



# Formatage du texte en CSS (partie 1/2)

---

## Informations sur le tutoriel

[Ajouter à mes tutoriels favoris](#) (6 fans)



Auteur : [M@teo21](#)

Licence :

[Plus d'informations](#)

## Popularité

Visualisations : 336 744 528

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 arrivons maintenant à un chapitre qui devrait beaucoup vous intéresser 😊

Non, le "formatage du texte" n'a rien à voir avec la destruction de toutes les données présentes sur votre disque dur 😊 Cela signifie simplement que l'on va modifier l'apparence du texte (on dit qu'on le "met en forme").

Soyez rassurés donc, à la fin de ce chapitre votre disque dur sera toujours vivant 😊

Pas de surprise particulière : nous sommes toujours dans le CSS, et nous allons réutiliser ce que nous venons d'apprendre dans le chapitre précédent. J'espère donc, ceci dit en passant, que vous avez bien lu et compris comment on se servait d'un .css 😊

Cette fois, on passe au concret : nous allons voir successivement comment modifier la taille du texte, changer la police, aligner le texte etc... Vous avez une foule de choses à découvrir, et afin de ne pas vous assommer d'un chapitre trop gros j'ai décidé de le scinder en 2 parties.

La bonne nouvelle, c'est qu'à la fin de ces 2 chapitres votre site commencera (enfin) à ressembler à quelque chose 😊

Sommaire du chapitre :



- [Taille du texte](#)
- [Polices](#)

- [Alignement](#)
- [Q.C.M.](#)

## Taille du texte

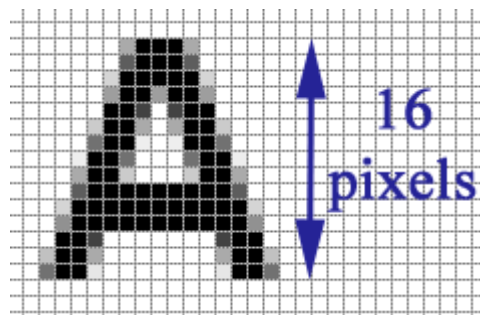
Hé, mais modifier la taille du texte je sais déjà le faire ! Tu l'as expliqué dans le chapitre précédent !

En fait, dans le chapitre précédent je vous ai donné un exemple sans trop d'explications, afin d'illustrer un peu le fonctionnement d'un CSS. En réalité, les possibilités pour modifier la taille du texte sont nombreuses (et variées !).

Ce que vous savez déjà, c'est que la propriété CSS qui permet de changer la taille du texte est *font-size*. Ca, ça ne change pas. Mais par contre, on peut indiquer la taille du texte de différentes manières :

- **En pixels** : c'est une façon très précise d'indiquer la taille du texte. C'est à vous de dire combien de pixels exactement doit faire le texte.  
Pour avoir un texte de 16 pixels de hauteur, vous devez écrire :  
`font-size:16px;`

Les lettres auront une taille de 16 pixels, comme le montre l'image suivante :



Voici un exemple d'utilisation (je ne vous mets pas le fichier XHTML qui va avec, vous êtes des grands maintenant 😊) :

Code : CSS

```
p
{
    font-size: 14px; /* Paragraphes de 14 pixels */
}
h1
{
    font-size: 22px; /* Titres de 22 pixels */
}
```

[Essayer !](#)

Si vous êtes allergique aux pixels, sachez qu'il est aussi possible d'indiquer une taille en centimètres ("2cm" par exemple) et en millimètres ("14mm" par exemple).

Indiquer la taille du texte en pixels, centimètres et millimètres est très pratique et très précis. Certes. Mais il est préférable d'indiquer une taille relative (comme nous allons le voir) afin que la taille du texte s'adapte aux choix de tout le monde (certains préfèrent avoir du texte plutôt gros, d'autres plutôt petit etc...)

- **En donnant une valeur relative** : c'est-à-dire en écrivant carrément "gros", "très gros", "petit", "minuscule". Mais bien sûr, comme tout c'est en anglais :p. Voici la liste des différentes valeurs que vous pouvez mettre ainsi que leur signification :

- xx-small : minuscule
- x-small : très petit
- small : petit
- medium : moyen
- large : grand
- x-large : très grand
- xx-large : euh... gigantesque 😊

Voici un exemple de CSS qui utilise des valeurs relatives :

#### Code : CSS

```
.minuscule
{
    font-size: xx-small;
}
.trespetit
{
    font-size: x-small;
}
.petit
{
    font-size: small;
}
.moyen
{
    font-size: medium;
}
.grand
{
    font-size: large;
}
.tresgrand
{
    font-size: x-large;
}
.supermegagigatresgrand
{
    font-size: xx-large;
}
```

[Essayer !](#)

La taille du texte n'est peut-être pas exactement identique selon le navigateur... mais ce n'est pas bien grave, car dans tous les cas "xx-small" correspond à un texte très petit, et "xx-large" à un texte très grand. Et c'est tout ce qu'on a besoin de savoir 😊

Je vous encourage à utiliser cette méthode pour indiquer la taille du texte plutôt que d'utiliser des pixels. Votre site sera ainsi bien plus "adaptable" aux différentes configurations de vos visiteurs.

- **En em** : c'est une autre façon d'indiquer de manière relative la taille du texte. Un peu délicat à comprendre, je vous l'accorde, mais une fois qu'on a un peu testé ça vient tout seul, et on se rend compte que c'est une méthode vraiment pratique.  
On doit indiquer la taille du texte par rapport à la taille normale de la police. Je m'explique : "1em" signifie "Taille normale". Si vous mettez un nombre supérieur (généralement un nombre décimal) comme "1.3em", le texte aura une taille 1,3 fois plus grande. De même pour réduire : "0.8em" et votre texte aura une taille 0,8 fois plus petite.  
C'est personnellement la méthode que j'utilise le plus souvent pour indiquer la taille du texte (la plupart de mes *font-size* sont donc en "em").

Faites attention, il faut mettre un point à la place de la virgule pour les nombres décimaux. Vous devez donc écrire "1.4em" et non pas "1,4em" !

**Code : CSS**

```
.petit_em
{
    font-size: 0.7em;
}
.grand_em
{
    font-size: 1.3em;
}
```

[Essayer !](#)

- **En pourcentage** : ça c'est facile. Si vous indiquez "100%", le texte aura une taille "normale". Si vous mettez "130%", le texte aura une taille correspondant à 130% de la taille normale. Ça ressemble énormément aux "em" (en fait c'est plus ou moins pareil). Après, tout est une question de goût 😊

Pfiou ! Comme quoi, il y a l'embarras du choix pour indiquer la taille du texte 😊  
A tel point que l'on s'y perd un peu d'ailleurs...

Comme je vous l'ai dit, les pixels c'est très pratique pour être précis, mais ce n'est pas très "recommandé" car il se peut qu'une taille trop petite (ou trop grosse) gêne certaines personnes.  
En ce qui me concerne, la méthode des "em" (ou des %) est la plus pratique : elle a l'avantage de s'adapter automatiquement aux préférences du visiteur et d'être facile à utiliser.

---

## Polices

Ah... La police... On touche un point sensible 😊

En effet, le problème c'est que, pour qu'une police s'affiche correctement, il faut que tous les internautes l'aient. Si un internaute n'a pas la même police que vous, son navigateur prendra une police par défaut (une police standard) qui n'aura peut-être rien à voir avec ce à quoi vous vous attendiez.

Certes, tout le monde a "Times new roman" et "Arial" me direz-vous... Et encore, sur Mac c'est "Time" et pas "Times New Roman".

Mais si vous aviez envisagé d'utiliser la dernière police gothique à la mode pour votre site de Heavy Metal, il va peut-être falloir revoir vos plans 😊

Bon, et concrètement comment on fait ?

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

```
font-family: police;
```

Seulement, pour éviter qu'il n'y ait de problème si l'internaute n'a pas la même police que vous, on met en général **plusieurs** noms de police, séparés par des virgules :

```
font-family: police1, police2, police3, police4;
```

Le navigateur essaiera d'abord de mettre la police1. S'il ne l'a pas, il essaiera de mettre la police2. S'il ne l'a pas, il essaiera la police3 et ainsi de suite.

En général, on met en tout dernier "serif", ce qui correspond à une police standard (qui ne se met que si aucune autre police n'a été trouvée).

Oui, mais quelles sont les polices les plus courantes qu'on a le "droit" d'utiliser me direz-vous ?

Voici une liste de polices qui fonctionnent bien sur la plupart des navigateurs et que vous pouvez donc utiliser sans crainte :

- Arial

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

Si vous désirez tester les différentes polices listées ci-dessus, cliquez sur le bouton ci-dessous :

[Essayer !](#)

Ainsi, si j'écris :

```
font-family: Impact, "Arial Black", Arial, Verdana, serif;
```

... cela signifie : *"Mets la police Impact, ou, si elle n'y est pas, Arial Black, ou sinon Arial, ou sinon Verdana, ou si rien n'a marché mets une police standard (serif)"*

En général, il est bien d'indiquer un choix de 3-4 polices (+ serif) afin de s'assurer qu'au moins l'une d'entre elles aura été trouvée sur l'ordinateur du visiteur.

Si le nom de la police comporte des espaces, vous devez l'entourer de guillemets, comme je l'ai fait pour "Arial Black".

Allez, on se fait un petit CSS d'exemple pour récapituler tout ça et ça sera bon :

Code : CSS

```
h1
{
    font-family: "Arial Black", Arial, Verdana, serif; /* On essaie d'avoir
}
p
{
    /* La police Comic Sans MS est agréable à lire pour les paragraphes je t
    font-family: "Comic Sans MS", "Trebuchet MS", Georgia, serif;
```

[Essayer !](#)

---

## Alignement

### Alignements simples

Le langage CSS nous permet de faire tous les alignements que l'on connaît : 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 tout le temps justifiés.

Regardez les différents alignements sur cet exemple :

## Code : CSS

```

h1
{
    text-align: center; /* Pour centrer le titre */
    font-family: "Arial Black", Arial, Verdana, serif; /* Un titre en Arial
}
blockquote
{
    text-align: justify; /* La citation sera justifiée */
}
.signature
{
    text-align: right; /* Pour aligner à droite ma signature */
    font-family: "Comic Sans MS", Georgia, "Times New Roman", serif;
    font-size: 80%;
}

```

N'oubliez pas qu'il y a toujours du XHTML derrière, bien entendu. D'ailleurs pour cet exemple, je vais vous montrer le code que j'utilise. Sachez toutefois qu'il est toujours possible d'afficher le code XHTML d'une page en faisant [Clic droit / "Afficher le code source"](#).

Voici donc le code XHTML de cet exemple permettant de tester les alignements :

## Code : HTML

```

<h1>La parole du sage</h1>

<p>Un jour, un grand sage a dit :</p>

<blockquote><p>Morbi fermentum libero non libero. Nunc in turpis in justo a

<div class="signature">Signé : Moteo21</div>

```

[Essayer !](#)

Vous ne pouvez pas modifier l'alignement du texte d'une balise *inline* (comme span, a, em, strong...). L'alignement ne fonctionne que sur des balises de type *block* (p, div, blockquote, h1, h2, ...), et c'est un peu logique quand on y pense : on ne peut pas modifier l'alignement de quelques mots au milieu d'un paragraphe ! C'est donc en général le paragraphe entier qu'il vous faudra aligner.

## L'indentation

Qu'est-ce que l'indentation ? C'est tout simplement la mise en retrait du texte. Cela permet par exemple de faire commencer un paragraphe un peu plus à droite, ce qui rend (je trouve) la lecture beaucoup plus facile et agréable. C'est un procédé que l'on retrouve dans la plupart des livres d'ailleurs.

Voici un exemple de paragraphe indenté :

Bien que cela :  
 plus en plus inquié  
 telles hostilités ven  
*Texte indenté*

On utilise la propriété *text-indent*. On doit indiquer quelle est la taille du retrait : on peut le faire en pixels, en centimètres, en millimètres...

Bref, là le mieux à mon avis c'est d'utiliser les pixels ^^ . Regardez comment je mets tous les textes des

paragraphes en retrait avec *text-indent* :

Code : CSS

```
P
{
    text-indent: 30px; /* Les paragraphes commenceront 30 pixels sur la droite */
    text-align: justify; /* Ils seront justifiés */
    font-size: large; /* Allez, soyons fous, grossissons le texte :o;) */
}
```

[Essayer !](#)

L'avantage en CSS, c'est qu'il vous suffit de dire une fois pour toutes : *"Je veux que tous mes paragraphes soient indentés de 30 pixels sur la droite"* et à chaque fois que vous écrirez un nouveau paragraphe, celui-ci sera automatiquement mis en retrait.

Avant le XHTML (à l'époque du HTML) ce genre de chose était impossible, ou plutôt difficile et répétitif. Là, c'est un jeu d'enfant d'aligner son texte comme on veut 😊

A noter 2 propriétés similaires sur lesquelles je n'ai pas besoin de m'étendre :

- word-spacing : l'espace entre les mots (en pixels).
- letter-spacing : l'espace entre les lettres (en pixels).

## Q.C.M.

Laquelle de ces unités ne permet pas d'indiquer la taille du texte ?

- ☐ em
- ☐ px
- ☐ km

Que signifie ce bout de CSS ?

Code : CSS

```
font-family: Verdana, "Arial Black", Arial, serif;
```

- ☐ Mets les polices Verdana, Arial Black, Arial et serif à ce texte.
- ☐ Essaie de mettre la police Verdana si le visiteur la possède, ou sinon Arial Black, ou sinon Arial, ou sinon une police standard.
- ☐ Mets la police serif si l'internaute la possède, ou sinon Arial, ou sinon Arial Black, ou enfin Verdana.

Laquelle de ces valeurs ne peut pas être appliquée à la propriété text-align ?

- ☐ central
- ☐ left
- ☐ justify

Comment faire pour que mes titres h3 soient décalés de 20 pixels sur la droite ?

- ☐ Utiliser text-indent et mettre la valeur 20px
- ☐ Taper plein de fois frénétiquement sur la touche espace pour le décaler
- ☐ Utiliser text-left et mettre la valeur 20px

Laquelle de ces méthodes est à éviter si possible pour indiquer la taille du texte ?

- ☐ Les px
- ☐ Les em
- ☐ Les %

Quel est le problème dans ce bout de code CSS ?

Code : CSS

```
font-family:Verdana, Trebuchet MS, Georgia, Impact, Arial, serif;
```

- ☐ Il y a trop de polices dans la liste
- ☐ Il manque des guillemets autour de Trebuchet MS
- ☐ Il manque une police standard en fin de liste

Correction !

[Statistiques de réponses au QCM](#)

---

Et voilà une moitié du morceau d'avalée 😊

Dès que vous vous sentez en forme, rendez-vous au chapitre suivant pour découvrir encore tout plein de propriétés CSS de formatage du texte 😊