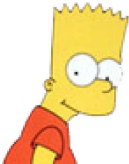




# Mettre en place le CSS

## Informations sur le tutoriel

[Ajouter à mes tutoriels favoris](#) (28 fans)



Auteur : [M@teo21](#)  
Licence :

[Plus d'informations](#)

## Popularité

Visualisations : 741 200 341

Appréciation 15

des lecteurs : 15

60

888

## Publicité

## Historique des mises à jour

- Le 08/08/2010 à 03:23:12  
Correction conjugaison.
- Le 03/08/2010 à 01:07:48  
Correction orthographique, #2605
- Le 31/07/2010 à 18:20:55  
Correction orthographique, ticket n° 2593

Nous y voici enfin 😊

Après avoir passé toute une première partie du cours à ne travailler que sur le XHTML, nous allons maintenant découvrir le CSS que j'avais volontairement mis à l'écart.

Le CSS n'est pas plus compliqué que le XHTML, il est juste différent car il sert à réaliser la présentation de votre page web.

Il y a, vous verrez, pas mal de "propriétés" à connaître (un peu comme les noms des balises XHTML, sauf que là il y en a plus). Heureusement, personne ne vous demande de tout retenir par coeur et un aide-mémoire sera à votre disposition en annexe pour vous aider (aide-mémoire dont j'ai moi-même besoin, je ne retiens pas tout 😊).

Dans ce premier chapitre sur le CSS, nous allons voir la théorie sur le CSS : qu'est-ce que c'est ? A quoi ça ressemble ? Où est-ce qu'on écrit du code CSS ? Cette théorie n'est pas bien compliquée, mais vous devez obligatoirement la connaître car ça constitue la base du CSS. C'est d'ailleurs la seule chose que je vous demanderai de retenir par coeur en CSS, le reste pourra être retrouvé dans le mémo en annexe.

Allez, ne traînons pas, je vois que vous brûlez d'impatience 😊

Sommaire du chapitre :



- [Où mettre du CSS ?](#)
- [Appliquer un style à des balises](#)
- [Utiliser les class et id](#)
- [Q.C.M.](#)

## Où mettre du CSS ?

Vous vous souvenez ce que CSS veut dire ? Je vous en ai rapidement parlé dans le premier chapitre du cours. Cela signifie : "Cascading Style Sheets", ce qui peut se traduire en français par "Feuilles de style en cascade".

On dit "Feuille de style" car en règle générale, on écrit le code CSS dans un fichier à part (à l'extension .css au lieu de .html). C'est un fichier dans lequel on écrit l'apparence que notre site doit avoir : la couleur et la police du texte, la taille des titres, la position des menus, la couleur ou l'image de fond etc... Croyez-moi, le CSS permet de faire une foule de choses ! 😊

On peut écrire du code CSS à 3 endroits différents, selon ce qu'on préfère :

- **Méthode A** : dans un fichier .css (le moyen le plus recommandé)
- **Méthode B** : dans l'en-tête du fichier XHTML
- **Méthode C** : dans les balises

Voici une description de chacune de ces techniques. Si vous ne devez en retenir qu'une, retenez la première :

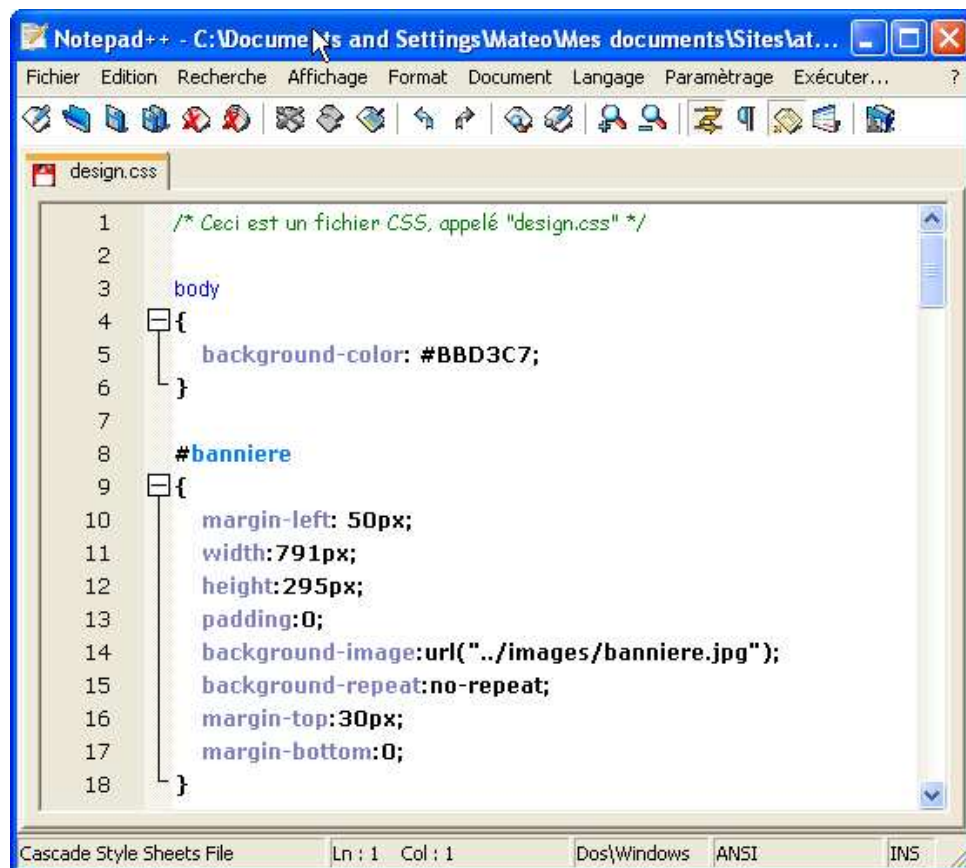
- **Dans un fichier .css (Méthode A) :**

Comme je viens de vous le dire, le plus souvent on écrit du code CSS dans un fichier spécial ayant l'extension .css (contrairement aux fichiers XHTML qui ont l'extension .html).

C'est la méthode la plus pratique que je vais utiliser quasiment tout le temps dans la suite du cours. Parmi les nombreux avantages que cette solution apporte, il y a la possibilité de pouvoir proposer facilement plusieurs designs différents à vos visiteurs.

Si vous utilisez Notepad++, pensez à aller dans le menu "Langage / CSS" pour activer la coloration du CSS. Vous penserez à enregistrer votre fichier en .css au lieu de .html.

Voici un exemple de fichier CSS sous Notepad++ :



```
1  /* Ceci est un fichier CSS, appelé "design.css" */
2
3  body
4  {
5      background-color: #BBD3C7;
6  }
7
8  #banniere
9  {
10     margin-left: 50px;
11     width: 791px;
12     height: 295px;
13     padding: 0;
14     background-image: url("../images/banniere.jpg");
15     background-repeat: no-repeat;
16     margin-top: 30px;
17     margin-bottom: 0;
18 }
```

Si vous voulez voir comment je procède dans une vidéo, cliquez sur le lien ci-dessous :

[Enregistrer un CSS sous Notepad++ \(170 Ko\)](#)

Ce que vous voyez est un aperçu du code CSS que nous allons écrire. Ne vous préoccupez pas de savoir ce que ça veut dire pour le moment, je vais vous expliquer ça un peu plus loin 😊

Maintenant, retournez dans votre fichier XHTML. Il faut y ajouter une ligne entre les balises <head> </head> qui va permettre de "dire" au fichier XHTML quel fichier CSS il doit charger :

#### Code : HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
  <head>
    <title>Exemple d'utilisation de CSS externe</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css" title="Design" href="design.css" />
  </head>
  <body>
    <p>Cette page comporte une feuille de style externe. C'est la meilleure méthode pour le moment.</p>
  </body>
</html>
```

La balise <link /> comporte plusieurs attributs. Vous pouvez en modifier deux d'entre eux pour le moment :

- *title* : c'est le nom que vous donnez à votre feuille de style (mettez ce que vous voulez)
- *href* : c'est l'emplacement où se trouve votre feuille de style sous forme de lien relatif. Dans cet exemple le CSS se trouve dans le même dossier.

#### • Directement dans le header du fichier XHTML (Méthode B) :

Il est aussi possible de taper du CSS directement à l'intérieur même du fichier XHTML, entre les balises <head> </head>. Vous devrez y mettre une balise <style> </style> à l'intérieur, comme ceci :

#### Code : HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
  <head>
    <title>Exemple de CSS dans le header</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <style type="text/css">
      /* Vous taperez votre code CSS ici */
    </style>
  </head>
  <body>
    <p>Cette page comporte du CSS directement dans le header.</p>
  </body>
</html>
```

Cette seconde méthode ressemble beaucoup à la première. Cependant, la première solution (utiliser un .css externe) est quand même bien plus pratique, car elle vous permettra de faire changer le design du site en un clin d'oeil. De plus, le fichier ne sera téléchargé qu'une seule fois pour toutes les pages de votre site, ce qui allégera la taille des fichiers .html et rendra donc votre site plus rapide !

Au risque de me répéter (faut bien que ça rentre :p), je vous recommande donc d'utiliser un fichier .css externe (la première solution proposée) plutôt que de mettre directement le CSS dans le fichier XHTML 😊

#### • La méthode "à l'arrache" dans les balises (Méthode C) :

ça c'est la troisième (et la moins recommandée) des méthodes. Elle consiste à ajouter un attribut *style* sur les balises pour leur appliquer un style particulier. Cet attribut fonctionne sur toutes les balises. Par exemple sur un titre :

#### Code : HTML

```
<h1 style="/* Votre style pour cette balise ici */">Titre</h1>
```

Cette méthode a tous les défauts : non seulement le CSS sera peu lisible à l'intérieur des balises, mais en plus ça ne nous permet pas de profiter de tous les avantages du CSS, comme la possibilité de changer la couleur de tous les titres du site en un clic.

Bref, je vous ai montré la méthode seulement pour que vous ne soyez pas perdus si vous la voyez.

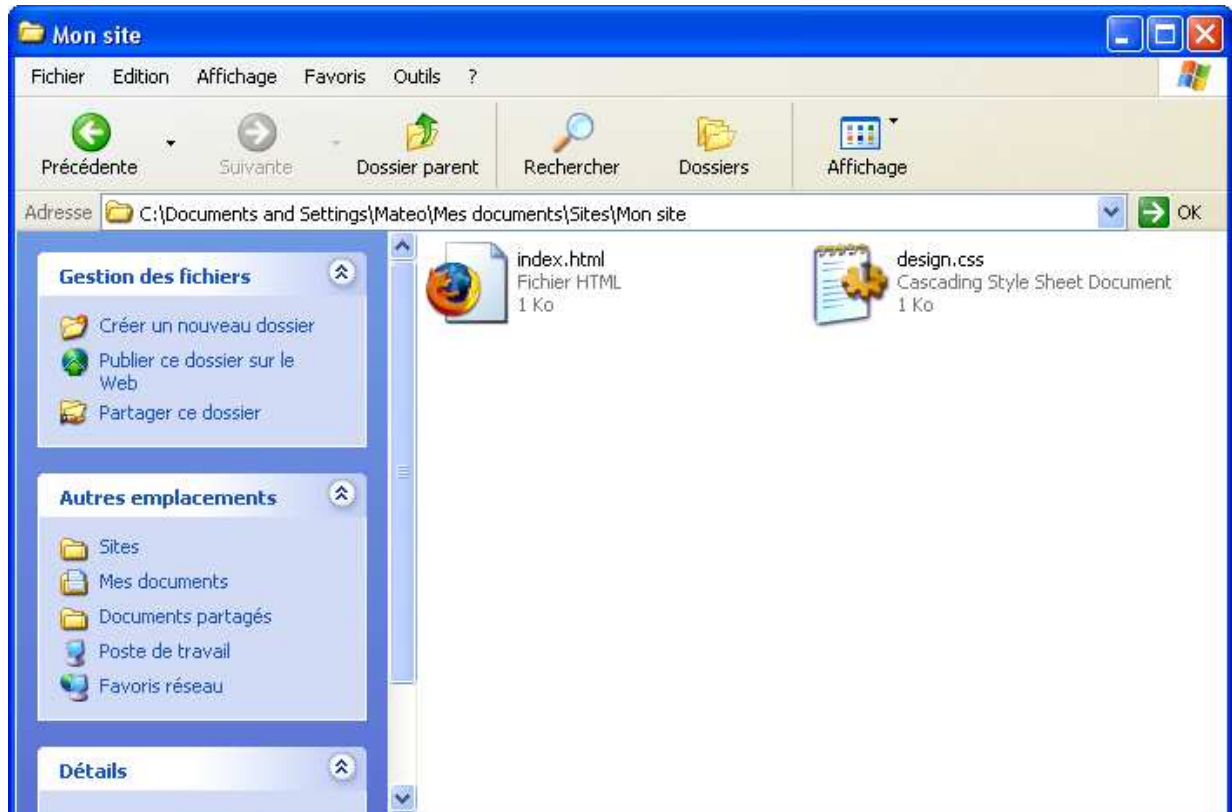
Voilà, vous venez de voir quelles sont les 3 méthodes que l'on peut utiliser pour insérer du CSS (de la meilleure à la pire). L'idéal est donc d'utiliser une feuille de style externe (= mettre son CSS dans un fichier .css) et c'est cette méthode que je vais utiliser dans la suite du cours.

## Résumé visuel

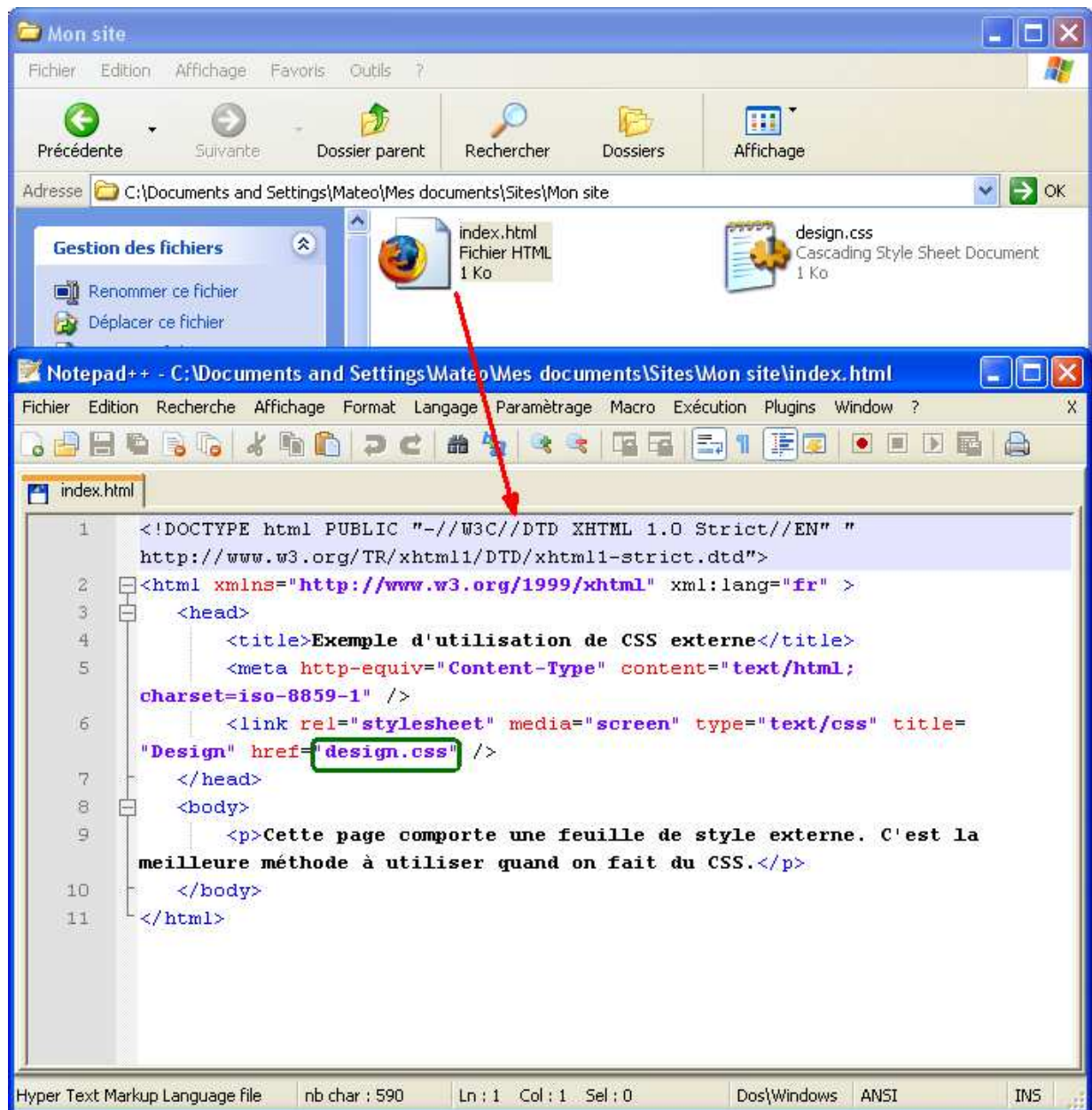
Je suppose ici que vous avez choisi la **Méthode A** que je vous ai conseillée. Jusqu'ici on travaillait sur un seul fichier, le .html. A partir de maintenant on va travailler sur 2 fichiers : un .css et toujours notre bon vieux .html.

Résumons donc les fichiers que nous devons avoir sous les yeux pour que ça soit clair pour tout le monde :

- Dans le dossier de votre site, il doit y avoir au moins 2 fichiers : un .html et un .css.

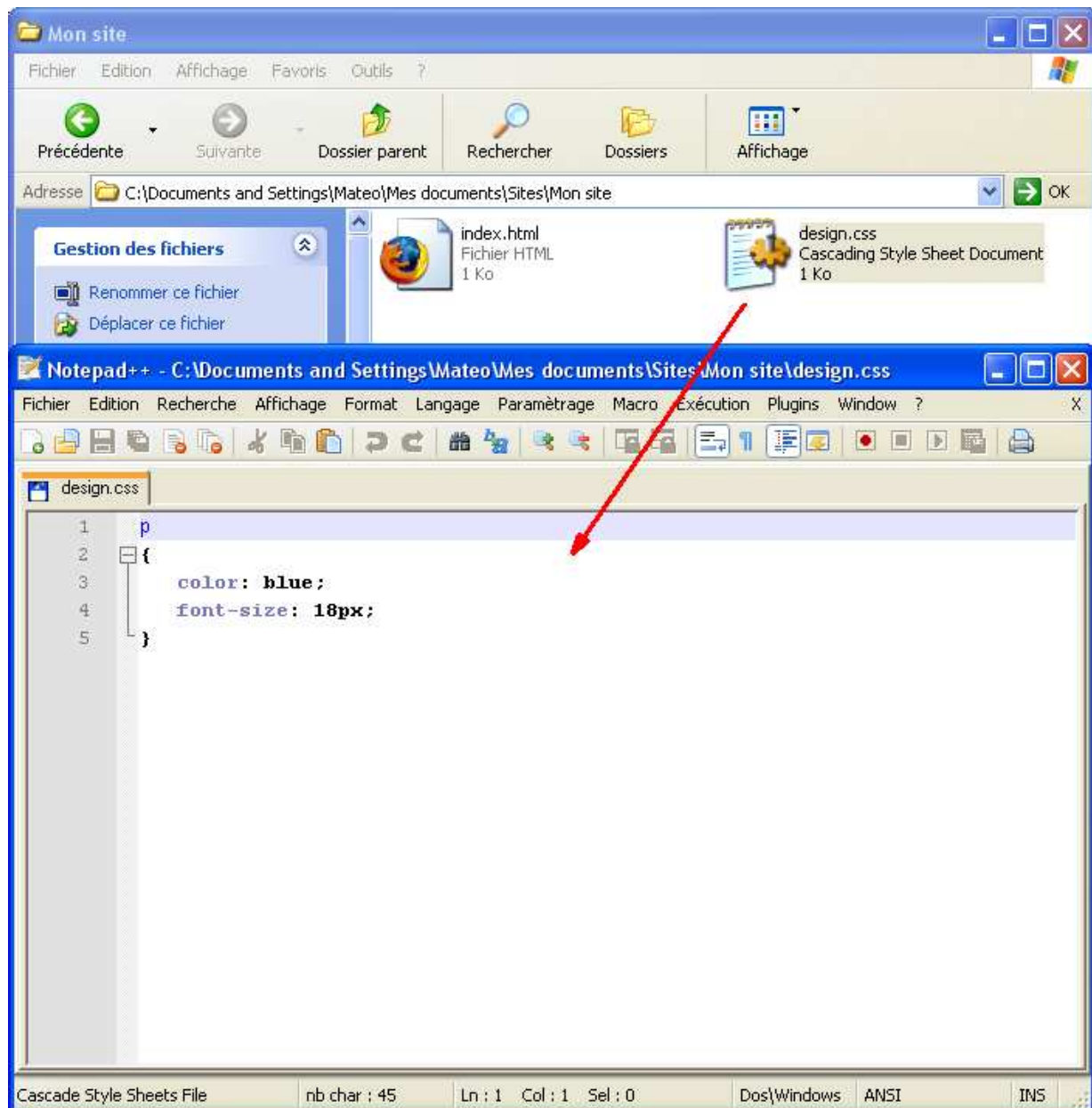


- Le fichier .html contient le code XHTML (qu'on a appris jusqu'ici) avec en particulier la ligne pour faire la liaison avec le fichier .css :



- Le fichier .css contient du code CSS (que nous allons apprendre à partir de maintenant) :





Si vous voulez voir le résultat, il faut ouvrir le fichier .html comme avant (double-cliquez sur l'icône du fichier index.html dans l'explorateur). N'essayez pas d'ouvrir le .css : seul, ce fichier ne sert à rien. C'est le .html qui, lors de son ouverture, va charger le .css et appliquer les informations de style.

Bien, maintenant que cela est clair pour tout le monde (du moins j'espère 😊), intéressons-nous de plus près au fonctionnement du code CSS.

## Appliquer un style à des balises

Dans un CSS, on trouve 3 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 titres <h1>, je dois écrire *h1*
- Des propriétés CSS : les "effets de style" de la page sont rangés dans des propriétés. Il y a par exemple la propriété *color* qui permet d'indiquer la couleur du texte, *font-size* qui permet d'indiquer la taille du texte etc etc... Il y a beaucoup de propriétés CSS et, comme je vous l'ai dit, je ne vous obligerai pas à les connaître toutes par coeur (sauf s'il me prend une envie sadique de vous faire souffrir 😊)
- Les valeurs : à chaque propriété CSS on doit indiquer une valeur. Par exemple, pour la couleur, il faut indiquer le nom de

la couleur. Pour la taille, il faut indiquer quelle taille on veut etc etc...

Schématiquement, une feuille de style CSS ressemble à ça :

**Code : CSS**

```
balise1
{
    propriete: valeur;
    propriete: valeur;
    propriete: valeur;
}
balise2
{
    propriete: valeur;
    propriete: valeur;
    propriete: valeur;
    propriete: valeur;
}
balise3
{
    propriete: valeur;
}
```

Vous repérez sur ce schéma les balises, propriétés et valeurs dont je viens de vous parler.

Comme vous le voyez, on écrit le nom de la balise (par exemple *h1*), et on ouvre des accolades pour y mettre les propriétés et valeurs que l'on veut à l'intérieur.

On peut mettre autant de propriétés que l'on veut à l'intérieur des accolades (enfin, il faut en mettre au moins une sinon ça sert à rien 😊)

Chaque propriété est suivie du symbole "deux-points" (:) puis de sa valeur correspondante. Enfin, chaque ligne se termine par un point-virgule (;)

Le point-virgule n'est pas obligatoire s'il n'y a qu'une seule propriété pour la balise (comme c'est le cas pour la balise 3). Toutefois, il vaut mieux prendre l'habitude de le mettre tout le temps, c'est plus sûr

Je vous apprendrai de nombreuses propriétés dans les chapitres suivants. Pour le moment, on va en utiliser deux ou trois au hasard pour l'exemple, mais rassurez-vous : je vous les expliquerai dans les chapitres suivants plus en détail.

Supposons par exemple que je veuille modifier tous mes paragraphes pour qu'ils soient écrits en bleu, avec une taille de 18 pixels.

Je sais que tous les textes de mes paragraphes sont entre des balises `<p>` `</p>`, je vais donc écrire le code CSS suivant :

**Code : CSS**

```
p
{
    color: blue;
    font-size: 18px;
}
```

Cela signifie en français : *"Je veux que tous mes paragraphes soient écrits en bleu avec une taille de 18 pixels."*

Ne mettez jamais d'espace entre "18" et "px", car sinon le code CSS ne fonctionnera pas !

Enregistrez ce code CSS dans un fichier "essai.css" par exemple.

Il va falloir maintenant créer un fichier XHTML banal pour tester notre CSS. Il ne faut pas oublier de mettre la balise `<link />` dont je vous ai parlé plus haut pour indiquer où se trouve notre fichier CSS.

Voici le fichier de test que j'ai créé, gardez-le dans un coin car il contient de nombreuses balises et il est probable qu'on s'en serve un moment pour les tests de nos futurs CSS :

**Code : HTML**

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
  <head>
    <title>Page d'exemple pour tester le CSS</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css" title="Essai" href="essai
  </head>
  <body>
    <h1>Découverte du CSS</h1>

    <p>
      Bonjour !<br />
      Je suis une page XHTML <em>apparemment</em> banale, mais je sers en fait de
    </p>

    <h2>Ce n'est que du blabla</h2>

    <p>
      Comme l'a dit Neil Armstrong un certain 20 juillet 1969 : <q>C'est un petit
      J'aime la choucroute en conserve.<br />
      Vive les frites !
    </p>
    <p>
      Quoi, ça ne veut rien dire ce que j'écris ? On s'en fout, c'est pour tester
    </p>
  </body>
</html>

```

Le bouton ci-dessous ouvre le fichier exemple.html (on n'ouvre jamais directement un fichier .css) :

[Essayer !](#)

Essayez de modifier le fichier CSS, en mettant une autre couleur ("red" pour rouge par exemple, et "24px" pour avoir un texte écrit très gros 🤪).

Ce qui est génial avec le CSS, c'est que vous pouvez changer l'apparence de tous les paragraphes de vos pages d'un seul coup ! Il suffit de modifier le fichier .css et hop, les modifications sont appliquées 🤪

Pour vous entraîner, essayez d'appliquer des styles CSS aussi à d'autres balises comme les titres (h1, h2, em etc...). Ok, pour le moment je ne vous ai montré que 2 propriétés CSS, mais ça devrait vous suffire pour faire des petits tests 🤪

## Appliquer un style à plusieurs balises

Prenons le code CSS suivant :

Code : CSS

```

h1
{
  color: red;
}
h2
{
  color: red;
}

```

Il signifie que nos balises h1 doivent être en rouge, ainsi que nos balises h2. Oui mais voilà, c'est répétitif car c'est exactement la même chose.

Heureusement, il existe un moyen en CSS d'aller plus vite si les styles de 2 balises doivent avoir la même présentation. Il suffit de séparer les noms des balises par des virgules, comme ceci :

Code : CSS

```

h1, h2
{
  color: red;
}

```



Cela signifie : "Je veux que le texte de mes titres <h1> et <h2> soient écrits en rouge".

Vous pouvez indiquer autant de balises à la suite que vous le désirez. Par exemple, si vous voulez mettre en rouge vos titres, vos liens et vos citations, vous taperez :

h1, h2, a, q

C'est compris ? 😊

## Des commentaires dans du CSS

Comme en XHTML, il est possible de mettre des **commentaires**. Les commentaires ne seront pas affichés, ils servent simplement à indiquer des informations pour vous, par exemple pour vous y retrouver dans un looong fichier CSS.

D'ailleurs, vous allez vous en rendre compte, en général le fichier XHTML est assez petit, et la feuille CSS assez grande (si elle contient tous les éléments de style de votre site, c'est un peu normal). Il est possible (et recommandé) de créer plusieurs fichiers CSS : un fichier pour le design, un autre pour le forum, un autre pour le menu etc etc...

... De quoi on parlait déjà ? Ah oui, les commentaires en CSS (désolé je deviens un peu gâteux :p).

Donc, pour faire un commentaire, c'est fastoche : vous tapez */\**, suivi de votre commentaire, puis *\*/* pour terminer votre commentaire.

Vos commentaires peuvent être sur une ou plusieurs lignes. Par exemple :

### Code : CSS

```
/*
essai.css
-----

Par M@teo21
Fichier créé le 10/10/2004
*/

p
{
    color: blue; /* Les paragraphes seront bleus */
    font-size: 18px; /* La taille de 18 pixels me semble pas mal */
}
```

Il est probable que j'utilise pas mal de commentaires dans la suite du cours pour vous donner des explications à l'intérieur même des fichiers .css

## Utiliser les class et id

Ce que je vous ai montré jusqu'ici a quand même un défaut : cela implique par exemple que TOUS les paragraphes soient écrits par exemple en bleu / 18 pixels.

Comment faire pour que seulement certains paragraphes (ou certains titres, ou certaines citations) soient écrits d'une manière différente ?

On a la possibilité pour faire cela d'utiliser des attributs spéciaux qui fonctionnent sur toutes les balises :

- L'attribut *class*
- L'attribut *id*

Nous allons voir les choses suivantes dans l'ordre :

1. Ce que sont les class et id et comment s'en servir
2. Les balises dites "universelles" et leur utilité
3. Les imbrications de balises

Que de réjouissances en perspective !!! 😊

## Class et id

Que les choses soient claires dès le début : les attributs class et id sont quasiment identiques. Il y a seulement une petite différence que je vous dévoilerai plus bas.  
Pour le moment et pour faire simple, on ne va s'intéresser qu'à l'attribut *class*.

Comme je viens de vous le dire, c'est un attribut que l'on peut mettre sur n'importe quelle balise, aussi bien titre que paragraphe, image que etc...

```
<h1 class=""> </h1>
<p class=""></p>
<img class="" />
```

Oui mais que met-on comme valeur à l'attribut class ?

En fait, vous mettez un nom. Ce que vous voulez.

Les class vous permettent de **définir un style personnalisé**.

Supposons que vous vouliez définir un style appelé "important" qui écrive le texte en rouge / 18 pixels. Vous rajouterez l'attribut *class="important"* à chacune des balises que vous voulez modifier.

Sur cet exemple j'ai rajouté un attribut *class* à un paragraphe et à la citation :

### Code : HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
  <head>
    <title>Page d'exemple pour tester le CSS</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css" title="Essai 2" href="ess
  </head>
  <body>
    <h1>Découverte du CSS</h1>
    <p>
      Bonjour !<br />
      Je suis une page XHTML <em>apparemment</em> banale, mais je sers en fait de
    </p>
    <h2>Ce n'est que du blabla</h2>
    <p>
      Comme l'a dit Neil Armstrong un certain 20 juillet 1969 : <q class="importa
      J'aime la choucroute en conserve.<br />
      Vive les frites !
    </p>
    <p class="important">
      Quoi, ça ne veut rien dire ce que j'écris ? On s'en fout, c'est pour tester
    </p>
  </body>
</html>
```

Comment faire à partir de là pour dire dans le CSS : "Je veux que toutes mes balises qui ont la class "important" soient écrits en rouge / 18 pixels" ?

C'est très simple : au lieu de mettre le nom de la balise avant les accolades (comme *p* ou encore *h1*), vous devez écrire un point suivi du nom de la class, comme par exemple :

.important

Ce qui nous donnerait le CSS suivant :

### Code : CSS

```
.important
{
  color: red;
  font-size: 18px;
}
```

Testez maintenant la page XHTML précédente, et voyez comment la class "important" que nous avons créée modifie le texte :

[Essayer !](#)

Pratique, n'est-ce pas ? 😊

Et l'attribut *id* alors ?

Lui, il fonctionne exactement de la même manière que *class*, à un détail près : il ne peut être utilisé qu'une fois. Un nom d'id doit commencer par une lettre et peut être suivi ensuite de lettres et de chiffres. Un id nommé "**3tableaux**" est invalide car il commence par un chiffre. Un id "**tableau3**" est en revanche valide.

Quel intérêt ? Il y en a assez peu pour tout vous dire, cela vous sera utile si vous faites du Javascript plus tard pour reconnaître certaines balises. D'ailleurs, nous avons déjà vu l'attribut *id* dans le chapitre sur les liens (pour réaliser des ancres).

En pratique, nous ne mettrons des "id" que sur des éléments qui sont uniques sur votre page, comme par exemple le logo :

```

```

Dans le CSS, ce n'est pas un point que vous devez mettre avant le nom de l'id, mais un dièse (#) :

Code : CSS

```
#logo
{
    /* Mettez les propriétés CSS ici */
}
```

Je ne vous propose pas de le tester, parce que de toute façon ça fonctionne exactement comme *class*.

Si vous vous emmêlez les pinceaux entre *class* et *id* (il ne devrait pas y avoir de raison, mais on sait jamais), vous pouvez utiliser tout le temps des *class* car ça marchera toujours.

Pour ma part, il se peut que j'utilise des *id* dans la suite de ce cours. J'ai pris l'habitude de mettre un *id* plutôt qu'une *class* quand je sais que je ne m'en servirai qu'une fois.

## Les balises universelles

Il arrivera parfois que vous ayez besoin d'appliquer une *class* (ou un *id* mais c'est plus rare) à certains mots qui ne sont pas entourés par des balises.

En effet, le problème des *class* c'est qu'il s'agit d'un attribut, vous ne pouvez donc en mettre que sur une balise.

Par exemple, si je veux modifier uniquement "Neil Armstrong" dans le paragraphe suivant :

Code : HTML

```
<p>Comme l'a dit Neil Armstrong un certain 20 juillet 1969...</p>
```

Ca serait facile à faire s'il y avait une balise autour de "Neil Armstrong", malheureusement il n'y en a pas. Heureusement, on a inventé la balise-qui-sert-à-rien 😊

En fait, on a inventé 2 balises qui ne servent à rien, avec une petite (mais importante !) différence entre les deux :

- **<span> </span>** : c'est une balise de type *inline*. Vous vous souvenez ce qu'est une balise inline ? C'est une balise qui se met à l'intérieur d'un paragraphe, comme <strong>, <em>, <q>. Cette balise s'utilise donc au milieu d'un paragraphe, et c'est celle dont nous allons nous servir pour colorer Neil Armstrong.
- **<div> </div>** : c'est une balise de type *block*. Comme <p>, <h1> etc... Elle crée un nouveau "bloc" dans la page, et provoque donc obligatoirement un retour à la ligne. C'est une balise très très utilisée pour faire un design. Nous lui dédions d'ailleurs 2 chapitres entiers dans la partie III du cours pour vous aider à construire le design de votre site web.

Pour le moment donc, nous allons utiliser plutôt la balise <span>. On la met autour de Neil Armstrong, on lui rajoute une class, on crée le CSS et puis basta 😊

Code : HTML

```
<p>Comme l'a dit <span class="nom">Neil Armstrong</span> un certain 20 juillet 1969...</p>
```

Code : CSS

```
.nom
{
    color: blue;
}
```

[Essayer !](#)

Vous me verrez probablement réutiliser des `<span>` dans la suite du cours, parce qu'il faut avouer que c'est bien pratique 😊

## Imbrications de balises

Une des dernières notions "de base" que je souhaite vous enseigner sur le CSS est qu'on a la possibilité d'indiquer l'ordre d'imbrication des balises dans le CSS.

On pourrait penser que c'est encore un truc tordu inventé par des informaticiens portant des grosses lunettes... mais que nenni !



C'est en fait simple à comprendre, et surtout utile.

Nous sommes dans notre fichier CSS. Nous souhaiterions définir un style uniquement pour les balises `<em>` qui se trouvent à l'intérieur d'un titre `<h3>`. On doit écrire ceci :

### Code : CSS

```
h3 em /* => tous les em situés à l'intérieur d'un h3 */
{
    color: blue;
}
```

Comme vous le voyez, on a juste séparé les 2 noms de balises par un espace, et non pas par une virgule comme on l'a fait plus haut.

Cela signifie "Je veux modifier le style de toutes les balises `<em>` situées à l'intérieur de titres `<h3>`". On a indiqué dans le CSS dans quel ordre devaient être imbriquées les balises.

L'ordre est **très important** ! Si vous aviez mis "em h3" cela aurait voulu dire "Tous les `<h3>` situés à l'intérieur de balises `<em>`". Cela est d'ailleurs impossible à faire car on ne peut pas mettre une balise de type block dans une balise de type inline.

Voici un fichier XHTML pour que vous compreniez bien comment ça fonctionne :

### Code : HTML

```
<h3>L'imbrication, c'est <em>pratique</em></h3>
```

```
<em>Get exemple vous montre comment fonctionne l'imbrication <em> /> Le mot "pratique" der
```

[Essayer !](#)

Pratique, n'est-ce pas ? 😊

Vous pouvez aussi mélanger des balises et des class :

```
p .important
```

Ce qui signifie : "Toutes les class "important" contenues dans des balises `<p>`".

Un petit exemple tordu pour terminer en beauté :

```
blockquote p strong, h1 .important
```

Alors là, attention de ne pas tout mélanger. La virgule "coupe" la ligne en deux, elle signifie "ET". Pour bien voir que la ligne est coupée en deux, je vais colorer les deux parties :

```
blockquote p strong, h1 .important
```

Ce qui veut dire : "Toutes les balises `<strong>` contenues dans un `<p>` elles-mêmes contenues dans un `<blockquote>` ET toutes les class "important" contenues dans un titre `<h1>`"

## Q.C.M.

Laquelle de ces méthodes est la plus recommandée pour mettre du CSS ?

- ☐ Tapez le CSS entre les balises <head></head> à l'aide de la balise <style>
- ☐ Tapez le CSS dans un fichier .css
- ☐ Tapez le CSS dans un attribut "style" ajouté aux balises

Quels attributs de la balise <link /> a-t-on besoin de modifier pour le moment ?

- ☐ rel et type
- ☐ href et rel
- ☐ title et href

Combien de propriétés peut-on appliquer à une balise ?

- ☐ De 1 à 10
- ☐ De 0 à 10
- ☐ De 1 à... autant qu'on veut

Que signifie ce bout de CSS ?

Code : CSS

```
strong
{
  color: red;
}
```

- ☐ Mettre tous les textes considérés comme importants en rouge
- ☐ Agrandir tous les textes considérés comme importants
- ☐ Mettre en gras tous les textes en rouge

Comment faire pour demander à ce qu'un style s'applique à tous les titres ?

- ☐ h1, h6
- ☐ h1, h2, h3, h4, h5, h6
- ☐ h1 h2 h3 h4 h5 h6

Lequel de ces textes n'est ni un commentaire CSS, ni un commentaire XHTML ?

- ☐ /\* Je suis un commentaire \*/
- ☐ <!-- Non c'est moi le commentaire ! -->
- ☐ <! Vous vous méprenez, c'est moi ! -->

Quelle est la particularité des id par rapport aux class ?

- ☐ Ils ne peuvent être mis que sur des balises de titre
- ☐ On ne peut donner un nom d'id qu'une seule fois, tandis qu'on peut répéter un nom de class plusieurs fois
- ☐ Les id ne fonctionnent que sur des balises de type block, tandis que class fonctionne sur des balises de type inline

Laquelle de ces balises n'est pas une balise universelle ?

- ☐ <block>
- ☐ <div>

- ☐ <span>

A qui va s'adresser le style suivant ?

Code : CSS

```
blockquote .gros, h1, h2 .gros
{
  font-size : 20px;
}
```

- ☐ Aux class "gros" situées à l'intérieur de blockquote, de h1 et de h2
- ☐ Aux blockquote, h1 et h2 ainsi qu'aux class "gros"
- ☐ Aux class "gros" situées à l'intérieur de blockquote et de h2, et à tous les titres h1

Correction !

[Statistiques de réponses au QCM](#)

---

Que de nouveautés, que de nouveautés...

Eh oui, le CSS est un langage "à part", les règles ne sont pas les mêmes. Mais comme vous pouvez le voir, XHTML et CSS sont très liés, et on peut difficilement concevoir l'un sans l'autre 😊

Toutes les nouvelles notions que je vous ai apprises dans ce chapitre sont importantes, aussi prenez votre temps pour bien les assimiler. Rien ne presse, personne ne vous oblige à passer au chapitre suivant (à moins que vous ayez fait un pari stupide avec des amis du genre "Tu n'arriveras pas à apprendre à créer un site web en 2 jours" :p)

Si tant de nouveautés d'un seul coup vous font un peu peur, ne vous affolez pas pour autant : le plus dur c'est toujours d'apprendre "les bases". Or, c'est justement ce qu'on vient de faire. Tout le reste devrait couler de source à partir de maintenant 😊

Dans le prochain chapitre, nous allons découvrir un nombre important de propriétés CSS. Car pour le moment, à part écrire notre texte en bleu avec une taille de 18 pixels, on ne sait rien faire de bien palpitant 😊