

Mémento des balises HTML

Cette page est une liste *non exhaustive* des balises HTML qui existent.

Vous trouverez ici un grand nombre de balises HTML. Nous en avons déjà vu certaines dans le cours, mais il y en a d'autres qu'on n'a pas eu l'occasion d'étudier. Généralement, les balises qu'on n'a pas étudiées sont des balises un peu plus rarement utilisées. Peut-être trouverez-vous votre bonheur dans ce lot de nouvelles balises. 😊

Vous pouvez vous servir de cette page comme d'un aide-mémoire lorsque vous développez votre site web. 😊



Attention j'insiste : cette page n'est pas complète et c'est volontaire. Je préfère mettre *moins* de balises et garder seulement celles qui me semblent les plus utiles dans la pratique.

Balises de premier niveau

Les balises de premier niveau sont les principales balises qui structurent une page HTML. Elles sont indispensables pour réaliser le "code minimal" d'une page web.

Balises	Description
<html>	Balise principale de toute page web. Doit englober tout le code de votre page web.
<head>	En-tête de la page
<body>	Corps de la page

Le code minimal d'une page HTML

Vous trouverez ci-dessous le code minimal d'une page HTML.

Code : HTML

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

  <body>

  </body>
</html>
```

Balises d'en-tête

Ces balises sont toutes situées dans l'en-tête de la page web, c'est-à-dire entre `<head>` et `</head>` :

Balise	Description
	<p>Cette balise permet d'indiquer certaines informations sur la page web. On l'utilise le plus souvent pour inclure une feuille de style CSS, comme ceci :</p> <p>Code : HTML</p> <pre><link rel="stylesheet" href="style.css" /></pre> <p>On peut aussi s'en servir pour 2-3 autres choses :</p>

<code><link /></code>	<p>Code : HTML</p> <pre><!-- Page d'accueil du site --> <link rel="start" title="Accueil" href="index.html" /> <!-- Page d'aide du site --> <link rel="help" title="Politique d'accessibilité" href="accessibilite.html" /> <!-- Fil RSS du site --> <link rel="alternate" type="application/rss+xml" title="News de mon site" href="news.xml" /> <!-- Icône du site (favicon) --> <link rel="shortcut icon" type="image/x-icon" href="favicon.ico" /></pre> <p>La favicon est une icône qui s'affiche généralement à gauche de l'adresse de votre site sur le navigateur de vos visiteurs. C'est un moyen de personnaliser un peu plus son site. Quant au fil RSS, il s'agit d'une technique permettant à vos visiteurs de suivre l'actualité de votre site depuis un logiciel spécial (un navigateur tel que Firefox le fait d'ailleurs). En général on génère des fils RSS en PHP (si vous ne faites que du HTML/ CSS ça ne vous intéresse donc pas pour le moment).</p>
	<p>Cette balise permet de définir les propriétés de la page web. On s'en sert pour une foule de choses. Voici quelques exemples pratiques :</p>

<div data-bbox="201 443 264 488" data-label="Text"> <code><meta /></code> </div>	<div data-bbox="371 185 504 208" data-label="Section-Header"> Code : HTML </div> <div data-bbox="395 257 1246 701" data-label="Text"> <pre> <!-- Table de caractères --> <meta charset="utf-8" /> <!-- Auteur de la page --> <meta name="author" content="Jean Dupont" /> <!-- Description de la page --> <meta name="description" content="La page personnelle de Jean Dupont" /> <!-- Mots-clés de la page --> <meta name="keywords" content="expériences, recherche, laboratoire, chimie" /> <!-- Adresse de contact --> <meta name="reply-to" content="monadresse@email.com" /> <!-- Empêcher la mise en cache de la page par le navigateur --> <meta http-equiv="pragma" content="no-cache" /> <!-- Rafraîchissement automatique au bout de 10 secondes --> <meta http-equiv="refresh" content="10; URL=http://www.monsite.com" /> </pre> </div> <div data-bbox="300 799 900 824" data-label="Text"> <p>En général, on utilise surtout le <code>meta</code> pour la table de caractères.</p> </div>
<div data-bbox="201 969 280 992" data-label="Text"> <code><script></code> </div>	<div data-bbox="300 846 791 896" data-label="Text"> <p>Permet de placer un script. On l'utilise souvent pour mettre du code Javascript :</p> </div> <div data-bbox="371 925 504 947" data-label="Section-Header"> Code : HTML </div> <div data-bbox="395 999 711 1070" data-label="Text"> <pre> <script> /* Votre script ici */ </script> </pre> </div>
	<div data-bbox="300 1137 711 1162" data-label="Text"> <p>Permet de définir du code CSS pour la page.</p> </div>

<style>	<p>Exemple :</p> <p>Code : HTML</p> <pre><style> /* Votre code CSS ici */ </style></pre>
<title>	<p>Titre de la page web.</p> <p>C'est probablement la balise la plus importante d'une page web. Choisissez bien votre titre car il a beaucoup d'importance pour les moteurs de recherche (ils donnent de l'importance aux mots qui se trouvent dans le titre).</p> <p>Code : HTML</p> <pre><title>Les petites expériences chimiques de M. Dupont</title></pre>

Balises de structuration du texte

Balise	Type	Description
<abbr>	<i>Inline</i>	Abréviation. Utiliser l'attribut <code>title</code> pour indiquer sa signification.
<blockquote>	<i>Block</i>	<p>Citation (longue) Vous devez obligatoirement mettre une balise de paragraphe à l'intérieur du blockquote. Par exemple :</p> <p>Code : HTML</p> <pre><blockquote> <p> Texte de la citation </p> </blockquote></pre>

<cite>	<i>Inline</i>	Citation du titre d'une oeuvre ou d'un évènement.
<q>	<i>Inline</i>	Citation (courte)
<sup>	<i>Inline</i>	Mise en exposant
<sub>	<i>Inline</i>	Mise en indice
	<i>Inline</i>	Mise en valeur (forte) Le texte est généralement mis en gras.
<mark>	<i>Inline</i>	Mise en valeur visuelle. Le texte est généralement surligné.
	<i>Inline</i>	Mise en valeur (faible) Le texte est généralement mis en italique.
<h6>	<i>Block</i>	Titre de niveau 6
<h5>	<i>Block</i>	Titre de niveau 5
<h4>	<i>Block</i>	Titre de niveau 4
<h3>	<i>Block</i>	Titre de niveau 3
<h2>	<i>Block</i>	Titre de niveau 2
<h1>	<i>Block</i>	Titre de niveau 1

<code></code>	<i>Inline</i>	<p>Insère une image. Utilisez les attributs <i>src</i> (pour indiquer l'adresse de l'image) et <i>alt</i> (pour indiquer un texte de remplacement). Ces 2 attributs sont obligatoires. Exemple :</p> <p>Code : HTML</p> <pre></pre>
<code><figure></code>	<i>Block</i>	Indique la présence d'une figure (image, code...) illustrant le texte.
<code><figcaption></code>	<i>Inline</i>	Description de la figure.
<code><audio></code>	<i>Inline</i>	Insère un son.
<code><video></code>	<i>Inline</i>	Insère une vidéo.
<code><source></code>	-	Indique un format possible pour les balises <code><audio></code> et <code><video></code> .
<code><a></code>	<i>Inline</i>	<p>Lien hypertexte. Indiquez l'url de destination grâce à l'attribut <i>href</i> :</p> <p>Code : HTML</p> <pre>Rendez-vous sur l'autre page</pre>
<code>
</code>	<i>Inline</i>	Retour à la ligne
<code><p></code>	<i>Block</i>	Paragraphe
<code><hr /></code>	<i>Block</i>	Crée une ligne de séparation horizontale
<code><address></code>	<i>Block</i>	Permet d'indiquer une adresse, ou éventuellement l'auteur d'un document. Le texte est généralement mis en italique.

<code></code>	<i>Inline</i>	Permet d'indiquer un texte qui a été supprimé. Le texte est généralement barré.
<code><ins></code>	<i>Inline</i>	Permet d'indiquer un texte qui a été inséré. Le texte est généralement souligné.
<code><dfn></code>	<i>Inline</i>	Permet d'indiquer une définition.
<code><kbd></code>	<i>Inline</i>	Permet d'indiquer un code que doit taper le visiteur.
<code><pre></code>	<i>Block</i>	Le texte à l'intérieur de la balise <code><pre></code> sera affiché tel qu'il a été tapé dans le code (espaces et entrées compris). Une police de taille fixe est utilisée.
<code><progress></code>	<i>Inline</i>	Affiche une barre de progression. A utiliser conjointement avec les attributs <i>value</i> et <i>max</i> .
<code><time></code>	<i>Inline</i>	Pour indiquer une date ou une heure.

Balises de liste

Cette partie énumère toutes les balises HTML permettant de créer des listes (listes à puces, listes numérotées, listes de définitions...)

Balise	Type	Description
<code></code>	<i>Block</i>	<p>Liste à puces non numérotée. Vous devez mettre un <code></code> par élément de la liste. Exemple :</p> <p>Code : HTML</p> <pre> Un élément</pre>

		<pre>Un autre élément </pre>
	Block	<p>Liste à puces numérotée. Vous devez mettre un par élément de la liste. Exemple :</p> <p>Code : HTML</p> <pre> Elément n°1 Elément n°2 </pre>
	list-item	<p>Permet de créer un élément de liste. Le type de la balise est particulier car elle n'est ni block ni inline. On dit qu'elle est de type <i>list-item</i>.</p>
<dl>	Block	<p>Liste de définitions. Vous devez alterner chaque terme <dt> par sa définition <dd>. Exemple :</p> <p>Code : HTML</p> <pre><dl> <dt>Porte</dt> <dd>Ouverture dans un mur permettant d'entrer et de sortir</dd> <dt>Théâtre</dt> <dd>Lieu où l'on représente des ouvrages dramatiques</dd> </dl></pre>
<dt>	Block	Terme à définir
<dd>	Block	Définition du terme

Balises de tableau

Balise	Type	Description
<table>	Block	<p>Délimite un tableau. Voici un exemple de tableau simple :</p> <p>Code : HTML</p> <pre><table> <caption>Passagers du vol 377</caption> <tr> <th>Nom</th> <th>Age</th> <th>Pays</th> </tr> <tr> <td>Carmen</td> <td>33 ans</td> <td>Espagne</td> </tr> <tr> <td>Michelle</td> <td>26 ans</td> <td>Etats-Unis</td> </tr> <tr> <td>François</td></pre>

		<pre> <td><td>43 ans</td> <td><td>France</td> </tr> </table> </td></td></pre>	<td>43 ans</td> <td><td>France</td> </tr> </table> </td>	<td>France</td> </tr> </table>
<caption>	-	Permet de donner un titre au tableau		
<tr>	-	Ligne de tableau		
<th>	-	Cellule d'en-tête du tableau (généralement mise en gras)		
<td>	-	Cellule du tableau		
<thead>	-	<p>Balise non obligatoire permettant d'insérer l'en-tête du tableau. Si vous choisissez d'utiliser <thead>, <tfoot> et <tbody>, vous devez les mettre dans l'ordre suivant dans votre code source :</p> <ol style="list-style-type: none"> 1. <thead> 2. <tfoot> 3. <tbody> 		
<tbody>	-	Balise non obligatoire permettant d'insérer le corps du tableau		
<tfoot>	-	Balise non obligatoire permettant d'insérer le pied du tableau		

Balises de formulaire

Balise	Type	Description
<form>	Block	<p>Délimite un formulaire. Vous devrez généralement donner 2 attributs à la balise <form></p> <ul style="list-style-type: none"> • method : indique la méthode d'envoi du formulaire (get ou post). Si vous ne savez pas quoi utiliser, mettez post. • action : la page vers laquelle le visiteur doit être redirigé lorsqu'il a validé votre formulaire.
<fieldset>	Block	<p>Permet de regrouper plusieurs éléments d'un formulaire. On l'utilise généralement dans de grands formulaires.</p> <p>Pour donner un titre à votre groupe, utilisez la balise <legend></p>
<legend>	Inline	<p>Titre d'un groupe dans un formulaire. A utiliser à l'intérieur d'un <fieldset></p>
<label>	Inline	<p>Titre d'un élément de formulaire. Généralement, vous devrez mettre l'attribut <i>for</i> sur cette balise pour indiquer l'ID de l'élément auquel correspond le label.</p>
<input />	Inline	<p>Champ de formulaire. Il existe de nombreux types de champs différents. Vous choisissez le type de champ que vous désirez grâce à l'attribut <i>type</i> :</p> <p>Code : HTML</p> <pre> <!-- Zone de texte d'une ligne --> <input type="text" /> <!-- Mot de passe (le texte est caché) --> <input type="password" /> <!-- Envoi de fichier --> <input type="file" /> <!-- Case à cocher --> <input type="checkbox" /> <!-- Bouton d'option --> </pre>

		<pre> <input type="radio" /> <!-- Bouton --> <input type="button" /> </pre>
--	--	---

		<pre> <!-- Bouton d'envoi --> <input type="submit" /> <!-- Bouton de remise à zéro --> <input type="reset" /> <!-- Champ caché --> <input type="hidden" /> </pre>
		Pensez à donner un nom à vos champs grâce à l'attribut <i>name</i>
<textarea>	Inline	Zone de saisie multiligne. Vous pouvez définir sa taille grâce aux attributs <i>rows</i> et <i>cols</i> (nombre de lignes et colonnes) ou bien le faire en CSS grâce aux propriétés <i>width</i> et <i>height</i> .
<select>	Inline	Liste déroulante. Utilisez la balise <option> pour créer chaque élément de la liste : Code : HTML <pre> <select name="pays"> <option value="france">France</option> <option value="espagne">Espagne</option> <option value="italie">Italie</option> </select> </pre>
<option>	Block	Element d'une liste déroulante
<optgroup>	Block	Groupe d'éléments d'une liste déroulante. A utiliser dans le cas d'une grande liste déroulante. Vous devez utiliser l'attribut <i>label</i> pour donner un nom au groupe.

Ces balises ont un intérêt uniquement si vous leur donnez un attribut *class*, *id* ou *style* :

- **class** : indique le nom de la classe CSS à utiliser.
- **id** : donne un nom à la balise. Ce nom doit être unique sur toute la page car il permet d'identifier la balise. Vous pouvez vous servir de l'ID pour de nombreuses choses, comme par exemple pour un lien vers une ancre, pour un style CSS de type ID, pour des manipulations en Javascript etc.
- **style** : cet attribut vous permet d'indiquer directement le code CSS à appliquer. Vous n'êtes donc pas obligés d'avoir une feuille de style à part, vous pouvez juste mettre directement les attributs CSS. Notez qu'il est préférable de ne pas utiliser cet attribut et de passer à la place par une feuille de style externe car cela rend votre site plus facile à mettre à jour par la suite.

Ces 3 attributs ne sont pas réservés aux balises génériques : vous pouvez aussi les mettre sur la plupart des autres balises sans aucun problème. 😊

Comme je vous l'ai dit au début de ce chapitre, il y a plusieurs balises que j'ai volontairement omises.

Vous l'aurez constaté, en HTML tout est affaire de *sens* (on parle de *sémantique*). Ce qui compte, c'est d'utiliser la balise qui convient le mieux à chaque moment.

En théorie, on pourrait faire presque tout un site rien qu'avec les balises génériques <div> et (en utilisant du CSS), mais votre site n'aurait aucun sens logique ! Or, respecter la logique de son code source est une chose que les webmasters considèrent comme fondamentale. Une page sémantique a notamment plus de chances d'être mieux indexée dans Google qu'une page utilisant des balises inadaptées.

Souvenez-vous en ! 😊

Mémento des propriétés CSS

Cette page est une liste non exhaustive des propriétés CSS qui existent en CSS3.

Pour la plupart, ce sont des propriétés que nous avons vues dans le cours, mais vous trouverez aussi quelques nouvelles propriétés que nous n'avons pas abordées.

La liste est non exhaustive car mon but n'est pas de faire la liste de toutes les propriétés CSS qui peuvent exister : il y en a vraiment trop (plus de 200 !) et certaines sont très rarement utilisées.

Propriétés de formatage de texte

Je résume ici la plupart des propriétés de **formatage de texte**.

Qu'est-ce que le formatage de texte ? C'est tout ce qui consiste à mettre en forme le texte : mettre en gras, italique, souligné, changer la police, l'alignement etc...

Police, taille et décorations

Type	Nom	Valeurs possibles
Nom de police	font-family	<p>Indiquer les noms de polices possibles par ordre de préférence :</p> <p>Code : CSS</p> <pre>font-family: police1, police2, police3;</pre> <p>Si le visiteur a la police 1, il l'utilisera. Sinon, il regarde s'il a la police 2, puis la police 3 etc. Utilisez des guillemets si le nom de la police comporte des espaces. Essayez de toujours mettre comme dernière police possible "serif" ou "sans-serif".</p>
		<p>Code : CSS</p> <pre>font-family: "Arial Black", Arial, Verdana, serif;</pre> <p>Il est possible d'utiliser des polices personnalisées en combinaison avec @font-face.</p>
Police personnalisée	@font-face	<p>Permet de déclarer une nouvelle police, qui sera téléchargée sur l'ordinateur de vos visiteurs.</p> <p>Code : CSS</p> <pre>@font-face { font-family: 'MaSuperPolice'; src: url('MaSuperPolice.eot') format('eot'), url('MaSuperPolice.woff') format('woff'), url('MaSuperPolice.ttf') format('truetype'), url('MaSuperPolice.svg') format('svg'); }</pre>
		<p>Indiquez la taille du texte.</p> <p>Plusieurs unités sont possibles :</p> <ul style="list-style-type: none">• px (pixels)

Taille du texte	font-size	<ul style="list-style-type: none"> • % (pourcentage, 100% = normal) • em (taille relative, 1.0 = normal) • ex (taille relative à la hauteur de la lettre "x", 1.0 = normal) • nom de taille : <ul style="list-style-type: none"> ◦ xx-small : très très petit ◦ x-small : très petit ◦ small : petit ◦ medium : moyen ◦ large : grand ◦ x-large : très grand ◦ xx-large : très très grand
Gras	font-weight	bold : gras bolder : plus gras lighter : plus fin normal : pas gras (par défaut)
Italique	font-style	italic : italique oblique : autre façon de mettre en italique normal : normal (par défaut)
Décoration	text-decoration	underline : souligné overline : ligne au-dessus line-through : barré blink : clignotant none : normal (par défaut)
Petites capitales	font-variant	small-caps : petites capitales normal : normal (par défaut)
Capitales	text-transform	uppercase : tout mettre en majuscules lowercase : tout mettre en minuscules capitalize : début des mots en majuscules none : normal (par défaut)

Super-propriété de police	font	<p>Indiquez dans n'importe quel ordre des valeurs possibles pour <i>font-weight</i>, <i>font-style</i>, <i>font-size</i>, <i>font-variant</i>, <i>font-family</i>.</p> <p>Attention exception : le nom de la police (<i>font-family</i>) doit être placé en dernier dans la liste dans tous les cas.</p> <p>Vous n'êtes pas obligés de mettre une valeur de chacune de ces propriétés.</p> <p>Exemple :</p> <p>Code : CSS</p> <pre>font: bold 16px Arial;</pre> <p>Cela mettra votre texte en gras, 16 pixels, Arial.</p>
---------------------------	------	---

Alignement

Type	Propriété	Valeurs possibles
Alignement horizontal	text-align	left : à gauche (par défaut) center : centré right : à droite justify : texte justifié (prend toute la largeur de la page)
Alignement vertical	vertical-align	A utiliser dans des cellules de tableau, ou dans des éléments inline-block.

Alignement vertical	vertical-align	top : en haut middle : au milieu bottom : en bas
Hauteur de ligne	line-height	Indiquer une valeur en pixels (px) ou en pourcentage (%)
Alinéa	text-indent	Indiquez une valeur en pixels (px) pour définir l'alinéa de vos paragraphes. Vos paragraphes commenceront avec le retrait que vous avez indiqué.
Césure	white-space	normal : le passage à la ligne est automatique (par défaut) nowrap : pas de passage à la ligne automatique, à moins qu'une balise HTML comme <code>
</code> ne soit présente. pre : le passage à la ligne se fait tel que le texte a été saisi dans le code source (comme la balise <code><pre></code>)
Césure forcée	word-wrap	Avec la valeur <code>break-word</code> , le texte sera coupé s'il dépasse du cadre.
Ombre de texte	text-shadow	Décalage vertical, décalage horizontal, adoucissement, couleur Code : CSS <pre>text-shadow: 2px 2px 4px black;</pre>

Propriétés de couleur et de fond

Couleur

Type	Propriété	Valeurs possibles
Couleur de texte	color	Indiquer une couleur avec l'une des méthodes suivantes : <ul style="list-style-type: none"> En tapant le nom de la couleur en anglais (black, blue, green, white, red...). En indiquant la couleur en hexadécimal (#CC48A1) En indiquant la couleur en RGB : rgb (128, 255, 0)
Couleur de fond	background-color	Même fonctionnement que <i>color</i> . Cela définit cette fois la couleur de fond du texte

Image de fond

Type	Propriété	Valeurs possibles
Image de fond	background-image	Indiquer l'url de l'image (notation absolue ou relative) Code : CSS <pre>background-image:url("images/fond.png"); /* Notation relative */ background- image:url("http://www.monsite.com/images/fond.png"); /* Notation absolue */</pre> Il est possible de combiner plusieurs images de fond, en séparant les déclarations par des virgules.
Fond fixé	background-attachment	fixed : le fond reste fixe quand on descend plus bas sur la page scroll : le fond défile avec le texte (par défaut)
Répétition du	background-	repeat : le fond se répète (par défaut) repeat-x : le fond ne se répète que sur une ligne, horizontalement

fond	repeat	repeat-y : le fond ne se répète que sur une colonne, verticalement no-repeat : le fond ne se répète pas, il n'est affiché qu'une fois
Position du fond	background-position	2 façons de faire : <ul style="list-style-type: none"> En notant une distance en px ou %, par rapport au coin en haut à gauche. Code : CSS <pre>background-position:50px 200px; /* 50 px à droite, 200px en bas */</pre> En utilisant des valeurs prédéfinies, une pour la verticale et une pour l'horizontale : top : en haut, verticalement center : au milieu, verticalement bottom : en bas, verticalement left : à gauche, horizontalement center : au centre, horizontalement right : à droite, horizontalement Code : CSS <pre>background-position : bottom right; /* en bas à droite */</pre>
		Indiquer une ou plusieurs valeurs issues des propriétés <i>background-image</i> , <i>background-repeat</i> , <i>background-attachment</i> , <i>background-position</i> . L'ordre des valeurs n'a pas d'importance et vous n'êtes pas obligés de mettre toutes les valeurs de ces propriétés (au moins une suffit)

Super-propriété de fond	background	Code : CSS <pre>/* Le fond fond.png reste affiché en haut à droite de l'écran et n'est pas répété. */ background:url("images/fond.png") no-repeat fixed top right;</pre>
Transparence	opacity	Valeur entre 0 (transparence totale) et 1 (opacité totale). Exemple pour une transparence de 50% : Code : CSS <pre>opacity:0.5;</pre>

Propriétés des boîtes

Dimensions

Type	Propriété	Valeurs possibles
Largeur	width	Valeur en px, %, ou encore "auto" (valeur par défaut, la largeur dépendra du texte à l'intérieur)
Hauteur	height	Idem
Largeur minimale	min-width	Indiquer une valeur, en pixels par exemple.
Largeur maximale	max-width	Idem

Hauteur minimale	min-height	Idem
Hauteur maximale	max-height	Idem

Marges extérieures

Type	Propriété	Valeurs possibles
Marge en haut	margin-top	Indiquer une valeur comme 20px, 1.5em...
Marge à gauche	margin-left	Idem
Marge à droite	margin-right	Idem
Marge en bas	margin-bottom	Idem
Super-propriété de marge	margin	<p>Indiquez de 1 à 4 valeurs à la suite. Selon le nombre de valeurs que vous mettez, la signification change :</p> <ul style="list-style-type: none"> • 1 valeur : ce sera la marge pour le haut, le bas, la gauche et la droite • 2 valeurs : la première correspond à la marge pour le haut et le bas, la seconde pour la gauche et la droite • 3 valeurs : la première correspond à la marge du haut, la seconde aux marges à gauche et à droite, la troisième à la marge du bas • 4 valeurs : respectivement la marge du haut, de la droite, du bas, de la gauche. <p>Par exemple, si je mets 2 valeurs :</p>

		<p>Code : CSS</p> <pre>margin:20px 5px; /* 20px de marge en haut et en bas, 5px à gauche et à droite */</pre>
--	--	--

Marges intérieures

Type	Propriété	Valeurs possibles
Marge intérieure en haut	padding-top	Indiquer une valeur comme 20px, 1.5em...
Marge intérieure à gauche	padding-left	Idem
Marge intérieure à droite	padding-right	Idem
Marge intérieure en bas	padding-bottom	Idem
Super-propriété de	padding	<p>Indiquez de 1 à 4 valeurs à la suite. Selon le nombre de valeurs que vous mettez, la signification change :</p> <ul style="list-style-type: none"> • 1 valeur : ce sera la marge pour le haut, le bas, la gauche et la droite • 2 valeurs : la première correspond à la marge pour le haut et le bas, la seconde

marge intérieure	padding	<p>pour la gauche et la droite</p> <ul style="list-style-type: none"> 3 valeurs : la première correspond à la marge du haut, la seconde aux marges à gauche et à droite, la troisième à la marge du bas 4 valeurs : respectivement la marge du haut, de la droite, du bas, de la gauche.
------------------	---------	--


Bordures

Type	Propriété	Valeurs possibles
Epaisseur de la bordure	border-width	Indiquer une valeur en px.
Couleur de la bordure	border-color	Indiquer une valeur de couleur.
Type de bordure	border-style	<p>none : pas de bordure (par défaut)</p> <p>hidden : bordure cachée</p> <p>solid : ligne pleine</p> <p>double : ligne double (nécessite une taille de bordure de 3px minimum)</p> <p>dashed : en tirets</p> <p>dotted : en pointillés</p> <p>inset : effet 3D "enfoncé"</p> <p>outset : effet 3D "surélevé"</p> <p>ridge : autre effet 3D</p>
		Indiquer la couleur, l'épaisseur et le type de bordure pour la bordure gauche. L'ordre n'a pas d'importance. Exemple :

Bordure à gauche	border-left	<p>Code : CSS</p> <pre>border-left: 2px inset blue; /* Bordure bleue de 2px avec effet 3D "enfoncé" */</pre>
Bordure en haut	border-top	Idem
Bordure à droite	border-right	Idem
Bordure en bas	border-bottom	Idem
Super-propriété de bordure	border	Indiquera l'apparence des bordures en haut, à droite, en bas et à gauche.
Bordure arrondie	border-radius	Indiquer une valeur en px, ou 4 valeurs pour chacun des coins, en partant de celui en haut à gauche.
Ombre	box-shadow	<p>Décalage vertical, décalage horizontal, adoucissement, couleur</p> <p>Code : CSS</p> <pre>box-shadow: 2px 2px 4px black;</pre>

Propriétés de positionnement et d'affichage

Affichage

Type	Propriété	Valeurs possibles
Type d'élément	display	none : l'élément ne sera pas affiché block : l'élément devient de type "block" (bloc, comme <p>) inline : l'élément devient de type "inline" (en ligne, comme) inline-block : l'élément est affiché comme un inline mais peut être redimensionné comme un block. list-item : l'élément devient de type "élément de liste à puce" (comme)
Affichage	visibility	hidden : masqué visible : visible (par défaut)  display:none ; fait complètement disparaître l'élément, tandis que visibility:hidden ; masque l'élément, qui continue quand même à prendre de la place sur l'écran.
Afficher seulement une partie	clip	Indiquer 4 valeurs comme ceci : Code : CSS <pre>clip: rect(valeur1, valeur2, valeur3, valeur4);</pre> Cela permet de n'afficher qu'une partie d'un élément. rect() permet d'indiquer les coordonnées du rectangle qui sera affiché. Les valeurs 1 à 4 correspondent respectivement aux coins haut, droite, bas et gauche du rectangle.
Limiter les dimensions	overflow	visible : tout l'élément sera affiché (par défaut). hidden : l'élément sera coupé s'il dépasse les limites définies par height et width. On ne pourra pas voir la partie du texte coupée. scroll : tout comme hidden, l'élément sera coupé s'il dépasse les limites. Toutefois, cette fois le navigateur ajoutera des barres de défilement pour qu'on puisse voir la suite du texte. auto : c'est le navigateur qui décide d'ajouter des barres de défilement ou pas en fonction des cas. Bien souvent, utiliser cette valeur revient à utiliser la valeur "scroll".

Positionnement

Type	Propriété	Valeurs possibles
Flottant	float	left : flottant à gauche right : flottant à droite none : pas de flottant (par défaut)
Stopper un flottant	clear	left : supprime l'effet d'un flottant à gauche précédent right : supprime l'effet d'un flottant à droite précédent both : supprime l'effet d'un flottant précédent, qu'il soit à gauche ou à droite none : pas de suppression de l'effet du flottant (par défaut)
Type de positionnement	position	absolute : position absolue par rapport au coin en haut à gauche fixed : position fixe (fonctionne comme la position absolue). L'élément reste à sa position même quand on descend plus bas dans la page. relative : position relative, par rapport à la position "normale" de l'élément static : positionnement normal (par défaut)
Position par rapport au haut	top	Valeur en px, %, em... A utiliser pour un positionnement absolu, fixe ou relatif.
Position par rapport au bas	bottom	Valeur en px, %, em... A utiliser pour un positionnement absolu, fixe ou relatif.

Position par rapport au bas	bottom	Idem
Position par rapport à gauche	left	Idem
Position par rapport à droite	right	Idem
Ordre d'affichage	z-index	<p>En cas de positionnement absolu par exemple, si 2 éléments se chevauchent, z-index permet d'indiquer quel élément doit être affiché au-dessus de l'autre.</p> <p>Indiquez un nombre. Plus ce nombre est élevé, plus l'élément sera affiché en avant.</p> <p>Par exemple, si vous avez 2 éléments positionnés en absolus avec un z-index de 10 pour l'un et de 20 pour l'autre, c'est celui qui a un z-index de 20 qui sera affiché par-dessus.</p>

Propriétés des listes

Type	Propriété	Valeurs possibles
Type de liste	list-style-type	<ul style="list-style-type: none"> Pour les listes non ordonnées () : <ul style="list-style-type: none"> disc : un disque noir. circle : un cercle (par défaut). square : un carré. none : aucune puce ne sera utilisée. Pour les listes ordonnées () : <ul style="list-style-type: none"> decimal : des nombres 1, 2, 3, 4, 5... (par défaut) decimal-leading-zero : des nombres commençant par zéro (01, 02, 03, 04, 05...). upper-roman : numérotation romaine, en majuscules (I, II, III, IV, V...) lower-roman : numérotation romaine, en minuscules (i, ii, iii, iv, v...) upper-alpha : numérotation alphabétique, en majuscules (A, B, C, D, E...) lower-alpha : numérotation alphabétique, en minuscules (a, b, c, d, e...) lower-greek : numérotation grecque.

Position en retrait	list-style-position	inside : sans retrait outside : avec retrait (par défaut)
Puce personnalisée	list-style-image	<p>Indiquer l'url de l'image qui servira de puce. Exemple :</p> <p>Code : CSS</p> <pre>list-style-image: url("images/puce.png");</pre>
Super-propriété de liste	list-style	<p>Vous pouvez réunir les valeurs de list-style-type, list-style-position et list-style-image. Vous n'êtes pas obligés de mettre toutes les valeurs, et l'ordre n'a pas d'importance.</p> <p>Exemple :</p> <p>Code : CSS</p> <pre>list-style: inside square;</pre>

Propriétés des tableaux

Type	Propriété	Valeurs possibles
Type de bordure	border-collapse	collapse : les bordures du tableau et des cellules sont mélangées. separate : les bordures du tableau et des cellules sont séparées (par défaut).
Cellules vides	border-collapse	show : les bordures des cellules vides sont affichées.

Cellules vides	empty-cells	collapse : les cellules vides sont masquées (par défaut).
Position du titre	caption-side	Indique la position du titre du tableau, défini via la balise <caption> top : en haut du tableau bottom : en bas du tableau left : à gauche du tableau right : à droite du tableau

Autres propriétés

Type	Propriété	Valeurs possibles
Curseur de souris	cursor	auto : curseur automatique (par défaut) default : curseur standard pointer : curseur en forme de main, comme quand on pointe sur un lien text : curseur utilisé quand on pointe sur du texte wait : curseur utilisé pour indiquer une attente (sablier) progress : curseur utilisé pour indiquer une tâche de fond (curseur avec sablier) help : curseur en forme de point d'interrogation, indiquant une aide move : curseur en forme de croix, indiquant un déplacement possible n-resize : flèche vers le nord ne-resize : flèche vers le nord-est e-resize : flèche vers l'est se-resize : flèche vers le sud-est s-resize : flèche vers le sud sw-resize : flèche vers le sud-ouest w-resize : flèche vers l'ouest nw-resize : flèche vers le nord-ouest url : curseur personnalisé, de type .cur ou .ani. Exemple :

		<p>Code : CSS</p> <pre>cursor: url("images/curseur.cur"), auto;</pre> <p>Vous devez utiliser un logiciel dédié à la création de curseurs pour créer des .cur et des .ani. Notez aussi la présence du mot auto à la fin. Cela signifie que le navigateur tentera de charger le curseur personnalisé. S'il n'a pu être chargé pour une quelconque raison, le curseur correspondant à la valeur auto sera utilisé.</p>
--	--	--

Pfiou !

Recenser les propriétés CSS n'est ni reposant, ni amusant je vous l'assure !

Les plus experts d'entre vous auront remarqué que je n'ai pas mis toutes les propriétés CSS, comme je l'ai dit au début. En effet, mon but n'était pas de faire la liste complète mais plutôt de vous fournir un support lorsque vous codez en CSS.

J'ai donc conservé les propriétés CSS qui me paraissaient les plus souvent utilisées.

Le cours s'arrête ici ! Pensez à consulter les annexes si vous ne l'avez pas fait, vous y découvrirez de nouvelles pistes pour poursuivre votre apprentissage de HTML et CSS ! 😊

Les sélecteurs CSS avancés indispensables

Comme incontournables...

En voici leur description accompagné à chaque fois d'un exemple d'utilisation.

Ces sélecteurs sont bien utiles, notamment pour modifier le look de CMS sur lesquels vous ne voulez (ou ne pouvez) pas modifier le code source.

Ils peuvent tous être combinés les uns avec les autres, amusez-vous à en tester les cocktails !

Table des matières

- * L'étoile
- # Le dièse (l'ID)
- . Le point (la classe)
- .X.Y Classes accolées
- .X.Y Deux classes
- x Le type
- :hover Au survol
- x:not Pseudo-classe de négation
- X Y Sélecteur descendant
- X > Y Sélecteur d'enfant
- X + Y Sélecteur adjacent direct
- X ~ Y Sélecteur adjacent indirect
- X:link Liens non visités
- X:visited Liens visités
- X[foo] Attribut
- X[Y=foo] Attribut Y précis
- X[Y*=foo] Attribut Y dans la valeur
- X[Y^=foo] Attribut Y commence par
- X[Y\$=foo] Attribut Y finit par
- data-xyz Propre attribut
- ~ Titled une des valeurs des data-xyz
- X:checked
- X:before et X:after
- X::first-line
- X::first-letter
- X:nth-child(n)
- X:nth-last-child(n)
- X:nth-of-type(n)
- X:first-child
- X:last-child
- X:only-child
- X:only-of-type
- X:first-of-type

* (l'étoile)

Le symbole étoile est le sélecteur universel, il cible tous les éléments de la page.
Pratique pour mettre à zéro les marges de tous les éléments par exemple.

```
* {  
  margin: 0;  
}
```

#X (le dièse, l'ID)

Le sélecteur # permet de cibler un élément par son identifiant. Condition : l'identifiant doit être unique, en d'autres termes, plusieurs éléments ne peuvent avoir le même identifiant sur une même page.

```
#menu {  
  width: 960px;  
  margin: 10px auto;  
}
```

.X (le point, la classe)

Le point est le sélecteur de classe, il peut cibler plusieurs éléments (à la différence du sélecteur d'ID).

```
.bleu {  
  color: #337ab7;  
}
```

.X.Y (deux classes accolées)

Ces deux sélecteurs de classe **collés l'une à l'autre**, ciblent uniquement des éléments qui ont ces deux classes (.X.Y)

```
.slogan {  
  font-style: italic;  
}  
.slogan.bleu {  
  color: #337ab7;  
}
```

Ici la couleur bleu ne sera appliquée qu'aux éléments ayant la classe "slogan" mais aussi la classe "bleu" (au moins)

Exemple : `<p class="slogan bleu">Je suis un slogan en italique et bleu</p>`

.X.Y (deux classes qui se suivent)

Ces deux sélecteurs de classe **avec un espace entre l'un et l'autre**, ciblent uniquement l'élément descendant Y du parent X

```
.article .rouge {  
  color: #f00;  
}
```

Ici la couleur rouge pourra s'appliquer à l'élément ".rouge" descendant (direct ou indirect) de l'élément ".article"

```
<div class="article">  
  <p class="rouge">Je suis un paragraphe rouge dans un conteneur "article"</p>  
</div>
```

Notez qu'ici on peut aussi styliser individuellement le div en utilisant la classe ".article" et le paragraphe avec la classe ".rouge".

X (le type)

Le sélecteur de type permet de cibler un élément grâce à son type.

```
body {  
  color: #333;  
}  
h1 {  
  font-size: 36px;  
  font-weight: 500;  
  line-height: 1.1;  
}
```

:hover (deux-points hover)

Le sélecteur **:hover** permet d'appliquer un style lorsque l'utilisateur survole un élément.

```
.bouton:hover {  
  background-color: #4e74c2;  
  color: #fff;  
}
```

:not(sélecteur)

La pseudo-classe de négation permet de sélectionner tous les types mentionnés sauf ceux pointés par le sélecteur.

```
p:not(.secondaire) {  
  text-decoration: underline;  
}
```

Dans cet exemple les paragraphes auxquels ne sont pas attribués la classe "secondaire" seront soulignés.

X Y (sélecteur descendant)

le sélecteur descendant permet d'être plus spécifique.

```
li a {  
  text-decoration: none;  
}
```

Dans cet exemple seuls les liens compris dans une liste seront soulignés.

X > Y (sélecteur d'enfant)

le sélecteur d'enfant cible seulement le descendant direct

```
div#menu > ul {  
  border: 1px solid #333;  
}
```

Dans cet exemple seuls les éléments **ul** de premier niveau, descendants direct du **div#menu** auront une bordure.

X + Y (sélecteur adjacent direct)

Le sélecteur adjacent direct cible tout élément Y immédiatement précédé par un élément X.

```
p + h2 {  
  margin-top: 0;  
}
```

Dans cet exemple seuls les titre de niveau 2 immédiatement précédés par un paragraphe n'auront pas de marge du dessus.

X ~ Y (sélecteur adjacent indirect)

Un peu similaire au précédent mais plus souple, ce sélecteur permet de cibler tous les éléments similaires Y précédés directement ou indirectement par un élément X.

```
h2 ~ ul {  
  margin-top: 10px;  
}
```

Dans cet exemple ce sélecteur cible toutes les listes non ordonnées placées après un titre h2 (mais pas forcément juste après).

X:link

La pseudo-classe :link cible tous les liens qui n'ont pas été encore visités.

```
a:link {  
  text-decoration: none;  
}
```

X:visited

La pseudo-classe :visited correspond à tous les liens qui ont été visités.

```
a:visited {  
  text-decoration: underline;  
}
```

X[foo]

Cible tout élément X ayant un attribut foo.

```
span[title] {  
  border-bottom: 1px dotted #777;  
  cursor: help;  
}
```

Dans cet exemple tous les seront soulignés (en pointillés) et le curseur de la souris prendra la forme d'un point d'interrogation.

X[Y="foo"]

Cible tout élément X ayant un attribut Y précis.

```
a[target="_blank"] {  
  border-right: 1px solid blue;  
}
```

Dans cet exemple tous les liens ouvrant de nouvelles pages seront stylisés avec une bordure droite bleue.

X[Y*="foo"]

On ajoute une étoile pour indiquer que "foo" doit se trouver quelque part dans la valeur de l'attribut Y.

```
a[href*="gnu"] {  
  font-weight: bold;  
}
```

Dans cet exemple tous les liens vers gnu.org, forum.gnu.org, support.gnu.org etc. seront pris en compte et mis en gras.

X[Y^="foo"]

Comme pour les [Regex](#), avec le symbole ^ circonflexe, on cible tous éléments X dont la valeur de l'attribut Y commence par "foo".

```
a[href^="http"] {  
  text-decoration: underline;  
}
```

Dans cet exemple tous les liens commençant par http seront soulignés.

X[Y\$="foo"]

Idem encore que pour les Regex, ici avec le \$, on cible tous éléments X dont la valeur de l'attribut Y finit par "foo"

```
a[href$=".png"] {  
  margin-right: 5px;  
}
```

Dans cet exemple tous les liens finissant par .png auront une marge à droite de 5px.

X[data-xyz="foo"]

Avec data- il est possible de créer ses propres attributs et pouvoir ainsi cibler spécifiquement un élément X.

Fausse Bonne Idée en HTML5 ? [Cet article](#) semble bien le prouver, car cette utilisation de data- donne à penser que les données ciblées sont d'une importance immédiate pour l'utilisateur (alors que ce ne serait là que pour donner du style). Utilisez-le donc en toute connaissance de cause et surtout si vous n'avez aucun autre moyen (plus sémantique) de cibler votre élément X.

```
a[data-info="video"] {  
  border-color: 1px solid green;  
}
```

Un lien exprimé ainsi par exemple `<a data-info="video" href="#">Voir la vidéo` sera entouré par une bordure verte.

X[foo~="un-truc"]

Avec cette utilisation, le symbole tilde ~ permet de cibler un élément dont l'attribut "foo" a pour valeur une liste de valeurs séparées par des espaces et dont une de celles-ci serait "un-truc".

```
a[data-exemple~="cinema"] {  
  border: 1px solid red;  
}
```

Dans cet exemple ce lien `En savoir plus` aura une bordure rouge. Mais idem que pour le sélecteur précédent, cela peut s'avérer une Fausse Bonne Idée en HTML5.

X:checked

Cette pseudo-classe cible uniquement les éléments de formulaire qui ont été cochés, bouton radio ou case à cocher.

```
input[type=checkbox]:checked {  
  background-color: gray;  
}
```

Ici les cases cochées d'un formulaires auront un fond gris.

X:before et X:after

Les pseudo-éléments before et after servent à générer en CSS du contenu esthétique voir textuel avant ou après l'élément ciblé. Il sont souvent utilisés avec la propriété CSS "**content**".

```
a[target=_blank]:after {  
  content: url("externe.png");  
}
```

Ici en associant le sélecteur :after au sélecteur de l'attribut target=_blank, l'image "externe.png" sera ajoutée après tous les liens sortant.

p::first-line

Applique un style à la première ligne d'un paragraphe

```
p::first-line {  
  font-size: 1.3em;  
}
```

p::first-letter

Applique un style à la première lettre d'un paragraphe

```
p::first-letter {  
  float: left;  
  font-size: 2.2em;  
  padding-right: 5px;  
  font-weight: bold;  
}
```

Ce pseudo-élément permet d'obtenir un effet lettrine d'éditorial de journal pour la première lettre d'un article.

X:nth-child(n)

Assez compliqué à expliquer clairement, ce pseudo-élément permet de cibler par exemple toutes les lignes impaires d'un tableau HTML.

Mais il peut aussi servir à beaucoup d'autres choses, comme à styliser des listes par exemple.

```
tr:nth-child(2n+1) {  
  background-color: #F7F7F7;  
}
```

X:nth-last-child(n)

Ce pseudo-élément est très pratique pour cibler un élément particulier d'une liste pouvant avoir un nombre aléatoire d'éléments par exemple.

```
li:nth-last-child(2) {  
  text-decoration: underline;  
}
```

Ici l'avant dernier élément des listes sera toujours souligné.

X:nth-of-type(n)

Cette pseudo-classe permet de cibler un ou plusieurs éléments en fonction de l'ordre d'origine selon une formule construite en utilisant la syntaxe $an+b$.

```
li:nth-of-type(3n+2) {  
  background: #ccc;  
}
```

Dans cet exemple les éléments 2, 5, 8, 11, 14, 17... d'une liste auront un fond gris.

Plus simple

```
ul:nth-of-type(1) {  
  background: #ccc;  
}
```

Dans cet exemple la 1^{ère} liste d'éléments aura un fond gris.

X:first-child

Cette pseudo-classe permet de cibler le premier enfant de son parent.

```
ul li:first-child {  
  border-top: none;  
}
```

Ici on désactive la bordure du haut du premier élément d'une liste.

X:last-child

Cette pseudo-classe permet de cibler le dernier enfant de son parent.

```
ul li:last-child {  
  border-bottom: none;  
}
```

Ici on désactive la bordure du bas du dernier élément d'une liste.

X:only-child

Cette pseudo-classe permet de cibler les éléments qui sont enfants uniques de leur parent.

```
div p:only-child {  
  font-style: italic;  
}
```

Dans cet exemple s'il n'existe qu'un seul paragraphe dans un div, alors il sera mis en italique. S'il y en a plus d'un, tous les paragraphes resteront par défaut.

X:only-of-type

Permet de cibler tous les éléments sans aucun frère au sein de l'élément parent.

```
li:only-of-type {  
  font-weight: bold;  
}
```

Ici seuls les éléments uniques d'une *liste imbriquée* seront mis en gras.

X:first-of-type

Cette pseudo-classe permet de cibler un élément qui est le premier enfant de son type dans la liste des enfants de l'élément parent. Il est similaire à [nth-of-type\(1\)](#).

```
p:first-of-type {  
  color: blue;  
}
```

Ici chaque premier type d'élément compris dans un paragraphe sera bleu. Imaginons qu'il y ait plusieurs ``, `` et ``, chaque premier élément `` `` `` sera de couleur bleu, tous les autres resteront de la couleur par défaut.

Sélecteurs CSS (simples et complexes) – Liste et utilisation

Le CSS va nous permettre de mettre en forme nos contenus HTML en appliquant des styles aux différents éléments. Cependant, pour appliquer un style particulier à un ou plusieurs éléments HTML en CSS, il va avant tout falloir les cibler, c'est-à-dire indiquer avec précision à quels éléments doivent s'appliquer les styles créés en CSS.

Il existe de nombreux types de sélecteurs CSS et autant de moyens de cibler des contenus HTML en CSS :

- On peut cibler un type d'élément (comme l'élément `p` par exemple) en précisant le nom générique du type d'élément auquel on souhaite appliquer certains styles en CSS (auquel cas les styles s'appliqueront à tous les éléments de ce type) ;
- On peut donner un nom précis à un élément en HTML à l'aide d'un attribut `class` ou `id` par exemple et ensuite cibler l'élément ou les éléments en CSS grâce à leurs attributs ;
- On peut utiliser ce qu'on appelle des sélecteur CSS combineurs, c'est-à-dire combiner des sélecteurs CSS simples pour en créer de complexes nous permettant de cibler des contenus HTML de manière très précise ;
- On peut encore cibler les éléments HTML grâce au fait qu'ils possèdent un attribut en particulier ou qu'un attribut possède une valeur ;
- On peut cibler les éléments HTML en CSS en utilisant des pseudo classes et ainsi n'appliquer certains styles à ces éléments que lors d'un état particulier ;
- On peut cibler les éléments HTML en utilisant les pseudo éléments en CSS et donc n'appliquer des styles qu'à une certaine partie de l'élément sélectionné.

Les sélecteurs de type « éléments HTML » sont appelés des sélecteurs CSS simples car il suffit d'indiquer le nom d'un élément pour lui attribuer des styles.

En revanche, l'utilisation de caractères spéciaux pour naviguer dans notre page HTML et cibler un élément en particulier va créer ce qu'on appelle des sélecteurs CSS complexes car un sélecteur complexe va être composé de différents sélecteurs juxtaposés.

Liste complète de tous les sélecteurs CSS (sélecteurs CSS3 et CSS4 inclus)

Vous pourrez trouver ci-dessous une liste exhaustive de tous les sélecteurs CSS que l'on va pouvoir utiliser pour cibler du contenu HTML. Attention cependant : certains des sélecteurs de cette liste sont encore en phase expérimentale et ne vont pas fonctionner avec tous les navigateurs tout simplement car ils seront intégrés dans la prochaine grande version du CSS : le CSS4.

Pour cette raison, le tableau ci-dessous contient la version CSS à laquelle est attachée chaque sélecteur afin que vous puissiez rapidement voir et savoir si un sélecteur va pouvoir être directement utilisable ou pas.

Nous détaillerons dans la suite de ce tutoriel le fonctionnement de chacun des sélecteurs CSS supportés par la majorité des navigateurs.

Sélecteur	Description	Version CSS
E	Sélectionne tous les élément de type E	1
E.maclasse	Sélectionne les éléments E possédant un attribut <code>class="maclasse"</code>	1
E#monid	Sélectionne un élément E possédant un attribut <code>id="monid"</code>	1
E:link	Sélectionne tout élément E représentant l'ancree d'un lien non visité jusqu'à présent	1
E:visited	Sélectionne tout élément E représentant l'ancree d'un lien déjà visité	1
E, F	Sélectionne tous les éléments de type E et de type F	1
E F	Sélectionne tous les éléments F à l'intérieur des éléments E	1
E:active	Sélectionne un élément E qui est actuellement cliqué	1
E::first-letter	Sélectionne la première lettre de tout élément E	1
E::first-line	Sélectionne la première ligne (dans le résultat final) de tout élément E	1
*	Sélectionne tous les éléments	2
E > F	Sélectionne les éléments F enfants directs des éléments E	2
E + F	Sélectionne tout élément F placé directement après un élément E	2
E[foo]	Sélectionne tout élément E possédant un attribut <code>foo</code>	2
E[foo="bar"]	Sélectionne tout élément E possédant un attribut <code>foo</code> dont la valeur est exactement « bar »	2
E[foo~="bar"]	Sélectionne tout élément E possédant un attribut <code>foo</code> dont la valeur contient distinctement « bar » (c'est-à-dire dont la valeur contient le mot « bar » séparé du reste par des espaces)	2
E[foo]="en"]	Sélectionne tout élément E possédant un attribut <code>foo</code> dont la valeur commence par « en » séparé du reste par un tiret (ou hyphen en anglais)	2
E:hover	Sélectionne un élément E qui se situe sous le curseur de la souris	2
E:focus	Sélectionne un élément E qui a le focus	2
E:first-child	Sélectionne tout élément E étant le premier enfant de son parent	2

E:lang(fr)	Sélectionne tout élément E dont l'attribut langage possède la valeur « fr »	2
E::after	Ce sélecteur permet d'intégrer du contenu après un élément E	2
E::before	Ce sélecteur permet d'intégrer du contenu avant un élément E	2
E ~ F	Sélectionne tout élément F placé après un élément E dans la page	3
E[foo^="bar"]	Sélectionne tout élément E possédant un attribut foo dont la valeur commence exactement par « bar »	3
E[foo\$="bar"]	Sélectionne tout élément E possédant un attribut foo dont la valeur se termine exactement par « bar »	3
E[foo*="bar"]	Sélectionne tout élément E possédant un attribut foo dont la valeur contient la valeur « bar »	3
E:target	Sélectionne un élément E contenant une ancre qui vient d'être cliquée à partir d'un lien ancre	3
E:enabled	Sélectionne tout élément E avec lequel l'utilisateur peut interagir et qui est activé	3
E:disabled	Sélectionne tout élément E avec lequel l'utilisateur peut interagir et qui est désactivé	3
E:root	Sélectionne un élément E racine du document	3
E:empty	Sélectionne tout élément E qui ne possède pas d'enfant (ni de noeud de type texte)	3
E:nth-child(n)	Sélectionne tout élément E étant le n-ième enfant de son parent	3
E:nth-last-child(n)	Sélectionne tout élément E étant le n-ième enfant de son parent en comptant les enfants à partir du dernier	3
E:checked	Sélectionne tout élément E de type input coché au sens large (checked ou selected)	3
E:last-child	Sélectionne tout élément E étant le dernier enfant de son parent	3
E:only-child	Sélectionne tout élément E qui est le seul enfant de son parent	3
E:nth-of-type(n)	Sélectionne tout élément E étant le n-ième enfant d'un certain type par rapport à son parent	3
E:nth-last-of-type(n)	Sélectionne tout élément E étant le n-ième enfant d'un certain type par rapport à son parent en comptant à partir de la fin	3
E:first-of-type	Sélectionne tout élément E premier enfant de son type par rapport à son parent	3
E:last-of-type	Sélectionne tout élément E dernier enfant de son type par rapport à son parent	3
E:only-of-type	Sélectionne tout élément E seul enfant de son type par rapport à son parent	3

E:read-write	Sélectionne tout élément E de type input avec lequel l'utilisateur peut interagir (comme un champ dans lequel il peut écrire par exemple)	3-UI/4
E:read-only	Sélectionne tout élément E de type input avec lequel l'utilisateur ne peut pas interagir (éléments possédant un attribut disabled par exemple)	3-UI/4
E:placeholder-shown	Sélectionne tout élément E qui affiche actuellement la valeur de son attribut placeholder	3-UI/4
E:default	Sélectionne un élément E dans une liste ou un groupe qui est l'élément défini par défaut	3-UI/4
E:valid	Sélectionne tout élément E de type input dont la valeur est évaluée comme valide (dont la valeur possède une forme correspondant à ce qui est attendu)	3-UI/4
E:invalid	Sélectionne tout élément E de type input dont la valeur est évaluée comme invalide (valeur ne correspondant pas à ce qui est attendu)	3-UI/4
E:in-range	Sélectionne tout élément E de type input dont la valeur fournie se situe dans une fourchette de valeurs prédéfinies	3-UI/4
E:out-of-range	Sélectionne tout élément E de type input dont la valeur fournie se situe en dehors d'une certaine fourchette de valeurs prédéfinies	3-UI/4
E:required	Sélectionne tout élément E de type input dont la valeur doit être renseignée (élément possédant un attribut required)	3-UI/4
E:optional	Sélectionne tout élément E de type input dont la valeur ne doit pas obligatoirement être renseignée	3-UI/4
E:not(E1, .c1)	Sélectionne tout élément E qui n'est pas de type E1 et qui ne possède pas d'attribut class= «c1»	3/4
E::selection	Permet de sélectionner une partie active d'un élément E déjà sélectionnée par l'utilisateur	3/4
E >> F	Sélectionne tous les éléments F à l'intérieur des éléments E. Ce sélecteur est strictement équivalent au précédent mais représente la nouvelle syntaxe de celui-ci qui sera rendue officielle avec le CSS4.	4
F E	Sélectionne un élément E qui représente une cellule dans un tableau ou dans une grille appartenant à une colonne représentée par un élément F	4
E[foo="bar" i]	Sélectionne tout élément E possédant un attribut foo dont la valeur est exactement « bar » sans tenir compte de la casse (l'utilisation de majuscules / minuscules). Le « i » représente ici « case insensitive » c'est-à-dire « insensible à la casse » en français	4
E:any-link	Sélectionne tout élément E étant l'ancree d'un lien, que celui-ci ait été cliqué ou non. Utiliser cette pseudo classe est équivalent à utiliser :link et :visited	4

E:scope	Sélectionne un élément E étant l'élément de référence désigné.	4
E:current	Sélectionne un élément E actuellement affiché ou lu au sein d'un canvas comportant une dimension temporelle ou l'un de ses parents	4
E:current()	Alternative à E:current. On va pouvoir passer une liste de sélecteurs en argument de :current(). L'élément sélectionné sera l'élément :current qui correspond au(x) sélecteur(s) passé(s)	4
E:past	Sélectionne un élément E dont l'affichage précède immédiatement celui de l'élément :current	4
E:future	Sélectionne un élément E dont l'affichage suit immédiatement celui de l'élément :current	4
E:drop	Sélectionne un élément E qui peut potentiellement recevoir un objet HTML dans le cadre d'un glissé-déposé (drag and drop)	4
E:drop(active)	Version alternative de E:drop. Sélectionne l'élément E qui est actuellement la cible d'un drag and drop pour l'élément que l'on est en train de faire glisser	4
E:drop(valid)	Version alternative de E:drop. Sélectionne l'élément E qui est actuellement la cible d'un drag and drop pour l'élément que l'on est en train de faire glisser uniquement si cet élément peut recevoir l'objet qui va y être déposé	4
E:drop(invalid)	Sélectionne tout élément E qui ne peut pas recevoir l'objet actuellement glissé mais qui pourrait recevoir un autre type d'objet	4
E:indeterminate	Sélectionne tout élément E de type input qui est dans un état indéterminé, c'est-à-dire ni coché ni non-coché	4
E:user-error	Sélectionne tout élément E de type input altéré par l'utilisateur et dont la valeur passée est invalide	4
E:blank	Sélectionne tout élément E ne possédant pas de contenu à part éventuellement des espaces	4
E:nth-column(n)	Sélectionne tout élément E représentant une cellule appartenant à la n-ième colonne d'un tableau ou d'une grille	4
E:nth-last-column(n)	Sélectionne tout élément E représentant une cellule appartenant à la n-ième colonne d'un tableau ou d'une grille en comptant à partir de la fin	4
:playing	Sélectionne tout média capable d'être joué (audio, vidéo...) lorsque l'élément en question est en train d'être joué (ou lu)	4
:paused	Sélectionne tout média capable d'être joué (audio, vidéo...) lorsque l'élément en question n'est pas en cours de lecture (en pause, en cours de chargement etc.)	4
E:matches(E1, .c1)	Sélectionne tout élément E qui est soit de type E1 ou qui possède un attribut class= »c1"	4

E:has(E1, .c1)	Sélectionne un élément E si au moins un des sélecteurs passés en paramètre correspond à l'élément (selon la portée, :scope, de l'élément)	4
E:dir(ltr)	Sélectionne tout élément E dont on a spécifié une direction de lecture « de gauche à droite » (la direction est à priori déterminée par la langue déclarée dans le document. Par exemple, pour le français, ce sera de gauche à droite par défaut)	4
E::inactive-selection	Permet de sélectionner une partie inactive d'un élément E déjà sélectionnée par l'utilisateur	4
E::spelling-error	Sélectionne une partie d'un élément E qui a été identifiée par le user agent comme mal orthographiée	4
E::grammar-error	Sélectionne une partie d'un élément E qui a été identifiée par le user agent comme grammaticalement incorrecte	4
E::marker	Permet de sélectionner le marqueur de tout élément E dont le type d'affichage est display : list-item (par exemple la puce ou le numéro de l'élément d'un élément li)	4
E::placeholder	Sélectionne le texte de l'attribut placeholder d'un élément E de type input	4

