

HTML5 & CSS3





Le HTML5







Le fonctionnement des sites web

- **HTML** (*HyperText Markup Language*) : il a fait son apparition dès 1991 lors du lancement du Web. Son rôle est de gérer et organiser le contenu..
- **CSS** (*Cascading Style Sheets*, aussi appelées *Feuilles de style*) : le rôle du CSS est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...). Ce langage est venu compléter le HTML en 1996.
- Lu par un navigateur




[Wikipédia, l'encyclopédie libre](#)

[fr.wikipedia.org](#)
[www.Wikipedia.fr](#)
[Accueil](#)
[Principes](#)



[Accueil](#)
[Discussion](#)
[Lien](#)
[Voir la source](#)
[Afficher l'historique](#)
[Rechercher](#)



[Wikipédia](#)
[L'encyclopédie libre](#)

[Actualité](#)
[Portails thématiques](#)
[Index alphabétique](#)
[Afficher au hasard](#)
[Contacter l'administrateur](#)

[Contribuer](#)
[Premiers pas](#)
[Aide](#)
[Communauté](#)
[Modifications récentes](#)
[Faire un don](#)

[Wikipédia](#)
[L'encyclopédie libre](#)

[Wike Loves Monuments](#)
[Photographier un monument historique](#)
[aider Wikipédia](#)
[et gagner ?](#)

[Bienvenue sur Wikipédia](#)

[Le projet d'encyclopédie libre qui vous propose d'améliorer](#)

[Collaborer](#)
[Géographier](#)
[Traduire](#)
[Traductions](#)
[Soutenir](#)
[Soutiens](#)
[Soutiens](#)

[Liste des portails thématiques](#)

[Wikipédia](#)
[L'encyclopédie libre](#)

[1 512 131 articles en français](#)
[Modifier par exemple une page](#)

HTML
(pas de CSS)



HTML + CSS



Les versions du HTML

- **HTML 1** : c'est la toute première version créée par Tim Berners-Lee en 1991.
- **HTML 2** : la deuxième version du HTML apparaît en 1994 et prend fin en 1996 avec l'apparition du HTML 3.0. Cette version posera les bases des versions suivantes du HTML.
- **HTML 3** : apparue en 1996, cette nouvelle version du HTML ajoute de nombreuses possibilités au langage comme **les tableaux, les applets, les scripts, le positionnement du texte autour des images, etc.**
- **HTML 4** : cette version aura été utilisée un long moment durant les années 2000. Elle apparaît pour la première fois en 1998 et propose **l'utilisation de frames** (qui découpent une page web en plusieurs parties), **des tableaux plus complexes, des améliorations sur les formulaires**, etc. Mais surtout, cette version permet **pour la première fois d'exploiter des feuilles de style CSS**
- **HTML 5** : dernière version. De plus en plus répandue, elle apporte de nombreuses améliorations comme la possibilité d'inclure facilement **des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires, etc.**



Les versions du CSS

- **CSS 1** : dès 1996, on dispose de la première version du CSS. Elle pose les bases de ce langage qui permet de présenter sa page web, comme les couleurs, les marges, les polices de caractères, etc.
- **CSS 2** : apparue en 1999 puis complétée par CSS 2.1, cette nouvelle version de CSS ajoute de nombreuses possibilités. On peut désormais utiliser des techniques de positionnement très précises, qui nous permettent d'afficher des éléments où on le souhaite sur la page.
- **CSS 3** : c'est la dernière version, qui apporte des fonctionnalités particulièrement attendues comme les bordures arrondies, les dégradés, les ombres, etc.



**Créons une première
page**





Balises

Les balises

Les pages HTML sont remplies de balises. Les balises se repèrent facilement. Elles sont entourées de « chevrons », c'est-à-dire des symboles <et>, comme ceci : **<balise>**

Les balises en paires

Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Voici à quoi elles ressemblent :

```
<titre>Ceci est un titre</titre>
```

Les balises orphelines

Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis (par exemple une image). Il n'est pas nécessaire de délimiter le début et la fin de l'image, on veut juste dire à l'ordinateur « Insère une image ici ».

```
<image />
```



Attributs

Les attributs

Les attributs sont les options des balises. Ils viennent les compléter pour donner des informations supplémentaires.

L'attribut se place après le nom de la balise ouvrante et a le plus souvent une valeur, comme ceci :

```
<balise attribut="valeur">
```



Structure d'une page HTML5

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Titre</title>
6   </head>
7
8   <body>
9
10  </body>
11 </html>
```



Les balises structurantes

En-tête : `<header></header>`

Pied de page : `<footer></footer>`

Lien de navigation : `<nav></nav>`

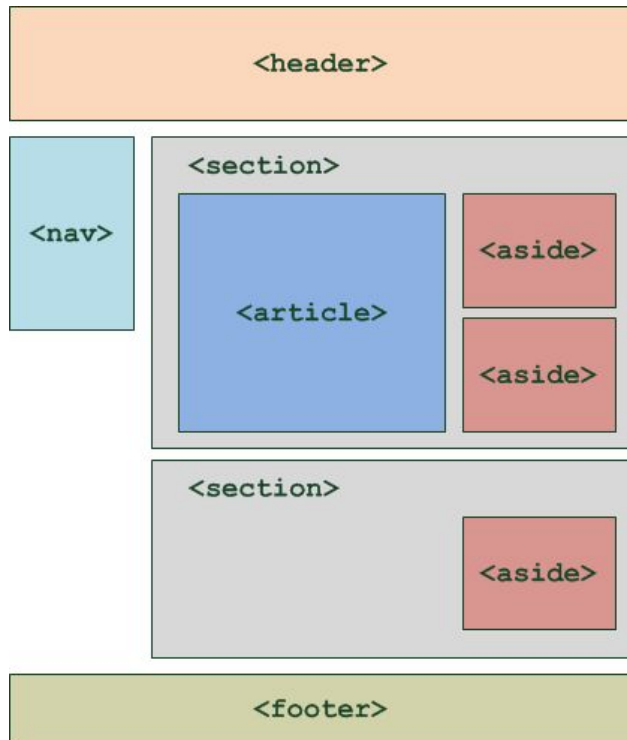
Section de page : `<section></section>`

Infos complémentaires : `<aside></aside>`

Un article : `<article></article>`



Les balises structurantes





Les commentaires

```
1 <!-- Ceci est un commentaire -->
```

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!-- En-tête de la page -->
5     <meta charset="utf-8" />
6     <title>Titre</title>
7   </head>
8
9   <body>
10    <!-- Corps de la page -->
11  </body>
12 </html>
```



Organiser le contenu

Les paragraphes :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Paragraphes</title>
6   </head>
7
8   <body>
9     <p>Bonjour et bienvenue sur mon site !</p>
10  </body>
11 </html>
```



Les titres

- `<h1> </h1>`: signifie « titre très important ». En général, on s'en sert pour afficher le titre de la page au début de celle-ci.
- `<h2> </h2>`: signifie « titre important ».
- `<h3> </h3>`: pareil, c'est un titre un peu moins important (on peut dire un « sous-titre » si vous voulez).
- `<h4> </h4>`: titre encore moins important.
- `<h5> </h5>`: titre pas important.
- `<h6> </h6>`: titre vraiment, mais alors là vraiment pas important du tout.



La mise en valeur

- Pour mettre *un peu* en valeur votre texte, vous devez utiliser la balise `` ``.
- Pour mettre un texte bien en valeur, on utilise la balise ``
- Pour faire ressortir visuellement une portion du texte : `<mark>`
`</mark>`

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8" />
5   <title>Le Titre</title>
6 </head>
7
8 <body>
9   <p>Bonjour et bienvenue sur dans ce cours !<br />
10  Ceci est la manière de mettre en valeur un <em>mot ou une phrase</em> en HTML, vous pouvez aussi utiliser la balise <strong>STRONG</strong> pour
    montrer quelque chose de très important !</p>
11 </body>
12 </html>
```



Les listes

```
1 <ul>
2   <li>Fraises</li>
3   <li>Framboises</li>
4   <li>Cerises</li>
5 </ul>
```

```
1 <h1>Ma journée</h1>
2
3 <ol>
4   <li>Je me lève.</li>
5   <li>Je mange et je bois.</li>
6   <li>Je retourne me coucher.</li>
7 </ol>
```



Les liens

```
<a href="http://www.epsi.fr">EPSI</a>
```

```
1 <p>Bonjour. Souhaitez-vous consulter <a href="page2.html">la page 2</a> ?</p>
```

```
1 <a href="contenu/page2.html">
```

```
1 <a href="../page2.html">
```

```
1 <h2 id="mon_ancre">Titre</h2>
```

```
1 <a href="#mon_ancre">Aller vers l'ancre</a>
```

```
1 <h1>Ma grande page</h1>
2
3 <p>
4     Aller directement à la partie traitant de :<br />
5     <a href="#cuisine">La cuisine</a><br />
6     <a href="#rollers">Les rollers</a><br />
7     <a href="#arc">Le tir à l'arc</a><br />
8 </p>
9 <h2 id="cuisine">La cuisine</h2>
10
11 <p>... (beaucoup de texte) ...</p>
12
13 <h2 id="rollers">Les rollers</h2>
14
15 <p>... (beaucoup de texte) ...</p>
16
17 <h2 id="arc">Le tir à l'arc</h2>
18
19 <p>... (beaucoup de texte) ...</p>
```



Les liens

Lien vers une ancre située dans une autre page

```
1 <h1>Le Mégamix</h1>
2 <p>
3   Rendez-vous quelque part sur une autre page :<br />
4   <a href="ancres.html#cuisine">La cuisine</a><br />
5   <a href="ancres.html#rollers">Les rollers</a><br />
6   <a href="ancres.html#arc">Le tir à l'arc</a><br />
7 </p>
```



Les liens

Un lien qui ouvre une nouvelle fenêtre

Il est possible de « forcer » l'ouverture d'un lien dans une nouvelle fenêtre. Pour cela, on ajoutera `target="_blank"` à la balise `<a>`

Un lien qui affiche une infobulle au survol

Vous pouvez utiliser l'attribut `title` qui affiche une bulle d'aide lorsqu'on pointe sur le lien. Cet attribut est facultatif.

```
<p>Bonjour. Souhaitez-vous visiter <a href="http://www.epsi.fr" title="Le site de l'école EPSI" target="_blank">Le site de l'EPSI</a> ?</p>
```

Bonjour. Souhaitez-vous visiter [Le site de l'EPSI](http://www.epsi.fr) ?

Le site de l'école EPSI



Les liens

Un lien pour envoyer un e-mail

Si vous voulez que vos visiteurs puissent vous envoyer un e-mail: `mailto`.

Rien ne change au niveau de la balise, vous devez simplement modifier la valeur de l'attribut `href` comme ceci :

```
1 <p><a href="mailto:votrenom@bidule.com">Envoyez-moi un e-mail !</a></p>
```



Les images

Il existe un format adapté à chaque image

Quel format adopter en fonction de l'image que vous avez :

- **Une photo** : utilisez un JPEG.
- **N'importe quel graphique avec peu de couleurs** (moins de 256) : utilisez un PNG 8 bits ou éventuellement un GIF.
- **N'importe quel graphique avec beaucoup de couleurs** : utilisez un PNG 24 bits.
- **Une image animée** : utilisez un GIF animé.



Les images

Insérer une image

Balise : ``

C'est une balise orpheline comme `
`.

La balise doit être accompagnée de deux attributs obligatoires :

- `src` : il permet d'indiquer où se trouve l'image que l'on veut insérer. Vous pouvez soit mettre un chemin absolu (ex. : `http://www.site.com/fleur.png`), soit mettre le chemin en relatif (`src="images/fleur.png"`)
- `alt` : « texte alternatif ». **OBLIGATOIRE**. Ce texte sera affiché à la place de l'image si celle-ci ne peut pas être téléchargée ou dans les navigateurs de personnes handicapées (non-voyants). Cela aide aussi les robots des moteurs de recherche pour les recherches d'images. Pour la fleur, on mettrait par exemple : `alt="Une fleur"`.

```
1 <p>
2   Voici une photo que j'ai prise lors de mes dernières vacances à la montagne :<br />
3   
4 </p>
```



Les images

Création d'une figure

En HTML5, on dispose de la balise `<figure>`.

Une figure est le plus souvent accompagnée d'une légende. Pour ajouter une légende, utilisez la balise `<figcaption>` à l'intérieur de la balise `<figure>`

```
1 <figure>
2   
3   <figcaption>Le logiciel Bloc-Notes</figcaption>
4 </figure>
```



Le logiciel Bloc-Notes



A vous de créer votre CV



Le CSS3





Où écrit-on le CSS ?

Vous avez le choix car on peut écrire du code en langage CSS à trois endroits différents :

- dans un fichier `.css` (*méthode la plus recommandée*) :
`<link rel="stylesheet" href="style.css" />`
- dans l'en-tête `<head>` du fichier HTML ;
- directement dans les balises du fichier HTML via un attribut `style` (*méthode la moins recommandée*).

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <link rel="stylesheet" href="style.css" />
6     <title>Premiers tests du CSS</title>
7   </head>
8
9   <body>
10    <h1>Mon super site</h1>
11
12    <p>Bonjour et bienvenue sur mon site !</p>
13    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>
14  </body>
15 </html>
```



Appliquer un style : sélectionner une balise

Dans un code CSS, on trouve trois éléments différents :

- **Des noms de balises** : on écrit les noms des balises dont on veut modifier l'apparence. Par exemple, si je veux modifier l'apparence de tous les paragraphes `<p>`, je dois écrire `p`.
- **Des propriétés CSS** : les « effets de style » de la page sont rangés dans des propriétés. Par exemple : `color`
- **Les valeurs** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété `color`, il faut indiquer le nom de la couleur.

```
1 p
2 {
3     color: blue;
4 }
```

```
1 balise1
2 {
3     propriete1: valeur1;
4     propriete2: valeur2;
5     propriete3: valeur3;
6 }
7
8 balise2
9 {
10     propriete1: valeur1;
11     propriete2: valeur2;
12     propriete3: valeur3;
13     propriete4: valeur4;
14 }
15
16 balise3
17 {
18     propriete1: valeur1;
19 }
```




Appliquer un style : class et id

Comment cibler une balise précise :

- l'attribut `class`;
- l'attribut `id`.

```
1 <h1 class=""> </h1>
2 <p class=""> </p>
3 <img class="" />
```

Appliquer un style : class et id

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <link rel="stylesheet" href="style.css" />
6     <title>Premiers tests du CSS</title>
7   </head>
8
9   <body>
10    <h1>Mon super site</h1>
11
12    <p class="introduction">Bonjour et bienvenue sur mon site !</p>
13    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>
14  </body>
15 </html>
```

```
1 .introduction
2 {
3   color: blue;
4 }
```



Appliquer un style à plusieurs balises

```
1 h1
2 {
3   color: blue;
4 }
5
6 em
7 {
8   color: blue;
9 }
```



```
1 h1, em
2 {
3   color: blue;
4 }
```



Appliquer un style à plusieurs balises

*** : sélecteur universel**

Sélectionne toutes les balises sans exception. On l'appelle le sélecteur universel.

```
1 *  
2 {  
3  
4 }
```

A B : une balise contenue dans une autre

Sélectionne toutes les balises `` situées à l'intérieur d'une balise `<h3>`.

```
1 h3 em  
2 {  
3  
4 }
```



Appliquer un style à plusieurs balises

A + B : une balise qui en suit une autre

Sélectionne la première balise `<p>` située après un titre `<h3>`.

```
1 h3 + p
2 {
3
4 }
```

A[attribut] : une balise qui possède un attribut

Sélectionne tous les liens `<a>` qui possèdent un attribut `title`

```
1 a[title]
2 {
3
4 }
```



Appliquer un style à plusieurs balises

A[attribut="Valeur"] : une balise, un attribut et une valeur exacte

Idem, mais l'attribut doit en plus avoir exactement pour valeur « Cliquez ici ».

```
1 a[title="Cliquez ici"]
2 {
3
4 }
```

A[attribut*="Valeur"] : une balise, un attribut et une valeur

Idem, l'attribut doit cette fois contenir dans sa valeur le mot « ici » (peu importe sa position).

```
1 a[title*="ici"]
2 {
3
4 }
```



Formater du texte



La Taille

Pour modifier la taille du texte, on utilise la propriété CSS `font-size`.

- Indiquer une **taille absolue** : en pixels, en centimètres ou millimètres. Cette méthode est très précise mais il est conseillé de ne l'utiliser que si c'est absolument nécessaire, car on risque d'indiquer une taille trop petite pour certains lecteurs.
- Indiquer une **taille relative** : en pourcentage, « em » ou « ex », cette technique a l'avantage d'être plus souple. Elle s'adapte plus facilement aux préférences de taille des visiteurs.



La Taille

Une valeur relative

C'est la méthode recommandée car le texte s'adapte alors plus facilement aux préférences de tous les visiteurs.

Il y a plusieurs moyens d'indiquer une valeur relative. Vous pouvez par exemple écrire la taille avec des mots en anglais comme ci-dessous ou en "em" :

- `xx-small`: minuscule ;
- `x-small`: très petit ;
- `small`: petit ;
- `medium`: moyen ;
- `large`: grand ;
- `x-large`: très grand ;
- `xx-large`: très très grand.



La police

Modifier la police utilisée

La propriété CSS qui permet d'indiquer la police à utiliser est `font-family`. Vous devez écrire le nom de la police comme ceci :

```
1 balise
2 {
3   font-family: police;
4 }
```



La police

Liste non-exhaustive :

- Arial ;
- Arial Black ;
- Comic Sans MS ;
- Courier New ;
- Georgia ;
- Impact ;
- Times New Roman ;
- Trebuchet MS ;
- Verdana.



La police

Pour indiquer la police de mes paragraphes, j'écris donc :

```
1 p
2 {
3   font-family: Impact, "Arial Black", Arial, Verdana, sans-serif;
4 }
```



La police

Vous pouvez également utiliser des polices personnalisées !

```
1 @font-face {  
2   font-family: 'MaSuperPolice';  
3   src: url('MaSuperPolice.eot') format('eot'),  
4       url('MaSuperPolice.woff') format('woff'),  
5       url('MaSuperPolice.ttf') format('truetype'),  
6       url('MaSuperPolice.svg') format('svg');  
7 }
```



Mettre en Italique

Pour mettre en forme le texte on utilise `font-style` qui peut prendre les 3 valeurs ci-dessous :

- `italic`: le texte sera mis en italique.
- `oblique`: le texte sera passé en oblique (les lettres sont penchées, le résultat est légèrement différent de l'italique proprement dit).
- `normal`: le texte sera normal (par défaut). Cela vous permet d'annuler une mise en italique.

```
1 h2
2 {
3   font-style: italic;
4 }
```



Mettre en gras

La propriété CSS pour mettre en gras est `font-weight` et prend les valeurs suivantes :

- `bold`: le texte sera en gras ;
- `normal`: le texte sera écrit normalement

```
1 h1
2 {
3   font-weight: bold;
4 }
```



Soulignement et autres

La propriété CSS associée porte bien son nom : `text-decoration`. Elle permet, entre autres, de souligner le texte, mais pas seulement. Voici les différentes valeurs qu'elle peut prendre :

- `underline`: souligné.
- `line-through`: barré.
- `overline`: ligne au-dessus.
- `blink`: clignotant. Ne fonctionne pas sur tous les navigateurs (Internet Explorer et Google Chrome, notamment).
- `none`: normal (par défaut).



L'alignement

Le langage CSS nous permet de faire tous les alignements connus : à gauche, centré, à droite et justifié.

C'est tout simple. On utilise la propriété `text-align` et on indique l'alignement désiré :

- `left`: le texte sera aligné à gauche (c'est le réglage par défaut).
- `center`: le texte sera centré.
- `right`: le texte sera aligné à droite.
- `justify`: le texte sera « justifié ». Justifier le texte permet de faire en sorte qu'il prenne toute la largeur possible sans laisser d'espace blanc à la fin des lignes. Les textes des journaux, par exemple, sont toujours justifiés.




Les Float

Le CSS nous permet de faire flotter un élément autour du texte. On dit aussi qu'on fait un « habillage ».

Pour cela on utilise la propriété `float`. Cette propriété peut prendre deux valeurs très simples :

- `left`: l'élément flottera à gauche.
- `right`: l'élément flottera à droite !



Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Donec
vitae lorem imperdiet lacus molestie
molestie. Cum sociis natoque penatibus
et magnis dis parturient montes, nascetur ridiculus
mus. Donec eu purus. Phasellus metus lorem,
blandit et, posuere quis, tincidunt vitae, ante.
Vivamus consequat mauris a diam. Vivamus nibh
erat, hendrerit nec, aliquet ut, hendrerit quis, nunc.
Vestibulum et turpis et elit tempor euismod.



Stopper un Float

Il existe une propriété CSS qui permet de dire : « *Stop, ce texte doit être en-dessous du flottant et non plus à côté* ». C'est la propriété `clear`, qui peut prendre ces trois valeurs :

- `left`: le texte se poursuit en-dessous après un `float: left`;
- `right`: le texte se poursuit en-dessous après un `float: right`;
- `both`: le texte se poursuit en-dessous, que ce soit après un `float: left`; ou après un `float: right`;

```
1 <p></p>
2 <p>Ce texte est écrit à côté de l'image flottante.</p>
3 <p class="dessous">Ce texte est écrit sous l'image flottante.</p>
```

```
1 .imageflottante
2 {
3     float: left;
4 }
5 .dessous
6 {
7     clear: both;
8 }
```



Couleur du texte

Pour déterminer la couleur du texte on utilise la propriété `color`.

On peut lui donner les valeurs suivantes :

- nom de la couleur en anglais : `red`
- Code hexadécimal : `#ffffff`
- Code RGB : `rgb(240,96,204)`

```
1 p
2 {
3   color: #FFFFFF;
4 }
```



La couleur de fond

Pour indiquer une couleur de fond, on utilise la propriété CSS `background-color`.

Elle s'utilise de la même manière que la propriété `color`, c'est-à-dire que vous pouvez taper le nom d'une couleur, l'écrire en notation hexadécimale ou encore utiliser la méthode RGB.

```
1 /* On travaille sur la balise body, donc sur TOUTE la page */
2 body
3 {
4     background-color: black; /* Le fond de la page sera noir */
5     color: white; /* Le texte de la page sera blanc */
6 }
```



Image de fond

La propriété permettant d'indiquer une image de fond est `background-image`. Comme valeur, on doit renseigner `url("nom_de_l_image.png")`.

```
1 body
2 {
3     background-image: url("neige.png");
4 }
```



Image de fond

Options disponibles pour l'image de fond

On peut compléter la propriété `background-image` que nous venons de voir par plusieurs autres propriétés qui permettent de changer le comportement de l'image de fond.

`background-attachment`: fixer le fond

La propriété CSS `background-attachment` permet de « fixer » le fond. L'effet obtenu est intéressant car on voit alors le texte « glisser » par-dessus le fond. Deux valeurs sont disponibles :

- `fixed`: l'image de fond reste fixe ;
- `scroll`: l'image de fond défile avec le texte (par défaut).



Image de fond

Options disponibles pour l'image de fond

background-repeat: répétition du fond

Par défaut, l'image de fond est répétée en mosaïque. Vous pouvez changer cela avec la propriété `background-repeat`:

- `no-repeat`: le fond ne sera pas répété. L'image sera donc unique sur la page.
- `repeat-x`: le fond sera répété uniquement sur la première ligne, horizontalement.
- `repeat-y`: le fond sera répété uniquement sur la première colonne, verticalement.
- `repeat`: le fond sera répété en mosaïque (par défaut).



Image de fond

Options disponibles pour l'image de fond

background-position: position du fond

On peut indiquer où doit se trouver l'image de fond avec `background-position`. Cette propriété n'est intéressante que si elle est combinée avec `background-repeat: no-repeat;` (un fond qui ne se répète pas).

Vous devez donner à `background-position` deux valeurs en pixels pour indiquer la position du fond par rapport au coin supérieur gauche de la page (ou du paragraphe, si vous appliquez le fond à un paragraphe)

```
1 background-position: 30px 50px;
```



Image de fond

Options disponibles pour l'image de fond

background-position: position du fond

Il est aussi possible d'utiliser ces valeurs en anglais :

- top: en haut ;
- bottom: en bas ;
- left: à gauche ;
- center: centré ;
- right: à droite.

Il est possible de combiner ces mots. Par exemple, pour aligner une image en haut à droite, vous taperez :

```
1 background-position: top right;
```



Image de fond

si je veux afficher un soleil en image de fond, en un unique exemplaire (`no-repeat`), toujours visible (`fixed`) et positionné en haut à droite (`top right`), je vais écrire ceci :

```
1 body
2 {
3     background-image: url("soleil.png");
4     background-attachment: fixed; /* Le fond restera fixe */
5     background-repeat: no-repeat; /* Le fond ne sera pas répété */
6     background-position: top right; /* Le fond sera placé en haut à droite */
7 }
```



Les super-propriétés

Si vous utilisez beaucoup de propriétés en rapport avec le fond (comme c'est le cas sur ce dernier exemple), vous pouvez utiliser une sorte de « super-propriété » appelée `background` dont la valeur peut combiner plusieurs des propriétés vues précédemment : `background-image`, `background-repeat`, `background-attachment` et `background-position`.

On peut donc tout simplement écrire :

```
1 body
2 {
3     background: url("soleil.png") fixed no-repeat top right;
4 }
```



La transparence

La propriété `opacity`

La propriété `opacity`, très simple, permet d'indiquer le niveau d'opacité (c'est l'inverse de la transparence).

- Avec une valeur de 1, l'élément sera totalement opaque : c'est le comportement par défaut.
- Avec une valeur de 0, l'élément sera totalement transparent.

Il faut donc choisir une valeur comprise entre 0 et 1. Ainsi, avec une valeur de 0.6, votre élément sera opaque à 60% et on verra donc à travers !

```
1 p
2 {
3   opacity: 0.6;
4 }
```



La transparence

La notation RGBa

CSS3 nous propose une autre façon de jouer avec la transparence : la notation RGBa.

```
1 p
2 {
3   background-color: rgba(255, 0, 0, 0.5); /* Fond rouge à moitié transparent */
4 }
```



Les bordures

Le CSS offre un large choix de bordures pour décorer votre page.

De nombreuses propriétés CSS vous permettent de modifier l'apparence de vos bordures :

`border-width, border-color, border-style...`

Mais nous utiliserons la super-propriété `border`



Les bordures

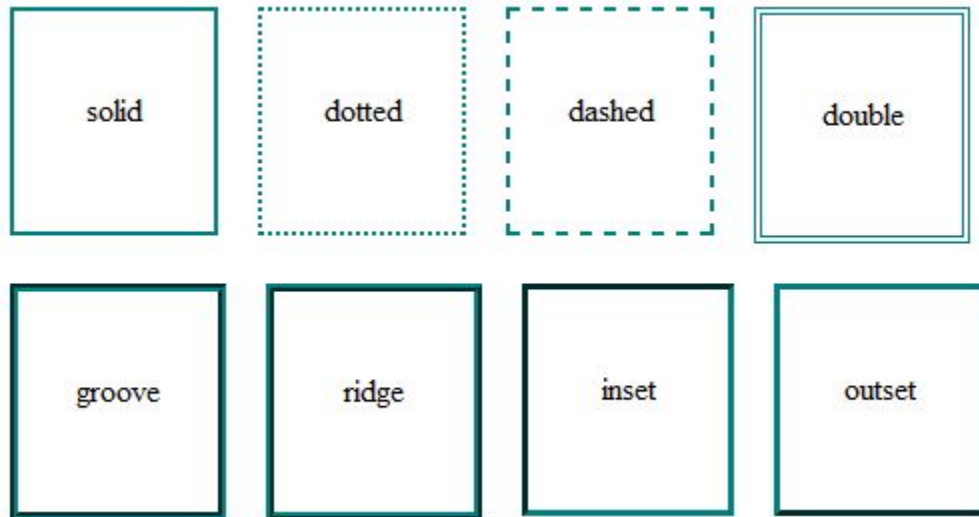
Pour `border` on peut utiliser jusqu'à trois valeurs pour modifier l'apparence de la bordure :

- **La largeur** : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme `2px`).
- **La couleur** : c'est la couleur de votre bordure. Utilisez, comme on l'a appris, soit un nom de couleur (`black, red, ...`), soit une valeur hexadécimale (`#FF0000`), soit une valeur RGB (`rgb(198, 212, 37)`).
- **Le type de bordure** : là, vous avez le choix. Votre bordure peut être un simple trait, ou des pointillés, ou encore des tirets, etc. Voici les différentes valeurs disponibles :
 - `none`: pas de bordure (par défaut) ;
 - `solid`: un trait simple ;
 - `dotted`: pointillés ;
 - `dashed`: tirets ;
 - `double`: bordure double ;
 - `groove`: en relief ;
 - `ridge`: autre effet relief ;
 - `inset`: effet 3D global enfoncé ;
 - `outset`: effet 3D global surélevé.



Les bordures

Les types de bordures





Les bordures

Pour cibler les différents côtés des bordures :

- `border-top`: bordure du haut ;
- `border-bottom`: bordure du bas ;
- `border-left`: bordure de gauche ;
- `border-right`: bordure de droite.

```
1 p
2 {
3   border-left: 2px solid black;
4   border-right: 2px solid black;
5 }
```



Les bordures arrondies

La propriété `border-radius` va nous permettre d'arrondir facilement les angles de n'importe quel élément. Il suffit d'indiquer la taille (« l'importance ») de l'arrondi en pixels

```
1 p
2 {
3   border-radius: 10px;
4 }
```

On peut aussi préciser la forme de l'arrondi pour chaque coin. Dans ce cas, indiquez quatre valeurs :

```
1 p
2 {
3   border-radius: 10px 5px 10px 5px;
4 }
```



Les ombres

Nous allons ici découvrir deux types d'ombres :

- les ombres des boîtes ;
- les ombres du texte.

box-shadow: les ombres des boîtes

La propriété `box-shadow` s'applique à tout le bloc et prend quatre valeurs dans l'ordre suivant :

1. le décalage horizontal de l'ombre ;
2. le décalage vertical de l'ombre ;
3. l'adoucissement du dégradé ;
4. la couleur de l'ombre.

```
1 p
2 {
3   box-shadow: 6px 6px 0px black;
4 }
```



Les ombres

```
1 p
2 {
3   box-shadow: 6px 6px 0px black;
4 }
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.



Les ombres

text-shadow: l'ombre du texte

Avec `text-shadow`, vous pouvez ajouter une ombre directement sur les lettres de votre texte ! Les valeurs fonctionnent exactement de la même façon que `box-shadow`: décalage, adoucissement et couleur.

```
1 p
2 {
3   text-shadow: 2px 2px 4px black;
4 }
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.



Les apparences dynamiques

Au survol :

:hover

```
1 a /* Liens par défaut (non survolés) */
2 {
3     text-decoration: none;
4     color: red;
5     font-style: italic;
6 }
7
8 a:hover /* Apparence au survol des liens */
9 {
10    text-decoration: underline;
11    color: green;
12 }
```



Les apparences dynamiques

au moment du clic :

`:active`

```
1 a:active /* Quand le visiteur clique sur le lien */
2 {
3     background-color: #FFCC66;
4 }
```




Les apparences dynamiques

Lorsque l'élément est sélectionné :

`:focus`

```
1 a:focus /* Quand le visiteur sélectionne le lien */  
2 {  
3     background-color: #FFCC66;  
4 }
```

ou encore `:visited` sur un lien déjà visité



La cascade

L'ordre de préférence et d'application d'un style va dépendre de trois grands facteurs qui vont être :

- La présence ou non du mot clef **important** ;
- La précision du sélecteur ;
- L'ordre de déclaration dans le code ;



La cascade- la précision du sélecteur

- Un style défini dans un attribut HTML `style` sera toujours le plus précis et notamment plus précis qu'un style défini avec un sélecteur CSS ;
- Le sélecteur `#id` va être le sélecteur le plus précis mais sera moins précis qu'un style défini dans un attribut HTML `style` ;
- Un sélecteur `.class` ou un autre sélecteur d'attribut sera moins précis qu'un sélecteur `#id` ;
- Un sélecteur d'élément ou de pseudo-élément sera moins précis qu'un sélecteur d'attribut ou de pseudo-classe.

En gros :

style > id > classe > class:after > class:hover



L'héritage

La notion d'héritage est une autre notion fondamentale du CSS. Elle signifie que **certains styles CSS** appliqués à un élément **vont être hérités par les éléments enfants** de cet élément, c'est-à-dire par les **éléments contenus dans cet élément**.

Cette notion d'héritage est conditionnée par deux choses :

- Toutes les propriétés ne vont pas être héritées car cela ne ferait aucun sens pour certaines de l'être ;
- Les éléments enfants n'hériteront des styles de leur parent que si il n'y a pas de conflit c'est-à-dire uniquement dans la situation où ces mêmes styles n'ont pas été redéfinis pour ces éléments enfants en CSS.



L'héritage

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

- Élément de liste 1
 - Autre élément de liste

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>

    <p>Un premier paragraphe</p>
    <p>Un autre paragraphe</p>

    <ul>
      <li>Élément de liste 1</li>
      <li id="marge">Autre élément de liste</li>
    </ul>

  </body>
</html>
```

```
html{
  font-family: "Lucida Grande", sans-serif;
}

h1{
  font-family: "Times New Roman", serif
}

ul{
  margin-left: 50px;
  background-color: lightBlue
}

#marge{
  margin-left: 50px;
}
```



L'héritage

On peut toutefois forcer l'héritage avec 4 valeurs de propriété:

Valeur	Signification
<code>inherit</code>	Sert à indiquer que la valeur de propriété appliquée à l'élément sélectionné est la même que celle de l'élément parent
<code>initial</code>	Sert à indiquer que la valeur de propriété appliquée à l'élément sélectionné est la même que celle définie pour cet élément dans la feuille de style par défaut du navigateur
<code>unset</code>	Permet de réinitialiser la propriété à sa valeur naturelle, ce qui signifie que si la propriété est naturellement héritée elle se comporte comme si on avait donné la valeur <code>inherit</code> . Dans le cas contraire, son comportement sera le même que si on lui avait donné la valeur <code>initial</code>



L'héritage

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>

    <p>Un premier paragraphe</p>
    <p>Un autre paragraphe</p>

    <ul id="liste1">
      <li>Elément de liste 1</li>
      <li>Autre élément de liste</li>
    </ul>

    <ul id="liste2">
      <li>Elément de liste 1</li>
      <li>Autre élément de liste</li>
    </ul>
  </body>
</html>
```

```
html{
  font-family: "Lucida Grande", sans-serif;
}

/*Pour info, la police utilisée par défaut pour les h1 ici est Times*/
h1{
  font-family: initial;
}

ul{
  background-color: lightBlue;
}
#liste1{
  margin-left: 50px;
}
#liste2{
  margin-left: 10px;
}

/*On demande explicitement à ce que les éléments li héritent du même
comportement pour la propriété margin-left que leur parent (qui est ul)*/
li{
  margin-left: inherit;
}
```



L'héritage

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

- Élément de liste 1
- Autre élément de liste

- Élément de liste 1
- Autre élément de liste



Inline et Block : les différents types d'éléments

Deux grands types d'affichage : Inline et Block

Ces types d'affichage vont définir la façon dont les éléments vont se comporter dans une page par rapport aux autres et la place qu'ils vont prendre dans la page.

Ces deux types se comportent de façon radicalement différente.

Le CSS a des comportements différents selon le type



Inline et Block : les différents types d'éléments

Le type d'affichage d'un élément va toujours être défini en CSS par la propriété `display`.

Si cette propriété n'est pas explicitement renseignée pour un élément, c'est la valeur par défaut de `display` qui va être appliquée à l'élément c'est-à-dire `display: inline`.

Cependant, chaque navigateur possède une feuille de styles pour la définition des styles basiques.

Parmi ces styles par défaut appliqués par n'importe quel navigateur se trouve la définition du type d'affichage ou du `display` pour chaque élément.



Inline et Block : les différents types d'éléments

L'immense majorité des navigateurs va appliquer les recommandations W3C ce qui fait que l'un des `display` suivants est attribué à la plupart des éléments HTML en fonction de l'élément :

- `display : block` : affichage sous forme d'un bloc ;
- `display : inline` : affichage en ligne ;
- `display : none` : l'élément n'est pas affiché.

Le W3C préconise d'autres types d'affichage pour certains éléments HTML particuliers. Les deux autres valeurs de `display` généralement respectées et appliquées par les navigateurs sont :

- `display : list-item` va être appliquée par défaut pour les éléments de liste `li`. L'affichage se fait sous forme de bloc mais une boîte avec un marqueur est également générée ;
- `display : table` va être appliquée par défaut pour les éléments de tableau `table`. L'affichage se fait sous forme de bloc.



Les éléments Inline

Les éléments de type `inline` vont posséder les caractéristiques suivantes qui vont les différencier des éléments de type `block` :

- Un élément de type `inline` ne va occuper que la largeur nécessaire à l'affichage de son contenu par défaut ;
- Les éléments de type `inline` vont venir essayer de se placer en ligne, c'est-à-dire à côté (sur la même ligne) que l'élément qui les précède dans le code HTML ;
- Un élément de type `inline` peut contenir d'autres éléments de type `inline` mais ne devrait pas contenir d'éléments de type `block`.

Note : impossible de gérer `width` ou `height` à un élément de type `inline`



Les éléments Inline

Les éléments HTML dont le type d'affichage recommandé par le W3C est le type `inline` les plus courants sont les suivants :

- Les éléments de distinction d'importance du contenu `em` et `strong` ;
- L'élément `span` ;
- L'élément de liens `a` ;
- L'élément `button` ;
- Les éléments de formulaire `input`, `label`, `textarea` et de liste de choix `select` ;
- L'élément d'insertion d'images `img` (cas intéressant et souvent source de confusions car on va pouvoir passer une largeur et une hauteur à l'image à afficher en soi qui va « remplacer » l'élément `img` lors de l'affichage, mais il n'empêche que l'élément `img` est bien `inline` en soi);
- Les éléments `code`, `script`, etc.



Les éléments Block

Les éléments de type `block` vont posséder les caractéristiques de disposition suivantes :

- Un élément de type `block` va toujours prendre toute la largeur disponible au sein de son élément parent (ou élément conteneur) ;
- Un élément de type `block` va toujours « aller à la ligne » (créer un saut de ligne avant et après l'élément), c'est-à-dire occuper une nouvelle ligne dans une page et ne jamais se positionner à côté d'un autre élément par défaut ;
- Un élément de type `block` peut contenir d'autres éléments de type `block` ou de type `inline`.



Les éléments Block

Les éléments HTML dont le type d'affichage recommandé par le W3C est le type `block` les plus communs sont les suivants :

- L'élément `body`, cas particulier mais qui est concrètement considéré comme un élément `block` ;
- L'élément de division du contenu `div` ;
- Les paragraphes `p` et titres `h1`, `h2`, `h3`, `h4`, `h5`, `h6` ;
- Les éléments structurants `article`, `aside`, `header`, `footer`, `nav` et `section` ;
- Les listes `ul`, `ol` ;
- L'élément de définition de tableaux `table` ;
- L'élément de définition de formulaires `form` et l'élément `fieldset` ;
- Les éléments `figure` et `figcaption` ;
- Les éléments `canvas`, `video`, etc.



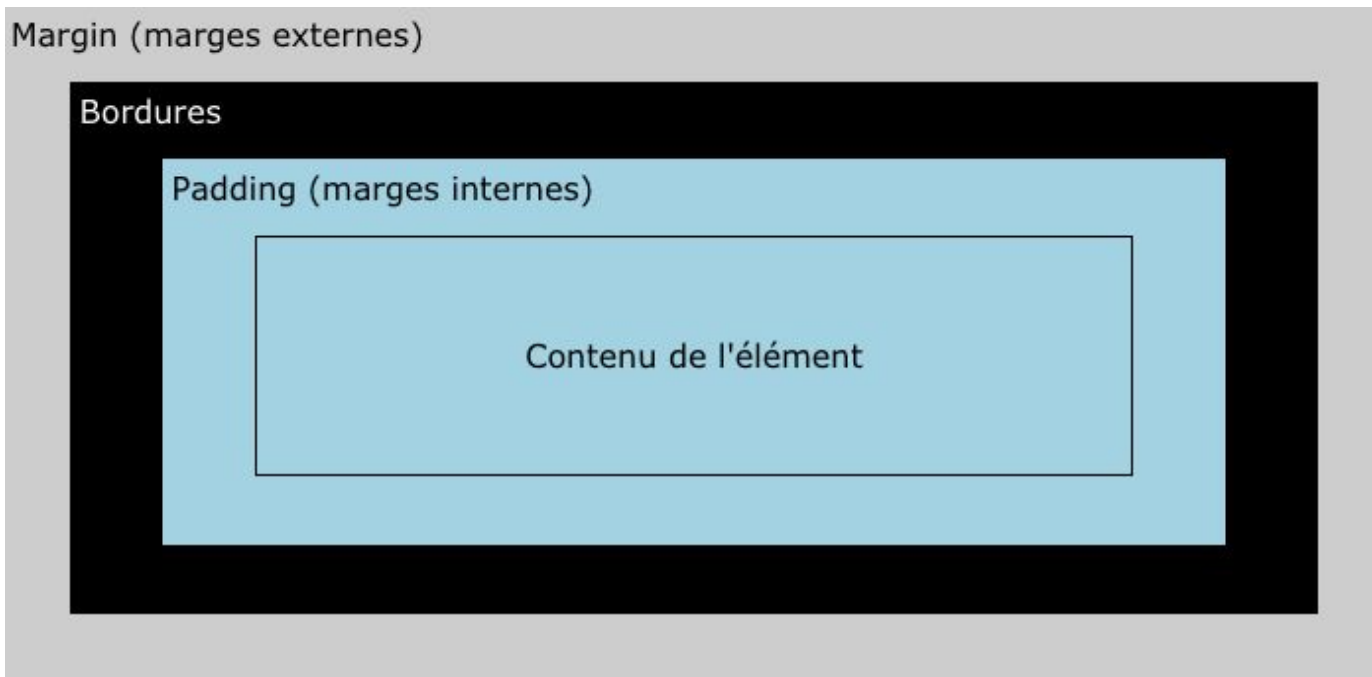
Le Modèle des boîtes

L'idée centrale du modèle des boîtes est que tout élément HTML peut être représenté par un empilement de différentes boîtes rectangulaires :

- La **première boîte**, centrale, va être composée du **contenu de l'élément en soi** ;
- La **deuxième boîte** va être composée **de la première boîte ainsi que des marges internes** de l'élément ;
- La **troisième boîte** va être composée de la **deuxième boîte et des bordures de l'élément** ;
- La **quatrième boîte** va être composée de la **troisième boîte et des marges externes** de l'élément.



Le Modèle des boîtes





Le Modèle des boîtes

Le CSS va déjà nous fournir différentes propriétés qui vont nous permettre de spécifier la taille des différents éléments composant les différentes boîtes :

- Les propriétés `width` et `height` vont nous permettre de définir la largeur et la hauteur de la boîte « contenu » ;
- La propriété `padding` va nous permettre de définir la taille des marges internes ;
- La propriété `border` va nous permettre de définir des bordures pour notre élément ;
- La propriété `margin` va nous permettre de définir la taille des marges externes.



Width & Height

Les propriétés `width` et `height` vont pouvoir accepter plusieurs types de valeurs :

- Des valeurs de type « longueur » qui vont être généralement exprimées en `px` ou en `em` ;
- Des valeurs en pourcentage, auquel cas le pourcentage donné sera relatif à la dimension de l'élément parent.

En cas de dépassement : utiliser `overflow` avec pour valeur `hidden` ou `scroll`



Les marges internes - Padding

Nous allons également pouvoir appliquer des marges intérieures de taille différentes de chaque côté d'un élément.

Pour cela, nous avons deux façons de faire : soit en passant plusieurs valeurs à la propriété `padding`, soit en utilisant les propriétés `padding-top`, `padding-left`, `padding-bottom` et `padding-right`.



Les marges internes - Padding

On va également pouvoir passer entre 1 et 4 valeurs à la super-propriété `padding` :

- En passant une valeur à `padding`, la valeur passée définira le comportement des 4 marges intérieures de l'élément ;
- En passant deux valeurs à `padding`, la première valeur passée définira le comportement des marges intérieures supérieure et inférieure de l'élément tandis que la seconde valeur définira le comportement des marges intérieures gauche et droite de l'élément ;
- En passant trois valeurs à `padding`, la première valeur passée définira le comportement de la marge interne supérieure, la deuxième définira le comportement des marges intérieures gauche et droite tandis que la troisième définira le comportement de la marge interne basse
- En passant quatre valeurs à `padding`, la première valeur passée définira le comportement de la marge interne supérieure, la deuxième définira le comportement de la marge interne droite, la troisième celui de la marge interne basse et la quatrième celui de la marge interne gauche.



Les marges externe - Margin

La propriété est en fait la notation raccourcie des propriétés `margin-top`, `margin-left`, `margin-bottom` et `margin-right` qui vont servir à définir les marges externes de chaque côté d'un élément.

Ces propriétés vont pouvoir accepter différents types de valeurs :

- Des valeurs de type longueur, généralement en `px` ou en `em` ;
- Des valeurs de type pourcentage. Dans ce cas, le `%` indiqué est calculé par rapport à la taille de l'élément parent ;
- Le mot clef `auto`. Ce mot clef va surtout nous servir pour centrer des blocs.

On peut également utiliser la super-propriété `margin`



Les marges externe - Margin

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>

    <p id="p1">Un premier paragraphe</p>
    <p id="p2">Un deuxième paragraphe</p>
    <p id="p3">Un <span>troisième</span> paragraphe</p>
    <p id="p4">Un quatrième paragraphe</p>
    <p id="p5">Un cinquième paragraphe</p>
    <p id="p6">Un sixième paragraphe</p>
    <p id="p7">Un septième paragraphe</p>
    <p id="p8">Un huitième paragraphe</p>
  </body>
</html>
```

```
/*On reset les marges pour ensuite bien les redéfinir*/
body, p{
  margin: 0px;
}

p{
  background-color: lightBlue;
}

#p3{
  margin: 20px;
}

span{
  margin: 100px;
  background-color: yellow;
}

#p6{
  margin-top: 40px;
  margin-right: 40px;
  margin-bottom: 40px;
  margin-left: 40px;
}
```



Les marges externe - Margin

Un titre de niveau 1

Un premier paragraphe
Un deuxième paragraphe

Un troisième paragraphe

Un quatrième paragraphe
Un cinquième paragraphe

Un sixième paragraphe

Un septième paragraphe
Un huitième paragraphe

Les Tableaux



Les marges externe - Margin

Un tableau est un ensemble de lignes et de colonnes. L'intersection entre une ligne et une colonne est une cellule de tableau.

Pour créer un tableau fonctionnel en HTML, nous allons devoir utiliser à minima 3 éléments :

- Un élément `table` qui va définir (balises ouvrante `<table>` et fermante `</table>`)
- Des éléments `tr`, pour « table row » qui vont nous permettre d'ajouter des lignes dans notre tableau ;
- Des éléments `td`, pour « table data » qui vont nous permettre d'ajouter des cellules dans nos lignes et ainsi de créer automatiquement de nouvelles colonnes.



Créer un tableau simple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <table>
      <tr>
        <td>Nom</td>
        <td>Prénom</td>
        <td>Age</td>
        <td>Mail</td>
      </tr>
      <tr>
        <td>Giraud</td>
        <td>Pierre</td>
        <td>28</td>
        <td>pierre.giraud@edhec.com</td>
      </tr>
      <tr>
        <td>Joly</td>
        <td>Pauline</td>
        <td>27</td>
        <td>pjl@gmail.com</td>
      </tr>
    </table>
  </body>
</html>
```



Nom	Prénom	Age	Mail
Giraud	Pierre	28	pierre.giraud@edhec.com
Joly	Pauline	27	pjl@gmail.com



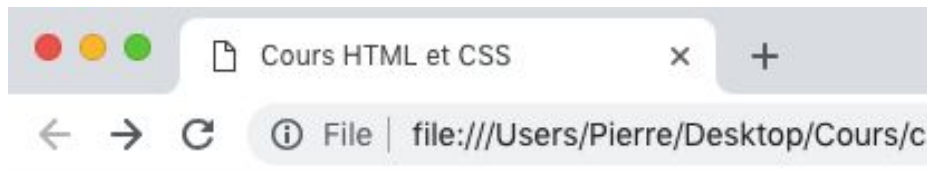
Créer un tableau structuré

On peut distinguer **trois parties** dans un tableau : **le haut du tableau** contenant généralement la ligne d'en tête, **le corps du tableau** contenant les informations de notre tableau en soi et **le pied du tableau** servant à calculer des totaux ou à rappeler les informations d'en tête si votre tableau est vraiment long.

Pour définir chaque partie d'un tableau, nous disposons d'un élément HTML spécifique :

- `thead` pour la tête du tableau ;
- `tbody` pour le corps du tableau ;
- `tfoot` pour le pied du tableau.

Créer un tableau structuré



Page	Nb visites 2017	Nb visites 2018	Total
Page 1	100	250	350
Page 2	1200	5400	6600
Total	1300	5650	9950

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <table>
      <thead>
        <tr>
          <th>Page</th>
          <th>Nb visites 2017</th>
          <th>Nb visites 2018</th>
          <th>Total</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Page 1</td>
          <td>100</td>
          <td>250</td>
          <td>350</td>
        </tr>
        <tr>
          <td>Page 2</td>
          <td>1200</td>
          <td>5400</td>
          <td>6600</td>
        </tr>
      </tbody>
      <tfoot>
        <tr>
          <td>Total</td>
          <td>1300</td>
          <td>5650</td>
          <td>9950</td>
        </tr>
      </tfoot>
    </table>
  </body>
</html>
```



Fusionner des cellules colspan / rowspan

Nom	Prénom 1	Prénom 2	Prénom 3	Adresse mail 1	Adresse mail 2	Inscrit ?
Giraud	Pierre	Victor		pierre.giraud@edhec.com		Oui
Joly	Pauline		pjl@gmail.com	jolyp@hotmail.fr		

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <table>
      <thead>
        <tr>
          <th>Nom</th>
          <th>Prénom 1</th>
          <th>Prénom 2</th>
          <th>Prénom 3</th>
          <th>Adresse mail 1</th>
          <th>Adresse mail 2</th>
          <th>Inscrit ?</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Giraud</td>
          <td>Pierre</td>
          <td colspan="2">Victor</td>
          <td colspan="2">pierre.giraud@edhec.com</td>
          <td rowspan="2">Oui</td>
        </tr>
        <tr>
          <td>Joly</td>
          <td colspan="2">Pauline</td>
          <td>pjl@gmail.com</td>
          <td>jolyp@hotmail.fr</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```



Mise en forme des colonnes `colgroup` & `col`

L'élément `colgroup` va représenter un groupe d'une ou plusieurs colonnes dans un tableau.

Cet élément est représenté par une paire de balises et doit être un enfant direct de l'élément `table`, précédé éventuellement uniquement par un élément `caption`. Cela signifie qu'il doit être inséré avant tout élément `thead`, `tbody`, `tfoot`, and `tr` au sein de l'élément `table`.

Au sein d'un élément `colgroup`, nous allons pouvoir insérer autant d'éléments `col` qu'il y a de colonnes dans notre tableau.

Chaque élément `col` représente une colonne de notre tableau et est orpheline

colgroup & col

Informations participants

Nom	Prénom	Adresse mail	Inscrit ?
Giraud	Pierre	pierre.giraud@edhec.com	Oui
Joly	Pauline	pjl@gmail.com	

```
table{  
  border-collapse: collapse;  
}  
  
th, td{  
  border: 1px solid black;  
  padding: 10px;  
}  
  
.info{  
  background-color: lightBlue;  
}
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Cours HTML et CSS</title>  
    <meta charset= "utf-8">  
    <link rel="stylesheet" href="cours.css">  
  </head>  
  
  <body>  
    <table>  
      <caption>Informations participants</caption>  
      <colgroup>  
        <col span="3" class="info">  
        <col>  
      </colgroup>  
      <thead>  
        <tr>  
          <th>Nom</th>  
          <th>Prénom</th>  
          <th>Adresse mail</th>  
          <th>Inscrit ?</th>  
        </tr>  
      </thead>  
      <tbody>  
        <tr>  
          <td>Giraud</td>  
          <td>Pierre</td>  
          <td>pierre.giraud@edhec.com</td>  
          <td rowspan="2">Oui</td>  
        </tr>  
        <tr>  
          <td>Joly</td>  
          <td>Pauline</td>  
          <td>pjl@gmail.com</td>  
        </tr>  
      </tbody>  
    </table>  
  </body>  
</html>
```




Le CSS dans les tableaux

Merci le W3C !

https://www.w3schools.com/css/css_table.asp



A Vous !



Les Formulaires





Les formulaires

Les éléments HTML constitutif d'un formulaire :

- `<form>`
- `<label>`
- `<input>`
- `<textarea>`
- `<button>`

```
<form action="/my-handling-form-page" method="post">  
  
</form>
```



Les formulaires

Les éléments HTML constitutif d'un formulaire :

- `<form>`
- `<label>`
- `<input>`
- `<textarea>`
- `<button>`

Tous les formulaires HTML débutent par un élément `<form>`
Comme ci-dessous :

```
<form action="/my-handling-form-page" method="post">  
  
</form>
```



Les formulaires

`<form>` est un élément conteneur au même titre que les éléments `<div>` ou `<p>`,

Tous ses attributs sont optionnels sauf les attributs `action` et `method` est considéré comme de bonne pratique.



Les formulaires

```
1  <form action="/ma-page-de-traitement" method="post">
2      <div>
3          <label for="name">Nom :</label>
4          <input type="text" id="name" name="user_name">
5      </div>
6      <div>
7          <label for="mail">e-mail :</label>
8          <input type="email" id="mail" name="user_mail">
9      </div>
10     <div>
11         <label for="msg">Message :</label>
12         <textarea id="msg" name="user_message"></textarea>
13     </div>
14 </form>
```



Les formulaires

Pour mettre une valeur par défaut à nos `<input>` on utilise l'attribut `value`

```
1 | <input type="text" value="par défaut cet élément sera renseigné avec ce texte">
```

A l'inverse pour `<textarea>`, il faut inscrire la valeur par défaut au centre des balises

```
1 | <textarea>par défaut cet élément sera renseigné avec ce texte</textarea>
```




Les formulaires

Une fois le formulaire terminé il faut le soumettre !

Ceci se fait simplement en ajoutant d'un élément `<button>` ; ajoutez-le juste avant la balise fermante `</form>`

```
1 | <div class="button">
2 |     <button type="submit">Envoyer le message</button>
3 | </div>
```



Les formulaires

Comme vous le voyez l'élément `<button>` accepte aussi un attribut de type — il peut prendre une des trois valeurs : `submit`, `reset` ou `button`.

- Un clic sur un bouton `submit` (valeur par défaut) envoie les données du formulaire vers la page définie par l'attribut `action` de l'élément `<form>`.
- Un clic sur un bouton `reset` réinitialise tous les widgets du formulaire à leurs valeurs par défaut immédiatement. Du point de vue de l'expérience utilisateur, utiliser un tel bouton est une mauvaise pratique.
- Un clic sur un bouton `button` ne fait... rien ! Cela peut paraître stupide mais c'est en réalité très pratique pour concevoir des boutons personnalisés avec JavaScript.



A Vous !