus courant serait dans les résultats d'une recherche si l'utilisateur a entré une requête de recherche et que les résultats sont affichés en mettant en évidence la requête souhaitée.

```
<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>
```

Sortie:



Un format standard courant est le texte noir sur fond jaune, mais cela peut être modifié avec CSS.

Inséré, supprimé ou affaibli

Pour marquer le texte comme inséré, utilisez la `<ins>` :

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

Pour marquer le texte comme étant supprimé, utilisez la `<del>` :

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

Pour parcourir le texte, utilisez la `<s>` :

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

En exposant et indice

Pour décaler le texte vers le haut ou vers le bas, vous pouvez utiliser les balises `<sup>` et `<sub>` .

Pour créer un exposant:

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

Pour créer un indice:

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

Abréviation

Pour marquer une expression comme une abréviation, utilisez la `<abbr>` :

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

S'il est présent, l'attribut `title` est utilisé pour présenter la description complète de cette abréviation.

Lire Formatage du texte en ligne: https://riptutorial.com/fr/html/topic/526/formatage-du-texte

Chapitre 25: Formes

Introduction

Pour regrouper des éléments d'entrée et soumettre des données, HTML utilise un élément de formulaire pour encapsuler des éléments d'entrée et de soumission. Ces formulaires gèrent l'envoi des données dans la méthode spécifiée vers une page gérée par un serveur ou un gestionnaire. Cette rubrique explique et illustre l'utilisation des formulaires HTML dans la collecte et la soumission des données d'entrée.

Syntaxe

- `<form method="post|get" action="somePage.php" target="_blank|_self|_parent|_top|framename">`

Paramètres

Attribut	La description
`accept-charset`	Spécifie les codages de caractères à utiliser pour la soumission du formulaire.
`action`	Spécifie où envoyer les données de formulaire lorsqu'un formulaire est soumis.
`autocomplete`	Spécifie si une saisie semi-automatique doit être activée ou non pour un formulaire.
`enctype`	Spécifie comment les données de formulaire doivent être encodées lors de leur soumission au serveur (uniquement pour method = "post").
`method`	Spécifie la méthode HTTP à utiliser lors de l'envoi de données de formulaire (POST ou GET).
`name`	Spécifie le nom d'un formulaire.
`novalidate`	Spécifie que le formulaire ne doit pas être validé lors de la soumission.
`target`	Spécifie où afficher la réponse reçue après la soumission du formulaire.

Remarques

L'élément `<form>` représente une section contenant des éléments associés à un formulaire (par exemple, `<button>` `<fieldset>` `<input>` `<label>` `<output>` `<select>` `<textarea>`) qui soumet des informations à un serveur. Les deux balises de début (`<form>`) et de fin (`</form>`) sont obligatoires.

Examples

Soumission

L'attribut d'action

L'attribut action définit l'action à effectuer lorsque le formulaire est soumis, ce qui conduit généralement à un script qui collecte les informations soumises et fonctionne avec lui. si vous le laissez vide, il l'enverra au même fichier

```
<form action="action.php">
```

L'attribut de la méthode

L'attribut method est utilisé pour définir la méthode HTTP du formulaire GET ou POST.

```
<form action="action.php" method="get">
<form action="action.php" method="post">
```

La méthode GET est principalement utilisée pour *obtenir des* données, par exemple pour recevoir une publication par son identifiant ou son nom, ou pour soumettre une requête de recherche. La méthode GET ajoutera les données de formulaire à l'URL spécifiée dans l'attribut d'action.

```
www.example.com/action.php?firstname=Mickey&lastname=Mouse
```

La méthode POST est utilisée lors de la soumission de données à un script. La méthode POST n'ajoute pas les données de formulaire à l'URL d'action mais envoie à l'aide du corps de la requête.

Pour soumettre correctement les données du formulaire, un nom d'attribut de nom doit être spécifié.
Par exemple, envoyons la valeur du champ et définissons son nom sur *lastname* :

```
<input type="text" name="lastname" value="Mouse">
```

Plus d'attributs

```
<form action="action.php" method="post" target="_blank" accept-charset="UTF-8"
enctype="application/x-www-form-urlencoded" autocomplete="off" novalidate>

<!-- form elements -->

</form>
```

Attribut cible dans la balise de formulaire

L'attribut target spécifie un nom ou un mot-clé qui indique où afficher la réponse reçue après la soumission du formulaire.

L'attribut target définit un nom ou un mot-clé pour un contexte de navigation (par exemple, tabulation, fenêtre ou cadre en ligne).

De balise avec un attribut cible:

```
<form target="_blank">
```

Valeurs d'attribut

Valeur	La description
_blanc	La réponse est affichée dans une nouvelle fenêtre ou un nouvel onglet
_soi	La réponse est affichée dans le même cadre (par défaut)
_parent	La réponse est affichée dans le cadre parent
_Haut	La réponse est affichée dans le corps entier de la fenêtre
framename	La réponse est affichée dans un iframe nommé

> Remarque: l'attribut target a été ***déprécié*** dans **HTML 4.01** . L'attribut cible est ***pris*** en ***charge*** dans ***HTML5*** .
>
> Les cadres et les jeux de cadres ne sont pas pris en charge en ***HTML5*** . Par conséquent, les ***valeurs _parent, _top et framename sont désormais principalement utilisées avec les iframes*** .

Téléchargement de fichiers

Les images et les fichiers peuvent être téléchargés / soumis au serveur en définissant l'attribut `enctype` de la balise de `form` sur `multipart/form-data` . `enctype` spécifie comment les données de formulaire seraient encodées lors de la soumission au serveur.

Exemple

```
<form method="post" enctype="multipart/form-data" action="upload.php">
    <input type="file" name="pic" />
    <input type="submit" value="Upload" />
</form>
```

Grouper quelques champs d'entrée

Lors de la conception d'un formulaire, vous souhaiterez peut-être regrouper quelques champs de saisie dans un groupe pour vous aider à organiser la mise en forme du formulaire. Cela peut être

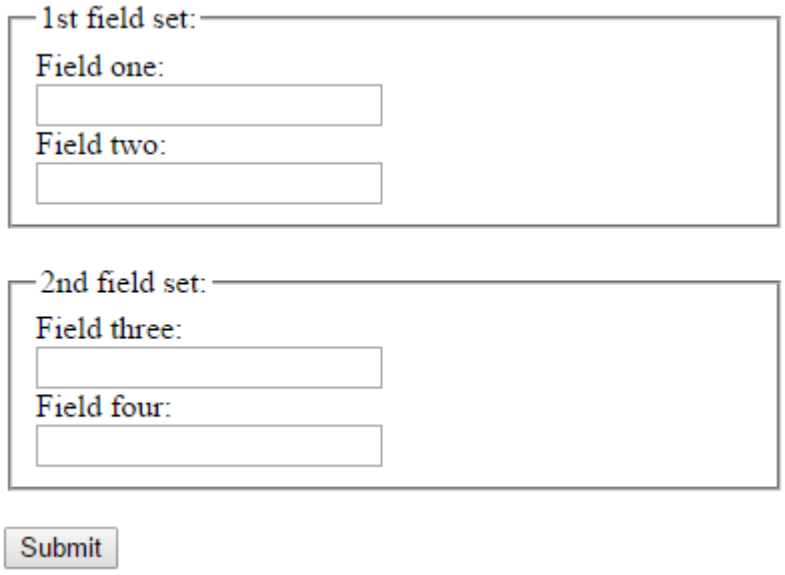
fait en utilisant la balise. Voici un exemple d'utilisation.

Pour chaque jeu de champs, vous pouvez définir une légende pour l'ensemble à l'aide de la balise LEGEND TEXT.

Exemple

```
<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>
```

Résultat



Prise en charge du navigateur

Les dernières versions de Chrome, IE, Edge, FireFox, Safari et Opera prennent également en charge la balise

Lire Formes en ligne: https://riptutorial.com/fr/html/topic/1160/formes

Chapitre 26: HTML 5 Cache

Remarques

Le fichier manifeste est un simple fichier texte qui indique au navigateur ce qu'il doit mettre en cache (et ce qu'il ne faut jamais mettre en cache). L'extension de fichier recommandée pour les fichiers manifest est: ".appcache" Le fichier manifeste comporte trois sections:

CACHE MANIFEST - Les fichiers listés sous cet en-tête seront mis en cache après leur premier téléchargement

NETWORK - Les fichiers répertoriés sous cet en-tête nécessitent une connexion au serveur et ne seront jamais mis en cache

FALLBACK - Les fichiers répertoriés sous cet en-tête spécifient les pages de secours si une page est inaccessible

Examples

Exemple de base de cache Html 5

c'est notre fichier index.html

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

alors nous allons créer le fichier index.appcache avec les codes ci-dessous

```
CACHE MANIFEST
index.html
```

écrivez les fichiers que vous voulez mettre en cache

Remarque: Les deux fichiers doivent être dans le même dossier dans cet exemple

Lire HTML 5 Cache en ligne: https://riptutorial.com/fr/html/topic/8024/html-5-cache

Chapitre 27: IFrames

Paramètres

Attribut	Détails
`name`	Définit le nom de l'élément, à utiliser avec `a` étiquette pour changer de l'iframe `src` .
`width`	Définit la largeur de l'élément en pixels.
`height`	Définit la hauteur de l'élément en pixels.
`src`	Spécifie la page qui sera affichée dans le cadre.
`srcdoc`	Spécifie le contenu qui sera affiché dans le cadre, en supposant que le navigateur le supporte. Le contenu doit être HTML valide.
`sandbox`	Une fois défini, le contenu de l'iframe est traité comme étant d'une origine unique et les fonctionnalités, y compris les scripts, les plug-ins, les formulaires et les fenêtres contextuelles, seront désactivées. Les restrictions peuvent être assouplies de manière sélective en ajoutant une liste de valeurs séparées par des espaces. Voir le tableau dans Remarques pour les valeurs possibles.
`allowfullscreen`	`requestFullscreen()` le contenu de l'iframe doit être utilisé avec `requestFullscreen()`

Remarques

Un iframe est utilisé pour incorporer un autre document dans le document HTML en cours.

Vous pouvez utiliser les iframes pour afficher:

- autres pages HTML sur le même domaine;
- autres pages HTML sur un autre domaine (voir ci-dessous - Politique de même origine);
- Documents PDF (bien que IE puisse avoir des problèmes, cette question SO peut aider);

Vous devez utiliser un iframe en dernier recours, car il a des problèmes avec la mise en signet et la navigation, et il y a toujours de meilleures options autres qu'un iframe. Cette question devrait vous aider à mieux comprendre les hauts et les bas des iframes.

Politique de même origine

Certains sites ne peuvent pas être affichés en utilisant un iframe, car ils appliquent une stratégie appelée stratégie de même origine . Cela signifie que le site sur lequel se trouve l'iframe doit se trouver sur le même domaine que celui à afficher.

Cette politique s'applique également à la manipulation du contenu qui réside dans un iFrame. Si l'iFrame accède à du contenu provenant d'un autre domaine, vous ne pourrez pas accéder au contenu d'un iFrame ni le manipuler.

L'élément iframe sur W3C

---

attribut `sandbox`

L'attribut `sandbox` , lorsqu'il est défini, ajoute des restrictions supplémentaires à l'iframe. Une liste de jetons séparés par des espaces peut être utilisée pour assouplir ces restrictions.

Valeur	Détails
`allow-forms`	Permet de soumettre des formulaires.
`allow-pointer-lock`	Active l'API du pointeur JavaScript.
`allow-popups`	Les fenêtres contextuelles peuvent être créées à l'aide de `window.open` ou de `<a target="_blank"`
`allow-same-origin`	Le document iframe utilise son origine réelle au lieu d'en recevoir une unique. S'il est utilisé avec `allow-scripts` le document iframe peut supprimer tout sandboxing s'il provient de la même origine que le document parent.
`allow-scripts`	Active les scripts. Le document iframe et le document parent peuvent être en mesure de communiquer entre eux à l'aide de l'API `postMessage()` . S'il est utilisé avec `allow-same-origin` le document iframe peut supprimer tous les sandboxing de la même origine que le document parent.
`allow-top-navigation`	Permet au contenu de l'iframe de modifier l'emplacement du document de niveau supérieur.

Examples

Les bases d'un cadre en ligne

Le terme "IFrame" signifie Inline Frame. Il peut être utilisé pour inclure une autre page dans votre page. Cela donnera un petit cadre qui montre le contenu exact du `base.html` .

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

Définition de la taille du cadre

Le format IFrame peut être redimensionné à l'aide des attributs `width` et `height` , où les valeurs sont représentées en pixels (valeurs de pourcentage autorisées pour HTML 4.01, mais HTML 5 n'autorise que les valeurs en pixels CSS).

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

Utiliser des ancres avec IFrames

Normalement, un changement de page Web dans un Iframe est initié avec l'Iframe, par exemple, en cliquant sur un lien à l'intérieur de Ifame. Cependant, il est possible de modifier le contenu d'un IFrame depuis l'extérieur de IFrame. Vous pouvez utiliser une balise d'ancrage dont l'attribut `href` est défini sur l'URL souhaitée et dont `target` attribut `target` est défini sur l'attribut `name` l'iframe.

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

Utiliser l'attribut "srcdoc"

L'attribut `srcdoc` peut être utilisé (au lieu de l'attribut `src`) pour spécifier le contenu exact de l'iframe en tant que document HTML complet. Cela donnera un IFrame avec le texte "IFrames is cool!"

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

Si l'attribut `srcdoc` n'est pas pris en charge par le navigateur, l'IFrame utilisera plutôt l'attribut `src` , mais si les attributs `src` et `srcdoc` sont présents et pris en charge par le navigateur, `srcdoc` a priorité.

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

Dans l'exemple ci-dessus, si le navigateur ne prend pas en charge l'attribut `srcdoc` , il affichera plutôt le contenu de la page `base.html` .

Bac à sable

Ce qui suit incorpore une page Web non fiable avec toutes les restrictions activées

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

Pour permettre à la page d'exécuter des scripts et de soumettre des formulaires, ajoutez `allow-scripts` et `allow-forms` à l' `sandbox attribute` .

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

S'il existe un contenu non fiable (tel que des commentaires utilisateur) sur le même domaine que la page Web parent, un iframe peut être utilisé pour désactiver les scripts tout en permettant au document parent d'interagir avec son contenu à l'aide de JavaScript.

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

Le document parent peut ajouter des écouteurs d'événement et redimensionner l'IFrame en fonction de son contenu. Ceci, associé à `allow-top-navigation` , peut faire en sorte que l'iframe en bac à sable apparaisse comme faisant partie du document parent.

Ce bac à sable ne remplace pas l'injection d'assainissement mais peut être utilisé dans le cadre d'une stratégie de défense en profondeur .

Sachez également que ce sandbox peut être corrompu par un attaquant qui convainc un utilisateur de visiter directement la source de l'iframe. L'en-tête HTTP de la stratégie de sécurité du contenu peut être utilisé pour atténuer cette attaque.

Lire IFrames en ligne: https://riptutorial.com/fr/html/topic/499/iframes

Chapitre 28: Images

Syntaxe

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

Paramètres

Paramètres	Détails
`src`	Spécifie l'URL de l'image
`srcset`	Images à utiliser dans différentes situations (par exemple, écrans haute résolution, petits écrans, etc.)
`sizes`	Taille de l'image entre les points d'arrêt
`crossorigin`	Comment l'élément gère les requêtes croisées
`usemap`	Nom de la carte image à utiliser
`ismap`	Si l'image est une carte d'image côté serveur
`alt`	Texte de remplacement à afficher si, pour une raison quelconque, l'image n'a pas pu être affichée
`width`	Spécifie la largeur de l'image (facultatif)
`height`	Spécifie la hauteur de l'image (facultatif)

Examples

Créer une image

Pour ajouter une image à une page, utilisez la balise image.

Les tags d'image (`img`) n'ont pas de balise de fermeture. Les deux principaux attributs que vous attribuez à la balise `img` sont `src` , la source d'image et `alt` , qui est un autre texte décrivant l'image.

```
<img src="images/hello.png" alt="Hello World">
```

Vous pouvez également obtenir des images depuis une URL Web:

```
<img src="https://i.stack.imgur.com/ALgZi.jpg?s=48&g=1" alt="StackOverflow user Caleb
```

```
Kleveter">
```

Remarque: les images ne sont pas techniquement insérées dans une page HTML, les images sont liées à des pages HTML. La `<img>` crée un espace de maintien pour l'image référencée.

Il est également possible d'intégrer des images directement dans la page en utilisant base64:

```
<img src="..." alt="Hello World">
```

Conseil: Pour lier une image à un autre document, imbriquez simplement la `<img>` dans les balises `<a>` .

Largeur et hauteur de l'image

> **Remarque:** Les **attributs width et height *ne* sont *pas* obsolètes sur les images** et ne l'ont jamais été. Leur utilisation a été rendue beaucoup plus stricte.

Les dimensions d'une image peuvent être spécifiées à l'aide des attributs `width` et `height` de la balise image:

```
<img src="images/200x200-img.png" width="200" height="200" alt="A 200x200 image">
```

En spécifiant la `width` et la `height` d'une image, votre structure donne au navigateur un aperçu de la disposition de la page, même si vous spécifiez uniquement la taille réelle de l'image. Si les dimensions de l'image ne sont pas spécifiées, le navigateur devra recalculer la mise en page de la page après le chargement de l'image, ce qui pourrait provoquer un "saut de page" de la page pendant le chargement.

4.1

Vous pouvez attribuer à l'image une largeur et une hauteur correspondant au nombre de pixels CSS ou à un pourcentage des dimensions réelles de l'image.

Ces exemples sont tous valables:

```
<img src="images/50x50-img.png" width="50" height="50" alt="A 50x50 image">
<img src="images/50x50-img.png" width="200" height="200" alt="A 200x200 image">
<img src="images/200x200-img.png" width="50" height="50" alt="A 50x50 image">
<img src="images/200x200-img.png" width="50%" height="50%" alt="A 100x100 image">
```

5

La largeur et la hauteur de l'image doivent être spécifiées en pixels CSS; une valeur en pourcentage n'est plus une valeur valide. De plus, si les deux attributs sont spécifiés, ils doivent correspondre à l' une des trois formules qui conservent les proportions. Bien que valide, vous ne devez pas utiliser les attributs width et height pour étendre une image à une taille supérieure.

Ces exemples sont valides:

```
<img src="images/50x50-img.png" width="50" height="50" alt="A 50x50 image">
<img src="images/200x200-img.png" width="50" height="50" alt="A 50x50 image">
<img src="images/50x50-img.png" width="0" height="0" alt="A hidden tracking image">
```

Cet exemple n'est pas recommandé:

```
<img src="images/50x50-img.png" width="200" height="200" alt="A 200x200 image">
```

Ces exemples ne sont pas valides:

```
<img src="images/200x200-img.png" width="50%" height="50%" alt="A 100x100 image">
<img src="images/200x200-img.png" width="1" height="200" alt="A 1x200 image">
```

Choisir un texte alt

Alt-text est utilisé par les lecteurs d'écran pour les utilisateurs malvoyants et par les moteurs de recherche. Il est donc important d'écrire un bon texte alt pour vos images.

Le texte devrait être correct même si vous remplacez l'image par son attribut alt. Par exemple:

```
<!-- Incorrect -->
<img src="anonymous.png" alt="Anonymous user avatar"/> An anonymous user wrote:
<blockquote>Lorem ipsum dolor sed.</blockquote>
<a href="https://google.com/"><img src="edit.png" alt="Edit icon"/></a> /
<a href="https://google.com/"><img src="delete.png" alt="Delete icon"/></a>
```

Sans les images, cela ressemblerait à ceci:

> Anonymous user avatar Un utilisateur anonyme a écrit:
>
> > Lorem ipsum dolor sed.
>
> Icône d'édition / Icône de suppression

Pour corriger ceci:

- Supprimez le texte alternatif pour l'avatar. Cette image ajoute des informations pour les utilisateurs voyants (une icône facilement identifiable pour montrer que l'utilisateur est anonyme) mais cette information est déjà disponible dans le texte. [1]
- Supprimez l'icône du texte alt pour les icônes. Sachant que ce serait une icône si elle l'était, cela n'aide pas à transmettre son objectif réel.

```
<!-- Correct -->
<img src="anonymous.png" alt=""/> An anonymous user wrote:
<blockquote>Lorem ipsum dolor sed.</blockquote>
<a href="https://google.com/"><img src="edit.png" alt="Edit"/></a> /
<a href="https://google.com/"><img src="delete.png" alt="Delete"/></a>
```

> Un utilisateur anonyme a écrit:

Lorem ipsum dolor sed.

Editer / Supprimer

Notes de bas de page

[1] Il y a une différence sémantique entre l'inclusion d'un attribut alt vide et son exclusion totale. Un attribut alt vide indique que l'image n'est *pas* une partie clé du contenu (comme c'est le cas dans ce cas - c'est juste une image additive qui n'est pas nécessaire pour comprendre le reste) et peut donc être omis du rendu. Cependant, l'absence d'un attribut alt indique que l'image *est* un élément clé du contenu et qu'il n'y a tout simplement pas d'équivalent textuel disponible pour le rendu.

Image réactive utilisant l'attribut srcset

Utiliser srcset avec des tailles

```
<img sizes="(min-width: 1200px) 580px,
      (min-width: 640px) 48vw,
      98vw"
    srcset="img/hello-300.jpg 300w,
      img/hello-600.jpg 600w,
      img/hello-900.jpg 900w,
      img/hello-1200.jpg 1200w"
    src="img/hello-900.jpg" alt="hello">
```

`sizes` sont comme des requêtes média, décrivant l'espace occupé par l'image dans la fenêtre d'affichage.

- Si viewport est supérieur à 1200px, l'image est exactement de 580px (par exemple, notre contenu est centré dans un conteneur d'une largeur maximale de 1200px. L'image en prend la moitié moins les marges).
- Si viewport se situe entre 640px et 1200px, l'image prend 48% de la fenêtre (par exemple, l'image est mise à l'échelle avec notre page et prend la moitié de la largeur de la fenêtre moins les marges).
- Si viewport a une autre taille, dans notre cas inférieure à 640px, l'image prend 98% de la fenêtre (par exemple, l'image est mise à l'échelle avec notre page et prend toute la largeur de la fenêtre moins les marges). **La condition du support doit être omise pour le dernier élément.**

`srcset` dit simplement au navigateur quelles images sont disponibles et quelles sont leurs tailles.

- `img/hello-300.jpg` est large de 300px,
- `img/hello-600.jpg` est large de 600px,
- `img/hello-900.jpg` est large de 900px,
- `img/hello-1200.jpg` est large de 1200px

`src` est toujours une source d'image obligatoire. En cas d'utilisation avec `srcset` , `src` servira une

image de repli au cas où le navigateur ne supporterait pas `srcset` .

Utiliser srcset sans taille

```
<img src="img/hello-300.jpg" alt="hello"
     srcset="img/hello-300.jpg 1x,
       img/hello-600.jpg 2x,
       img/hello-1200.jpg 3x">
```

`srcset` fournit la liste des images disponibles, avec un rapport dispositif de pixel `x` descripteur.

- si le ratio pixel-appareil est de 1, utilisez `img/hello-300.jpg`
- si le ratio pixel-appareil est de 2, utilisez `img/hello-600.jpg`
- si le ratio pixel-appareil est de 3, utilisez `img/hello-1200.jpg`

`src` est toujours une source d'image obligatoire. En cas d'utilisation avec `srcset` , `src` servira une image de repli au cas où le navigateur ne supporterait pas `srcset` .

Image réactive utilisant un élément d'image

Code

```
<picture>
  <source media="(min-width: 600px)" srcset="large_image.jpg">
  <source media="(min-width: 450px)" srcset="small_image.jpg">
  <img src="default_image.jpg" style="width:auto;">
</picture>
```

Usage

Pour afficher des images différentes sous différentes largeurs d'écran, vous devez inclure toutes les images en utilisant l'étiquette source dans une balise image, comme illustré dans l'exemple ci-dessus.

Résultat

- Sur les écrans avec une largeur d'écran> 600px, il affiche large_image.jpg
- Sur les écrans avec une largeur d'écran supérieure à 450 pixels, on voit small_image.jpg
- Sur les écrans avec une autre largeur d'écran, il affiche default_image.jpg

Lire Images en ligne: https://riptutorial.com/fr/html/topic/587/images

Chapitre 29: Inclure le code JavaScript dans HTML

Syntaxe

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

Paramètres

Attribut	Détails
`src`	Spécifie le chemin d'accès à un fichier JavaScript. Soit une URL relative ou absolue.
`type`	Spécifie le type MIME. Cet attribut est obligatoire dans HTML4, mais facultatif dans HTML5.
`async`	Indique que le script doit être exécuté de manière asynchrone (uniquement pour les scripts externes). Cet attribut ne nécessite aucune valeur (sauf XHTML).
`defer`	Indique que le script doit être exécuté lorsque l'analyse de la page est terminée (uniquement pour les scripts externes). Cet attribut ne nécessite aucune valeur (sauf XHTML).
`charset`	Spécifie le codage de caractères utilisé dans un fichier de script externe, par exemple UTF-8
`crossorigin`	Comment l'élément gère les requêtes croisées
`nonce`	Nonce cryptographique utilisé dans *les* vérifications de la *stratégie de sécurité du contenu* CSP3

Remarques

Si le code JavaScript incorporé (fichier) est utilisé pour manipuler les éléments http://stackoverflow.com/documentation/javascript/503/document-object-model-dom , placez vos balises `<script></script>` juste **avant la fermeture `</body>` tag** ou utiliser des méthodes ou des bibliothèques JavaScript (telles que jQuery pour gérer divers navigateurs) qui permettent de vérifier que le DOM est lu et prêt à être manipulé.

Examples

Liaison à un fichier JavaScript externe

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

L'attribut `src` fonctionne comme l'attribut href sur les ancres: vous pouvez spécifier une URL absolue ou relative. L'exemple ci-dessus renvoie à un fichier situé dans le même répertoire que le document HTML. Ceci est généralement ajouté à l'intérieur des balises `<head>` en haut du document HTML

Directement incluant le code JavaScript

Au lieu de créer un lien vers un fichier externe, vous pouvez également inclure le code JS tel quel dans votre code HTML:

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

Inclure un fichier JavaScript s'exécutant de manière asynchrone

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

Handling désactivé Javascript

Il est possible que le navigateur client ne prenne pas en charge Javascript ou que l'exécution de Javascript soit désactivée, peut-être pour des raisons de sécurité. Pour pouvoir indiquer aux utilisateurs qu'un script est censé s'exécuter dans la page, la `<noscript>` peut être utilisée. Le contenu de `<noscript>` est affiché lorsque Javascript est désactivé pour la page en cours.

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

Lire Inclure le code JavaScript dans HTML en ligne:
https://riptutorial.com/fr/html/topic/3719/inclure-le-code-javascript-dans-html

Chapitre 30: Incorporer

Paramètres

Paramètres	Détails
`src`	Adresse de la ressource
`type`	Type de ressource incorporée
`width`	Dimension horizontale
`height`	Dimension verticale

Examples

Utilisation de base

La balise `embed` est nouvelle en HTML5. Cet élément fournit un point d'intégration pour une application externe (généralement non HTML) ou un contenu interactif.

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

Définir le type MIME

Le type MIME doit être défini à l'aide de l'attribut `type` .

```
<embed type="video/mp4" src="video.mp4" width="640" height="480">
```

Lire Incorporer en ligne: https://riptutorial.com/fr/html/topic/8123/incorporer

Chapitre 31: Langues du contenu

Syntaxe

- `<element lang="language_code">` <! - Le code de langue doit être au format [ISO 639-1] (https://en.wikipedia.org/wiki/ISO_639-1) ->

Remarques

La valeur de l'attribut `lang` doit être une **étiquette de langue BCP 47** valide ou la **chaîne vide** (si la langue est inconnue).

Les balises de langue BCP 47 sont répertoriées dans le registre de sous- étiquettes de langues IANA .

Accessibilité

Les critères de réussite WCAG 2.0 pertinents sont les suivants:

- 3.1.1 Langue de la page
- 3.1.2 Langue des parties

Les techniques WCAG 2.0 associées sont les suivantes:

- H57: utilisation des attributs de langage sur l'élément html
- H58: utilisation des attributs du langage pour identifier les changements dans le langage humain

Examples

Langue de l'élément

L'attribut `lang` est utilisé pour spécifier la langue du contenu de l'élément et des valeurs de texte d'attribut:

```
<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 déclaration de langage est héritée:

```
<div lang="en">
  <p>This element contains English content.</p>
  <p title="This attribute, too.">Same with this element.</p>
</div>
```

Éléments avec plusieurs langues

Vous pouvez "écraser" une déclaration de langue:

```
<p lang="en">This English sentence contains the German word <span lang="de">Hallo</span>.</p>
```

Gestion des attributs avec différentes langues

Vous pouvez "écraser" une déclaration de langue de l' élément parent en introduisant un élément en dehors de l' `applet` , la `base` , `basefont` , `br` , `frame` , `frameset` , `hr` , `iframe` , `meta` , `param` , `script` (de HTML 4.0) avec un propre `lang` attribut:

```
<p lang="en" title="An English paragraph">
    <span lang="de" title="A German sentence">Hallo Welt!</span>
</p>
```

Langue du document de base

Il est recommandé de déclarer la langue principale du document dans l'élément `html` :

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

Si aucun autre attribut `lang` n'est spécifié dans le document, cela signifie que *tout* (c'est-à-dire le contenu de l'élément et les valeurs du texte de l'attribut) est dans cette langue.

Si le document contient des parties dans d'autres langues, ces parties doivent avoir leurs propres attributs `lang` pour "écraser" la déclaration de langue.

URL régionales

Il est possible d'ajouter l'attribut `hreflang` aux éléments `<a>` et `<area>` qui créent des liens hypertexte. Cela spécifie la langue de la ressource liée. La langue définie doit être une étiquette de langue BCP 47 [1] valide.

```
<p>
    <a href="example.org" hreflang="en">example.org</a> is one of IANA's example domains.
</p>
```

---

1. ↑ Groupe de travail du réseau IETF: *Balises* RFC 5646 *pour l'identification des langues* , IETF, septembre 2009

Lire Langues du contenu en ligne: https://riptutorial.com/fr/html/topic/737/langues-du-contenu

Chapitre 32: les tables

Introduction

L'élément HTML `<table>` permet aux auteurs Web d'afficher des données tabulaires (telles que du texte, des images, des liens, d'autres tables, etc.) dans un tableau à deux dimensions comportant des lignes et des colonnes de cellules.

Syntaxe

- `<table></table>`
- `<thead></thead>`
- `<tbody></tbody>`
- `<tfoot></tfoot>`
- `<tr></tr>`
- `<th></th>`
- `<td></td>`

Remarques

Les différents éléments de la table et leurs attributs de contenu définissent ensemble le modèle de table. L'élément `<table>` est l'élément conteneur pour les modèles de tableau / données tabulaires. Les tableaux ont des lignes, des colonnes et des cellules données par leurs descendants. Les lignes et les colonnes forment une grille; Les cellules d'une table doivent couvrir complètement cette grille sans se chevaucher. La liste ci-dessous décrit les différents éléments du modèle de tableau:

- `<table>` - Elément conteneur pour les modèles de tableau / données tabulaires. `<table>` représente des données avec plusieurs dimensions sous la forme d'une table.
- `<caption>` - Légende ou titre du tableau (comme une `figcaption` sur une `figure`)
- `<col>` - Une colonne (un élément sans contenu)
- `<colgroup>` - Un regroupement de colonnes
- `<thead>` - En-tête de tableau (un seul)
- `<tbody>` - Corps / contenu de la table (plusieurs sont corrects)
- `<tfoot>` - `<tfoot>` de page (un seul)
- `<tr>` - Rangée de table
- `<th>` - Cellule d'en-tête de table
- `<td>` - Cellule de données de table

Sémantiquement, les tableaux sont destinés à contenir des données tabulaires. Vous pouvez le voir comme un moyen d'afficher et de décrire des données qui auraient du sens dans une feuille de calcul (colonnes et lignes).

L'utilisation de tableaux pour la mise en page n'est pas recommandée. Au lieu de cela, utilisez les règles CSS pour la mise en page et le formatage, y compris l' `display: table` .

Une exception notable dans l'industrie concernant l'utilisation de la mise en page `<table>` concerne le courrier électronique HTML: certains clients de messagerie, y compris Outlook, sont revenus aux anciens moteurs de rendu après que Microsoft a perdu son monopole par rapport à l'UE. Afin que Microsoft ne fasse pas partie d'IE, ils ont simplement restauré le moteur de rendu Outlook dans une version antérieure de Trident. Ce retour en arrière ne prend tout simplement pas en charge les technologies Web modernes. L'utilisation de mises en page basées sur `<table>` pour les courriers électroniques HTML est la seule façon de garantir la compatibilité entre les navigateurs, les plates-formes et les clients.

Examples

Tableau simple

```
<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>
```

Cela rendra une `<table>` composée de trois lignes au total (`<tr>`): une ligne de cellules d'en-tête (`<th>`) et deux lignes de cellules de contenu (`<td>`). `<th>` éléments `<th>` sont *des en-têtes tabulaires* et les éléments `<td>` sont *des données tabulaires* . Vous pouvez mettre ce que vous voulez dans un `<td>` ou `<th>` .

Rubrique 1 / Colonne 1	Rubrique 2 / Colonne 2
Rangée 1 Données Colonne 1	Rangée 1 Données Colonne 2
Ligne 2 Données Colonne 1	Ligne 2 Données Colonne 2

Répartir des colonnes ou des lignes

Les cellules de tableau peuvent couvrir plusieurs colonnes ou lignes à l'aide des attributs `colspan` et `rowspan` . Ces attributs peuvent être appliqués aux éléments `<th>` et `<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>
```

Aura pour résultat

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

Notez que vous ne devez pas concevoir une table où les lignes et les colonnes se chevauchent car il ne s'agit pas d'un code HTML valide et que le résultat est traité différemment par les différents navigateurs Web.

`rowspan` = Un entier non négatif qui spécifie le nombre de lignes couvertes par une cellule. La valeur par défaut de cet attribut est un (`1`). Une valeur de zéro (`0`) signifie que la cellule s'étendra de la ligne en cours jusqu'à la dernière ligne de la table (`<thead>` , `<tbody>` ou `<tfoot>`).

`colspan` = Un entier non négatif qui spécifie le nombre de colonnes couvertes par la cellule en cours. La valeur par défaut de cet attribut est un (`1`). Une valeur de zéro (`0`) signifie que la cellule s'étendra du courant à la dernière colonne du groupe de colonnes `<colgroup>` dans lequel la cellule est définie.

Tableau avec thead, tbody, tfoot et légende

HTML fournit également les tables avec les éléments `<thead>` , `<tbody>` , `<tfoot>` et `<caption>` . Ces éléments supplémentaires sont utiles pour ajouter de la valeur sémantique à vos tables et pour fournir une place à un style CSS distinct.

Lorsque vous imprimez un tableau qui ne rentre pas sur une seule page (papier), la plupart des navigateurs répètent le contenu de `<thead>` sur chaque page.

Il y a un ordre spécifique à respecter, et nous devrions être conscients que tous les éléments ne sont pas en place comme on pourrait s'y attendre. L'exemple suivant montre comment nos 4 éléments doivent être placés.

```
<table>
 <caption>Table Title</caption> <!--| caption is the first child of table |-->
```

```
  <thead> <!--======================| thead 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>
```

Les résultats de l'exemple suivant sont démontrés deux fois: la première table ne contient aucun style, la deuxième table a quelques propriétés CSS appliquées: `background-color` , `color` et `border` *. Les styles sont fournis à titre de guide visuel et ne constituent pas un aspect essentiel du sujet traité.

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

Élément	Styles S'applique
`<caption>`	Texte jaune sur fond noir.
`<thead>`	Texte en gras sur fond violet.
`<tbody>`	Texte sur fond bleu.
`<tfoot>`	Texte sur fond vert.
`<th>`	Frontières orange

Élément	Styles S'applique
`<td>`	Les frontières rouges

Groupes de colonnes

Vous souhaiterez parfois appliquer un style à une colonne ou à un groupe de colonnes. Ou à des fins sémantiques, vous pouvez souhaiter regrouper des colonnes. Pour ce faire, utilisez les éléments `<colgroup>` et `<col>` .

La `<colgroup>` facultative `<colgroup>` vous permet de regrouper des colonnes. `<colgroup>` éléments `<colgroup>` doivent être des éléments enfants d'un `<table>` et doivent venir après tout élément `<caption>` et avant tout contenu de table (par exemple, `<tr>` , `<thead>` , `<tbody>` , etc.).

```
<table>
    <colgroup span="2"></colgroup>
    <colgroup span="2"></colgroup>
    ...
</table>
```

La `<col>` facultative `<col>` vous permet de référencer des colonnes individuelles ou une plage de colonnes sans appliquer un regroupement logique. `<col>` éléments `<col>` sont facultatifs, mais s'ils sont présents, ils doivent se trouver dans un élément `<colgroup>` .

```
<table>
    <colgroup>
        <col id="MySpecialColumn" />
        <col />
    </colgroup>
    <colgroup>
        <col class="CoolColumn" />
        <col class="NeatColumn" span="2" />
    </colgroup>
    ...
</table>
```

Les styles CSS suivants peuvent être appliqués aux éléments `<colgroup>` et `<col>` :

- `border`
- `background`
- `width`
- `visibility`
- `display` (comme dans l' `display: none`)
    - `display: none;` va effectivement supprimer les colonnes de l'affichage, provoquant le rendu de la table comme si ces cellules n'existaient pas

Pour plus d'informations, voir Données tabulaires HTML5 .

Champ d'application

`th` éléments sont très couramment utilisés pour indiquer rubriques pour les lignes de table et colonnes, comme suit:

```
<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>
```

Cela peut être amélioré pour l'accessibilité en utilisant l'attribut `scope` . L'exemple ci-dessus serait modifié comme suit:

```
<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` est un *attribut énuméré* , ce qui signifie qu'il peut avoir une valeur à partir d'un ensemble spécifique de valeurs possibles. Cet ensemble comprend:

- `col`
- `row`

- `colgroup`
- `rowgroup`

Les références:

- https://developer.mozilla.org/en-US/docs/Web/HTML/Element/th#attr-scope
- https://www.w3.org/TR/WCAG20-TECHS/H63.html

Lire les tables en ligne: https://riptutorial.com/fr/html/topic/274/les-tables

Chapitre 33: Les titres

Introduction

HTML fournit non seulement des balises de paragraphe simples, mais six balises d'en-tête distinctes pour indiquer les en-têtes de différentes tailles et épaisseurs. Sous la dénomination de 1 à 6, le titre 1 a le texte le plus grand et le plus épais, tandis que le titre 6 est le plus petit et le plus fin, jusqu'au niveau du paragraphe. Cette rubrique détaille l'utilisation correcte de ces balises.

Syntaxe

- `<h1>...</h1>`
- `<h2>...</h2>`
- `<h3>...</h3>`
- `<h4>...</h4>`
- `<h5>...</h5>`
- `<h6>...</h6>`

Remarques

- Un élément `h1` - `h6` doit avoir à la fois une balise de début et une balise de fin. [1]
- `h1` éléments `h1` - `h6` sont des éléments de niveau bloc par défaut (style CSS: `display: block`). [2]
- `h1` éléments `h1` - `h6` ne doivent pas être confondus avec l' élément section
- Les balises d'en-tête (`h1` - `h6`) ne sont pas liées à la balise `head` .
- Contenu autorisé: contenu du texte
- Les différents styles CSS pour les en-têtes diffèrent généralement en `font-size` et de `margin` . Les paramètres CSS suivants pour les éléments `h1` - `h6` peuvent servir d'orientation (caractérisé comme «informatif» par le W3C)
- Les moteurs de recherche (le code qui ajoute une page à un moteur de recherche) accorde automatiquement plus d'attention à une importance plus élevée (h1 a le plus, h2 a moins, h3 a encore moins, ...) pour discerner la nature d'une page.

Examples

Utiliser les en-têtes

Les en-têtes peuvent être utilisés pour décrire le sujet qu'ils précèdent et ils sont définis avec les balises `<h1>` à `<h6>` . Les en-têtes prennent en charge tous les attributs globaux .

- `<h1>`

définit l'en-tête le plus important.
- `<h6>` définit l'en-tête le moins important.

Définir un titre:

```
<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 structure correcte compte

Les moteurs de recherche et autres **agents utilisateurs** indexent généralement le contenu de la page en fonction d'éléments de titre, par exemple pour créer une table des matières. Il est donc important d'utiliser la structure appropriée pour les en-têtes.

En général, un article devrait avoir un élément `h1` pour le titre principal suivi des sous-titres `h2` - en descendant d'une couche si nécessaire. S'il y a des éléments `h1` à un niveau supérieur, ils ne devraient pas être utilisés pour décrire un contenu de niveau inférieur.

Exemple de document (intention supplémentaire pour illustrer la hiérarchie):

```
<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>
```

Lire Les titres en ligne: https://riptutorial.com/fr/html/topic/226/les-titres

Chapitre 34: Marquage du code informatique

Syntaxe

- `<pre>Formatted text</pre>`
- `<code>Inline Code</code>`

Remarques

L'élément de `code` doit être utilisé pour tout type de "chaîne qu'un ordinateur reconnaîtrait" (HTML5), par exemple:

- code source
- termes de balisage / langages de programmation (noms d'éléments, noms de fonctions, etc.)
- noms de fichiers

Éléments connexes

Pour les variables, l' élément `var` peut être utilisé.

Pour la sortie de l'ordinateur, l' élément `samp` peut être utilisé.

Pour l'entrée utilisateur, l' élément `kbd` peut être utilisé.

Examples

En ligne avec

Si une phrase contient du code informatique (par exemple, le nom d'un élément HTML), utilisez l'élément de `code` pour le marquer:

```
<p>The <code>a</code> element creates a hyperlink.</p>
```

Bloquer avec

```
 et
```

Si la mise en forme (espace blanc, nouvelles lignes, indentation) du code est importante, utilisez l'élément `pre` en combinaison avec l'élément de `code` :

```
<pre>
    <code>
    x = 42
    if x == 42:
        print "x is …           … 42"
    </code>
```

```
</pre>
```

Vous devez toujours échapper des caractères avec une signification spéciale en HTML (comme `<` avec `<`), donc pour *afficher* un bloc de code HTML (`<p>This is a paragraph.</p>`), cela pourrait ressembler à ceci:

```
<pre>
    <code>
    &lt;p>This is a paragraph.&lt;/p>
    </code>
</pre>
```

Lire Marquage du code informatique en ligne: https://riptutorial.com/fr/html/topic/1836/marquage-du-code-informatique

Chapitre 35: Méta-information

Introduction

Les balises META dans les documents HTML fournissent des informations utiles sur le document, notamment une description, des mots-clés, un auteur, des dates de modification et environ 90 autres champs. Cette rubrique couvre l'utilisation et le but de ces balises.

Syntaxe

- `<meta name="metadata name" content="value">`
- `<meta http-equiv="pragma directive" content="value">`
- `<meta charset="encoding label">`

Remarques

La balise meta est une balise HTML utilisée pour définir les métadonnées du document HTML. Les balises META doivent aller dans l'élément tête. Une page peut avoir un nombre quelconque de balises META.

Les `keywords` - `keywords` balise `keywords` ne sont généralement pas utilisés par les robots. La plupart des moteurs de recherche déterminent quels mots clés correspondent au contenu des pages Web. Cela étant dit, rien ne dit que vous ne devriez plus inclure la balise META de mots-clés.

Les métadonnées d'une page sont principalement utilisées par le navigateur (comme la mise à l'échelle d'un document) et les robots d'exploration utilisés par les moteurs de recherche (Google, Yahoo !, Bing).

La spécification donne un certain nombre de noms de métadonnées normalisés à utiliser avec les directives `<meta name>` et les directives de métadonnées normalisées pragma à utiliser avec `<meta http-equiv>` . Cependant, de nombreux services sur Internet (robots d'exploration Web, outils de création, services de partage social, etc.) utilisent le formulaire `<meta name>` comme point d'extension générique pour les métadonnées. Certains d'entre eux sont listés sur la page wiki de spec .

Examples

Encodage de caractère

L'attribut `charset` spécifie le codage des caractères du document HTML et doit être un codage de caractères valide (par exemple, `windows-1252` , `ISO-8859-2` , `Shift_JIS` et `UTF-8`). `UTF-8` (Unicode) est le plus utilisé et doit être utilisé pour tout nouveau projet.

5

```
<meta charset="UTF-8">
```

```
<meta charset="ISO-8859-1">
```

Tous les navigateurs ont toujours reconnu le formulaire `<meta charset>` , mais si, pour une raison quelconque, votre page doit être valide au format HTML 4.01, vous pouvez utiliser les options suivantes:

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

```
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
```

Reportez-vous également à la norme de codage pour afficher toutes les étiquettes de codage de caractères disponibles reconnues par les navigateurs.

Actualisation automatique

Pour actualiser la page toutes les cinq secondes, ajoutez ce `meta` élément dans l'élément `head` :

```
<meta http-equiv="refresh" content="5">
```

MISE EN GARDE! Bien qu'il s'agisse d'une commande valide, il est recommandé de ne pas l'utiliser en raison de ses effets négatifs sur l'expérience utilisateur. Si vous actualisez trop souvent la page, celle-ci risque de ne plus répondre et défile souvent vers le haut de la page. Si certaines informations de la page doivent être mises à jour en permanence, il existe de bien meilleures façons de le faire en actualisant uniquement une partie de la page.

Contrôle de disposition mobile

Les sites courants optimisés pour les mobiles utilisent la `<meta name="viewport">` comme ceci:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

L'élément `viewport` donne au navigateur des instructions sur la façon de contrôler les dimensions et la mise à l'échelle de la page en fonction du périphérique que vous utilisez.

Dans l'exemple ci-dessus, `content="width=device-width` signifie que le navigateur affichera la largeur de la page à la largeur de son propre écran. Donc, si cet écran est `480px wide` , la fenêtre du navigateur aura une `480px wide initial-scale=1` . `initial-scale=1` indique que le zoom initial (1 dans ce cas signifie qu'il ne fait pas de zoom).

Voici les attributs que cette balise prend en charge:

Attribut	La description
`width`	La largeur de la fenêtre virtuelle de l'appareil. Valeurs [1] : `device-width` ou la largeur réelle en pixels, comme `480`

Attribut	La description
`height`	La hauteur de la fenêtre virtuelle de l'appareil. Valeurs [2] : `device-height` ou largeur réelle en pixels, par exemple `600`
`initial-scale`	Le zoom initial lorsque la page est chargée. 1.0 ne fait pas de zoom
`minimum-scale`	Le montant minimum que le visiteur peut zoomer sur la page. `1.0` ne fait pas de zoom
`maximum-scale`	Le montant maximum que le visiteur peut zoomer sur la page. `1.0` ne fait pas de zoom
`user-scalable`	Permet à l'appareil de zoomer et dézoomer. Les valeurs sont `yes` ou `no` . Si ce paramètre est défini sur no, l'utilisateur ne peut pas zoomer sur la page Web. La valeur par défaut est yes. Les paramètres du navigateur peuvent ignorer cette règle.

Remarques:

[1] La propriété `width` peut être spécifiée en *pixels* (`width=600`) ou en *device-width* (`width=device-width`), qui représente la largeur physique de l'écran du périphérique.

[2] De même, la propriété `height` peut être spécifiée en `pixels` (`height=600`) ou en `device-height` de `device-height` (`height=device-height`) qui représente la hauteur physique de l'écran de l'appareil.

Informations sur la page

`application-name`

Donner le nom de l'application Web que représente la page.

```
<meta name="application-name" content="OpenStreetMap">
```

Si ce n'est pas une application Web, la balise META `application-name` ne doit pas être utilisée.

`author`

Définissez l'auteur de la page:

```
<meta name="author" content="Your Name">
```

Un seul nom peut être donné.

`description`

Définissez la description de la page:

```
<meta name="description" content="Page Description">
```

La balise META de `description` peut être utilisée par divers moteurs de recherche lors de l'indexation de votre page Web à des fins de recherche. Habituellement, la description contenue dans la balise META est le bref résumé qui apparaît sous le titre principal de la page / du site Web dans les résultats du moteur de recherche. Google n'utilise généralement que les 20-25 premiers mots de votre description.

generator

```
<meta name="generator" content="HTML Generator 1.42">
```

Identifie l'un des progiciels utilisés pour générer le document. A utiliser uniquement pour les pages où le balisage est généré automatiquement.

keywords

Définissez des mots clés pour les moteurs de recherche (séparés par des virgules):

```
<meta name="keywords" content="Keyword1, Keyword2">
```

La balise META de `keywords` est parfois utilisée par les moteurs de recherche pour connaître la requête de recherche qui concerne votre page Web.
En règle générale, il est préférable de ne pas ajouter trop de mots, car la plupart des moteurs de recherche utilisant cette balise META pour l'indexation indexeront uniquement les premiers 20 mots. Assurez-vous de mettre les mots-clés les plus importants en premier.

Des robots

L'attribut `robots` , pris en charge par plusieurs moteurs de recherche principaux, détermine si les moteurs de recherche sont autorisés à indexer une page ou non et s'ils doivent suivre les liens d'une page ou non.

```
<meta name="robots" content="noindex">
```

Cet exemple indique à tous les moteurs de recherche de ne pas afficher la page dans les résultats de recherche. Les autres valeurs autorisées sont les suivantes:

Valeur / Directive	Sens
`all`	**Défaut.** Équivalent à `index, follow` . Voir note ci-dessous.
`noindex`	Ne pas indexer la page du tout.
`nofollow`	Ne suivez pas les liens sur cette page
`follow`	Les liens sur la page peuvent être suivis. Voir note ci-dessous.
`none`	Équivalent à `noindex, nofollow` .

Valeur / Directive	Sens
`noarchive`	Ne mettez pas une version en cache de cette page disponible dans les résultats de recherche.
`nocache`	Synonyme de `noarchive` utilisé par certains robots tels que Bing.
`nosnippet`	Ne pas afficher un extrait de cette page dans les résultats de recherche.
`noodp`	N'utilisez pas les métadonnées de cette page à partir du projet Open Directory pour les titres ou les extraits dans les résultats de recherche.
`notranslate`	Ne pas proposer de traductions de cette page dans les résultats de recherche.
`noimageindex`	Ne pas indexer les images sur cette page.
`unavailable_after [RFC-850 date/time]`	Ne pas afficher cette page dans les résultats de recherche après la date / heure spécifiée. La date / heure doit être spécifiée au format RFC 850 .

Remarque: La définition explicite de l' `index` et / ou du `follow` , alors que les valeurs valides, n'est pas nécessaire, car pratiquement tous les moteurs de recherche supposent qu'ils sont autorisés à le faire s'ils n'en sont pas explicitement empêchés. À l'instar du fichier robots.txt, les moteurs de recherche ne recherchent généralement que ce qu'ils *ne* sont *pas autorisés* à faire. Seules les déclarations auxquelles un moteur de recherche n'est pas autorisé empêchent également de déclarer des oppositions accidentelles (telles que `index, ..., noindex`) que tous les moteurs de recherche ne traiteront pas de la même manière.

Reconnaissance du numéro de téléphone

Les plates-formes mobiles comme iOS reconnaissent automatiquement les numéros de téléphone et les transforment en `tel:` liens. Bien que cette fonctionnalité soit très pratique, le système détecte parfois les codes ISBN et autres numéros comme numéros de téléphone.

Pour que Safari mobile et certains autres navigateurs mobiles basés sur WebKit désactivent la reconnaissance et le formatage automatiques des numéros de téléphone, vous avez besoin de cette balise META:

```
<meta name="format-detection" content="telephone=no">
```

Des médias sociaux

Open Graph est une norme pour les métadonnées qui étend les informations normales contenues dans le balisage de tête d'un site. Cela permet aux sites Web tels que Facebook d'afficher des informations plus détaillées et plus riches sur un site Web dans un format structuré. Ces

informations sont alors automatiquement affichées lorsque les utilisateurs partagent des liens vers des sites Web contenant des métadonnées OG sur Facebook.

Facebook / Open Graph

```
<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
- Protocole Open Graph

Facebook / Articles instantanés

```
<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">
```

- Articles instantanés sur Facebook: création d'articles
- Articles instantanés: Référence du format

Twitter utilise son propre balisage pour les métadonnées. Ces métadonnées sont utilisées comme informations pour contrôler la façon dont les tweets sont affichés lorsqu'ils contiennent un lien vers le site.

Gazouillement

```
<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">
```

- Cartes Twitter: Guide de démarrage
- Validateur de carte Twitter

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">
```

Redirection automatique

Parfois, votre page Web nécessite une redirection automatique.

Par exemple, pour rediriger vers `example.com` après 5 secondes:

```
<meta http-equiv="refresh" content="5;url=https://www.example.com/" />
```

Ceci est la ligne vous enverra sur le site Web désigné (dans ce cas `example.com` après 5 secondes.

Si vous avez besoin de changer le délai avant une redirection, il vous suffit de changer le numéro juste avant votre `;url=` cela modifiera le délai.

Application Web

Vous pouvez configurer votre application Web ou votre site Web pour ajouter une icône de raccourci d'application à l'écran d'accueil d'un appareil et lancer l'application en "mode application" plein écran en utilisant l'élément de menu "Ajouter à l'écran d'accueil" de Chrome for Android.

La balise meta ci-dessous ouvrira l'application Web en mode plein écran (sans barre d'adresse).

Android Chrome

```
<meta name="mobile-web-app-capable" content="yes">
```

IOS

```
<meta name="apple-mobile-web-app-capable" content="yes">
```

Vous pouvez également définir la couleur de la barre d'état et de la barre d'adresse dans la balise meta.

Android Chrome

```
<meta name="theme-color" content="black">
```

IOS

```
<meta name="apple-mobile-web-app-status-bar-style" content="black">
```

Lire Méta-information en ligne: https://riptutorial.com/fr/html/topic/1264/meta-information

Chapitre 36: Paragraphes

Introduction

Les paragraphes sont l'élément HTML le plus élémentaire. Cette rubrique explique et illustre l'utilisation de l'élément de paragraphe en HTML.

Paramètres

Colonne	Colonne
`<p>`	Définit un paragraphe
` `	Insère un seul saut de ligne
`<pre>`	Définit le texte pré-formaté

Examples

HTML paragraphes

L'élément HTML `<p>` définit un **paragraphe** :

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

Afficher-

Vous ne pouvez pas être sûr de la manière dont HTML sera affiché.

De grands ou petits écrans et des fenêtres redimensionnées créeront des résultats différents.

Avec HTML, vous ne pouvez pas modifier la sortie en ajoutant des espaces supplémentaires ou des lignes supplémentaires dans votre code HTML.

Le navigateur supprimera tous les espaces supplémentaires et les lignes supplémentaires lorsque la page est affichée:

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

Lire Paragraphes en ligne: https://riptutorial.com/fr/html/topic/7997/paragraphes

Chapitre 37: Relier des ressources

Introduction

Bien que de nombreux scripts, icônes et feuilles de style puissent être écrits directement dans le balisage HTML, il est préférable et plus efficace d'inclure ces ressources dans leur propre fichier et de les lier à votre document. Cette rubrique traite de la liaison de ressources externes telles que les feuilles de style et les scripts dans un document HTML.

Syntaxe

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

Paramètres

Attribut	Détails
`charset`	Spécifie le codage des caractères du document lié
`crossorigin`	Spécifie comment l'élément gère les demandes d'origine croisée
`href`	Spécifie l'emplacement du document lié
`hreflang`	Spécifie la langue du texte dans le document lié
`media`	Spécifie sur quel périphérique le document lié sera affiché, souvent utilisé avec la sélection de feuilles de style en fonction du périphérique en question
`rel`	**Requis** Spécifie la relation entre le document actuel et le document lié
`rev`	Spécifie la relation entre le document lié et le document en cours
`sizes`	Spécifie la taille de la ressource liée. Seulement quand `rel="icon"`
`target`	Spécifie où le document lié doit être chargé
`type`	Spécifie le type de support du document lié
`integrity`	Spécifie un hachage codé en base64 (sha256, sha384 ou sha512) de la ressource liée permettant au navigateur de vérifier sa légitimité.

Examples

Feuille de style CSS externe

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

La pratique standard consiste à placer des balises CSS `<link>` dans la `<head>` en haut de votre code HTML. De cette façon, le CSS sera chargé en premier et s'appliquera à votre page au fur et à mesure de son chargement, au lieu d'afficher le code HTML non stylé jusqu'à ce que le CSS soit chargé. L'attribut de `type` n'est pas nécessaire en HTML5, car HTML5 prend généralement en charge CSS.

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

et

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

... faire la même chose en HTML5.

Une autre pratique, quoique moins courante, consiste à utiliser une instruction `@import` à l'intérieur de CSS direct. Comme ça:

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

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

JavaScript

Synchrone

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

La pratique standard consiste à placer des balises JavaScript `<script>` juste avant la `</body>` fermeture. Le chargement de vos scripts en dernier permet aux images de votre site de s'afficher plus rapidement et décourage votre JavaScript d'essayer d'interagir avec des éléments qui ne sont pas encore chargés.

Asynchrone

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

Une autre alternative, lorsque le code Javascript en cours de chargement n'est pas nécessaire pour l'initialisation de la page, peut être chargé de manière asynchrone, ce qui accélère le chargement de la page. En utilisant `async` le navigateur charge le contenu du script en parallèle et,

une fois téléchargé, interrompt l'analyse HTML pour analyser le fichier Javascript.

Différé

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

Les scripts différés sont comme les scripts asynchrones, à l'exception du fait que l'analyse ne sera effectuée qu'une fois que le code HTML sera entièrement analysé. Les scripts différés sont garantis être chargés dans l'ordre de déclaration, de la même manière que les scripts synchrones.

<noscript>

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

L'élément `<noscript>` définit le contenu à afficher si l'utilisateur a désactivé les scripts ou si le navigateur ne prend pas en charge l'utilisation de scripts. La `<noscript>` peut être placée dans `<head>` ou `<body>` .

Favicon

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

Utilisez l' `image/png` type mime pour les fichiers PNG et l' `image/x-icon` pour les fichiers d'icône (`*.ico`). Pour la différence, voir cette question SO .

Un fichier nommé `favicon.ico` à la racine de votre site Web sera généralement chargé et appliqué automatiquement, sans nécessiter de `<link>` . Si ce fichier change, les navigateurs peuvent être lents et obstinés à mettre à jour leur cache.

CSS alternatif

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

Certains navigateurs permettent d'appliquer d'autres feuilles de style si elles sont proposées. Par défaut, ils ne seront pas appliqués, mais ils peuvent généralement être modifiés via les paramètres du navigateur:

> Firefox permet à l'utilisateur de sélectionner la feuille de style en utilisant le sous-menu Affichage> Style de page, Internet Explorer prend également en charge cette fonctionnalité (commençant par IE 8), également accessible depuis View> Style de page (au moins depuis IE 11). utilisez la fonctionnalité (à partir de la version 48). La page Web peut également fournir sa propre interface utilisateur pour permettre à l'utilisateur de changer de style.

(Source: les documents MDN)

Flux Web

Utilisez l' `rel="alternate"` pour permettre la découverte de vos flux 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" />
```

Consultez la documentation MDN pour les flux RSS et Atomic RSS .

Attribut "média" de lien

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

Le support spécifie quelle feuille de style doit être utilisée pour quel type de support. L'utilisation de la valeur d' `print` n'afficherait que cette feuille de style pour les pages imprimées.

La valeur de cet attribut peut être l'une des valeurs du type de `mediatype` (similaire à une requête de support CSS).

Précédent et Suivant

Lorsqu'une page fait partie d'une série d'articles, par exemple, on peut utiliser `prev` et à `next` pour pointer vers des pages qui arrivent avant et après.

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

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

Conseil de ressource: dns-prefetch, prefetch, prerender

Préconnecter

La relation `preconnect` est similaire à `dns-prefetch` car elle résout le DNS. Cependant, cela permettra également d'établir le protocole TCP et la négociation TLS facultative. Ceci est une fonctionnalité expérimentale.

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

DNS-Prefetch

Informe les navigateurs de la résolution du DNS pour une URL, afin que tous les actifs de cette URL se chargent plus rapidement.

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

Prélecture

Informe les navigateurs qu’une ressource donnée doit être pré-extraite afin qu’elle puisse être chargée plus rapidement.

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

DNS-Prefetch résout uniquement le nom de domaine, tandis que prefetch télécharge / stocke les ressources spécifiées.

Prérendeur

Informe les navigateurs de l'extraction et du rendu de l'URL en arrière-plan, afin qu'ils puissent être livrés instantanément à l'utilisateur lorsque l'utilisateur accède à cette URL. Ceci est une fonctionnalité expérimentale.

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

Lire Relier des ressources en ligne: https://riptutorial.com/fr/html/topic/712/relier-des-ressources

Chapitre 38: SVG

Introduction

SVG signifie Scalable Vector Graphics. SVG est utilisé pour définir des graphiques pour le Web

L'élément HTML `<svg>` est un conteneur pour les graphiques SVG.

SVG dispose de plusieurs méthodes pour dessiner des chemins, des boîtes, des cercles, du texte et des images graphiques.

Remarques

SVG est un langage basé sur XML pour créer des images vectorielles évolutives. Il peut être écrit directement dans un document HTML ou intégré à partir de fichiers SVG externes. SVG en ligne peut être restylé et modifié en utilisant CSS et JavaScript respectivement.

Le support du navigateur pour SVG varie, mais peut être vérifié ici .

Pour plus d'informations, consultez la documentation SVG .

Examples

Incorporation de fichiers SVG externes en HTML

Vous pouvez utiliser les éléments `<img>` ou `<object>` pour incorporer des éléments SVG externes. Le réglage de la hauteur et de la largeur est facultatif mais fortement recommandé.

Utiliser l'élément image

```
<img src="attention.svg" width="50" height="50">
```

L'utilisation de `<img>` ne vous permet pas de styliser le SVG en utilisant CSS ou de le manipuler en utilisant JavaScript.

Utiliser l'élément object

```
<object type="image/svg+xml" data="attention.svg" width="50" height="50">
```

Contrairement à `<img>` , `<object>` importe directement le SVG dans le document et peut donc être manipulé avec Javascript et CSS.

SVG en ligne

SVG peut être écrit directement dans un document HTML. SVG en ligne peut être stylé et manipulé en utilisant CSS et 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>
```

Le SVG ci-dessus peut alors être stylé en utilisant la classe CSS correspondante:

```
.attention {
    fill: red;
    width: 50px;
    height: 50px;
}
```

Le résultat ressemble à ceci:

Intégration de SVG en CSS

Vous pouvez ajouter des fichiers SVG externes en utilisant la propriété `background-image` , comme vous le feriez avec toute autre image.

HTML:

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

CSS:

```
.attention {
    background-image: url(attention.svg);
    background-size: 100% 100%;
    width: 50px;
    height: 50px;
}
```

Vous pouvez également intégrer l'image directement dans un fichier CSS à l'aide d'une URL de données:

```
background-image:
url(data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsvg%22%20xmlns%3Axlink%3D%2
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);
```

Lire SVG en ligne: https://riptutorial.com/fr/html/topic/1183/svg

Chapitre 39: Tabindex

Paramètres

Valeur	Sens
négatif	l'élément sera focalisable, mais il ne devrait pas être accessible via la navigation séquentielle au clavier
0	l'élément sera focalisable et accessible via la navigation séquentielle au clavier, mais son ordre relatif est défini par la convention de la plateforme
positif	l'élément doit pouvoir être focalisé et accessible via la navigation séquentielle au clavier; son ordre relatif sera défini par la valeur de l'attribut: le séquentiel suit le nombre croissant du `tabindex`

Remarques

La valeur maximale de `tabindex` ne doit pas dépasser 32767, conformément à la section 17.11.1 du W3C. Sauf si la valeur par défaut spécifiée est -1

Un élément avec une valeur de 0, une valeur non valide ou aucune valeur `tabindex` doit être placé après les éléments avec un index positif dans l'ordre séquentiel de la navigation au clavier.

Examples

Ajouter un élément à l'ordre de tabulation

```
<div tabindex="0">Some button</div>
```

Remarque: Essayez d'utiliser un HTML natif `button` ou `a` étiquette , le cas échéant.

Supprimer un élément de l'ordre de tabulation

```
<button tabindex="-1">This button will not be reachable by tab</button>
```

L'élément sera supprimé de l'ordre de tabulation mais sera toujours focalisable.

Définir un ordre de tabulation personnalisé (non recommandé)

```
<div tabindex="2">Second</div>
<div tabindex="1">First</div>
```

Les valeurs positives insèrent l'élément à la position de tabulation de sa valeur respective. Les éléments sans préférence (c.-à-d. `tabindex="0"` ou les éléments natifs tels que le `button` et `a`) seront ajoutés après ceux qui ont la préférence.

Les valeurs positives ne sont **pas recommandées** car elles perturbent le comportement attendu de la tabulation et peuvent induire en erreur les utilisateurs de lecteurs d’écran. Essayez de créer un ordre naturel en réorganisant votre structure DOM.

Lire Tabindex en ligne: https://riptutorial.com/fr/html/topic/2594/tabindex

Chapitre 40: Toile

Paramètres

Attribut	La description
la taille	Spécifie la hauteur du canevas
largeur	Spécifie la largeur du canevas

Remarques

- Cette balise n'est pas compatible avec les versions d'Internet Explorer inférieures à 9. Vérifiez la compatibilité de votre navigateur avec caniuse.com .
- `canvas` n'est qu'un conteneur pour les graphiques, et le dessin des graphiques est effectué par JavaScript.

Examples

Exemple de base

L'élément `canvas` été introduit en HTML5 pour dessiner des graphiques.

```
<canvas id="myCanvas">
   Cannot display graphic. Canvas is not supported by your browser (IE<9)
</canvas>
```

Ce qui précède va créer un élément HTML `<canvas>` transparent de 300 × 150 px.

Vous pouvez utiliser l'élément **canvas** pour dessiner des éléments étonnants tels que des formes, des graphiques, manipuler des images, créer des jeux attrayants, etc. avec **JavaScript** .
Les `canvas` 2D d » surface *de* la *couche étirable* objet est appelé `CanvasRenderingContext2D` ; ou à partir d'un `HTMLCanvasElement` utilisant la `.getContext("2d")` :

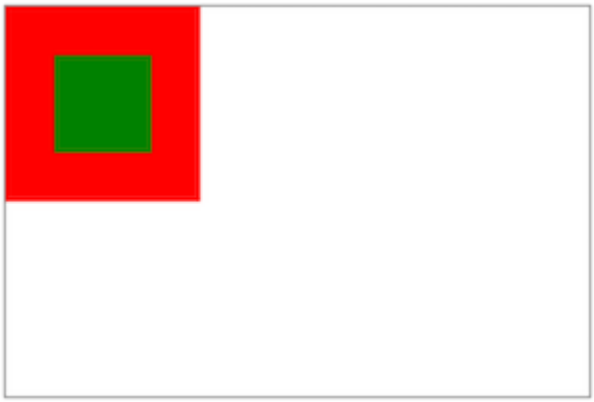
```
var ctx = document.getElementById("myCanvas").getContext("2d");
// now we can refer to the canvas's 2D layer context using `ctx`

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
```

jsFiddle exemple

Dessin de deux rectangles sur un

```
<!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>
```

Cet exemple ressemble à ceci:



Lire Toile en ligne: https://riptutorial.com/fr/html/topic/1162/toile

Chapitre 41: Utiliser HTML avec CSS

Introduction

CSS fournit des styles aux éléments HTML sur la page. Le style en ligne implique l'utilisation de l'attribut de style dans les balises et est fortement déconseillé. Les feuilles de style internes utilisent la `<style>` et sont utilisées pour déclarer des règles pour les parties dirigées de la page. Les feuilles de style externes peuvent être utilisées via une `<link>` qui prend un fichier externe de CSS et applique les règles au document. Cette rubrique couvre l'utilisation des trois méthodes de pièce jointe.

Syntaxe

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

Examples

Utilisation de la feuille de style externe

Utilisez l'attribut de `link` dans la `head` du document:

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

Vous pouvez également utiliser des feuilles de style fournies par des sites Web via un réseau de diffusion de contenu, ou CDN. (par exemple, Bootstrap):

```
<head>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
</head>
```

En règle générale, vous pouvez trouver un support CDN pour un framework sur son site Web.

Feuille de style interne

Vous pouvez également inclure des éléments CSS en interne en utilisant la `<style>` :

```
<head>
    <style type="text/css">
        body {
           background-color: gray;
        }
```

```
    </style>
</head>
```

Plusieurs feuilles de style internes peuvent également être incluses dans un programme.

```
<head>
    <style type="text/css">
        body {
           background-color: gray;
        }
    </style>

    <style type="text/css">
        p {
           background-color: blue;
        }
    </style>
</head>
```

Style en ligne

Vous pouvez styliser un élément spécifique en utilisant l'attribut de `style` :

```
<span style="color: red">This text will appear in red.</span>
```

> Remarque: essayez d'éviter cela - le but du CSS est de séparer le contenu de la présentation.

Feuilles de style multiples

Il est possible de charger plusieurs feuilles de style:

```
<head>
    <link rel="stylesheet" type="text/css" href="general.css">
    <link rel="stylesheet" type="text/css" href="specific.css">
</head>
```

Notez que les **fichiers et déclarations ultérieurs remplaceront les fichiers précédents** . Donc si `general.css` contient:

```
body {
    background-color: red;
}
```

et `specific.css` contient:

```
body {
    background-color: blue;
}
```

si les deux sont utilisés, l'arrière-plan du document sera bleu.

Lire Utiliser HTML avec CSS en ligne: https://riptutorial.com/fr/html/topic/4536/utiliser-html-avec-css

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec 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, Willl, Wolfgang, Zaz, zygimantus, Zze
2	Ancres et Hyperliens	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	Attributs d'événement HTML	Paresh Maghodiya
5	Attributs de données	animuson, Community, Faegy, Infuzed guy, James Donnelly, Manish, Nathan Tuggy, Nhan, Racil Hilan, rajarshig, swatchai, unor, Yasir T
6	Attributs globaux	animuson, Becca, unor, Zange-chan
7	Barres de navigation	Community
8	Cartes d'image	animuson, Lambda Ninja, RamenChef
9	Citations de marquage	Content Solutions, mnoronha, unor

10	Classes et identifiants	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
11	Commandes du menu de sélection	Ali Almoullim, amflare, animuson, GentlePurpleRain, Ilyas karim , Mosh Feu, Tot Zam
12	commentaires	Ani Menon, animuson, Ashwin Ramaswami, bdkopen, ChrisD, Epodax, JHS, jkdev, RamenChef, Robert Grant, Soaring Code, Squazz, Thomas Gerot, Ulrich Schwarz, Wolfgang
13	Des listes	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
14	Doctypes	Akshay Anand, Al.G., Angelos Chalaris, Ani Menon, animuson, Chris, Content Solutions, heerfk, mnoronha, pinjasaur, Right leg , Sumner Evans, Thomas Gerot, tmg, unor
15	Élément d'étiquette	Anthony Pham, fredden, Josiah Keller, m_callens, Roko C. Buljan
16	Élément de progrès	animuson, Content Solutions, Richard Hamilton
17	Élément de sortie	J F, Stephen Leppik, zer00ne
18	Élément div	animuson, Araknid, Black Mamba, Chris, Content Solutions, D M, Faust, feeela, Gal Ratzkin, JHS, MySpeed, S.L. Barth, SuperStormer, Thomas Gerot
19	Éléments de contrôle d'entrée	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, Edvin 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
20	Éléments de coupe	Andrew Brooke, Anil, animuson, Hanif Formoly, nalply, Shannon Young, SuperBiasedMan, SuperStormer, Yossi Aharon
21	Éléments multimédias	feeela, Isak Combrinck, LisaMM, Shiva, Yossi Aharon
22	Éléments vides	4444, ChrisD, Thomas Gerot, unor
23	Entités de caractère	animuson, MervS, stack-learner
24	Formatage du texte	animuson, Ben Rhys-Lewis, Emil, gustavohenke, J F, Matas Vaitkevicius, Peter L., Raystafarian, Stephen Leppik, Thomas Gerot, unor, Wolfgang
25	Formes	Ali Almoullim, Ani Menon, animuson, Aown Muhammad, Chris Rutherfurd, Cullub, Gabriel Chi Hong Lee, Greg T, j08691, Kimmax, Luca langella, Niek Brouwer, Thomas Gerot
26	HTML 5 Cache	Farhad, TricksfortheWeb, Valor Naram
27	IFrames	Adjit, Alexandre N., animuson, ChrisD, dorukayhan, Duh-Wayne-101, Emanuel Vintilă, J F, Ojen, Wojciech Kazior
28	Images	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
29	Inclure le code JavaScript dans HTML	Alexandre N., andreaem, animuson, Anselm Urban, Charles, Marjorie Pickard, MervS, Roko C. Buljan, Sildoreth, SuperStormer
30	Incorporer	Alexandre N.
31	Langues du contenu	animuson, FelipeAls, Gerold Broser, Isak Combrinck, Muntasir, Shannon Young, unor
32	les tables	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
33	Les titres	Ani Menon, animuson, dippas, Evan, Infuzed guy, joe_young, MervS, Nathan Arthur, Pseudonym Patel, sasha, Thomas Gerot , unor, V-Kopio

34	Marquage du code informatique	4444, Naveen Gogineni, Shannon Young, SuperStormer, Tot Zam, unor
35	Méta-information	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
36	Paragraphes	Abrar Jahin, Thomas Gerot, Valor Naram
37	Relier des ressources	AA2992, animuson, Anselm Urban, Aravind Suresh, Callan Heard, Chris Rutherfurd, cone56, DawnPaladin, Domenic, feeela, Henrique Barcelos, Infuzed guy, JHS, Lambda Ninja, Matas Vaitkevicius, Nhan, Thomas Gerot, unor, V4karian, vladdobra
38	SVG	andreas, Black Mamba, ChrisD, HerrSerker, Patrickdev, Timothy Miller, w5m
39	Tabindex	Content Solutions, Psaniko
40	Toile	cone56, Richard Hamilton, Roko C. Buljan, tonethar, Trevor Clarke, user4040648
41	Utiliser HTML avec CSS	animuson, bdkopen, Christian Ternus, Community, Euan Williams, feeela, Jones Joseph, Michael Moriarty, Thomas Gerot, thouusten