

AP1 : prise en main d'un site WEB existant, ajout de contenu et développement personnel

B1.3 Développer la présence en ligne

B1.6 Organiser son développement personnel

Objet du document : Apprentissage des éléments principaux de HTML 5 et CSS 3 afin de pouvoir réaliser un site web

Prérequis : Editeur de texte Notepad++ ou VS code Navigateur Firefox et Chrome

Chapitre 9 : CSS Formatez du Texte

La taille

Pour modifier la taille du texte, on utilise la propriété CSS `font-size`. Mais comment indiquer la taille du texte ? C'est là que les choses se corsent, car plusieurs techniques vous sont proposées :

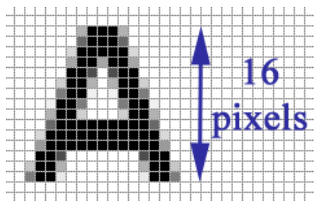
- indiquer une **taille absolue** : en pixels, en centimètres ou millimètres. Cette méthode est très précise, mais il est conseillé de ne l'utiliser que si c'est absolument nécessaire, car on risque d'indiquer une taille trop petite pour certains lecteurs ;
- indiquer une **taille relative** : en pourcentage,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

Une taille absolue

Pour indiquer une taille absolue, on utilise généralement les pixels. Pour avoir un texte de 16 pixels de hauteur, vous devez donc écrire :

```
font-size: 16px;
```

Les lettres auront une taille de 16 pixels, comme le montre la figure suivante.



Une lettre de 16 pixels de hauteur

Voici un exemple d'utilisation (placez ce code dans votre fichier `.css`) :

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

```

}
h1
{
    font-size: 40px; /* Titres de 40 pixels */
}

```

Et le résultat est visible à la figure suivante.



Différentes tailles de texte

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 ceux-ci :

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

Vous pouvez tester l'utilisation de ces valeurs dans votre code CSS :

```
p
{
    font-size: small;
}
h1
{
    font-size: large;
}
```

Bon, cette technique a un défaut : il n'y a que sept tailles disponibles (car il n'y a que sept noms). Heureusement, il existe d'autres moyens. Ma technique préférée consiste à indiquer la taille en « em ».

- Si vous écrivez `1em` , le texte a une taille normale.
- Si vous voulez grossir le texte, vous pouvez inscrire une valeur supérieure à 1, comme `1.3em` .
- Si vous voulez réduire le texte, inscrivez une valeur inférieure à 1, comme `0.8em` .

Faites attention : pour les nombres décimaux, il faut mettre un point et non une virgule. Vous devez donc écrire « `1.4em` » et non pas « `1,4em` » !

Exemple :

```
p
{
    font-size: 0.8em;
}
h1
{
    font-size: 1.3em;
}
```

D'autres unités sont disponibles. Vous pouvez essayer le « ex » (qui fonctionne sur le même principe que le em, mais qui est plus petit de base) et le pourcentage (80 %, 130 %...).

La police

Ah... la police... On touche un point sensible.

En effet, il se pose un problème : 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.

La bonne nouvelle, c'est que depuis CSS 3, il est possible de faire télécharger automatiquement une police au navigateur. Je vous expliquerai dans un second temps comment faire cela.

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 :

```
balise
{
    font-family: nom_de_la_police;
}
```

Seulement, pour éviter les problèmes si l'internaute n'a pas la même police que vous, on précise en général *plusieurs* noms de police, séparés par des virgules :

```
balise
{
    font-family: police1, police2, police3, police4;
}
```

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

En général, on indique en tout dernier `serif` , ce qui correspond à une police par défaut (qui ne s'applique que si aucune autre police n'a été trouvée).

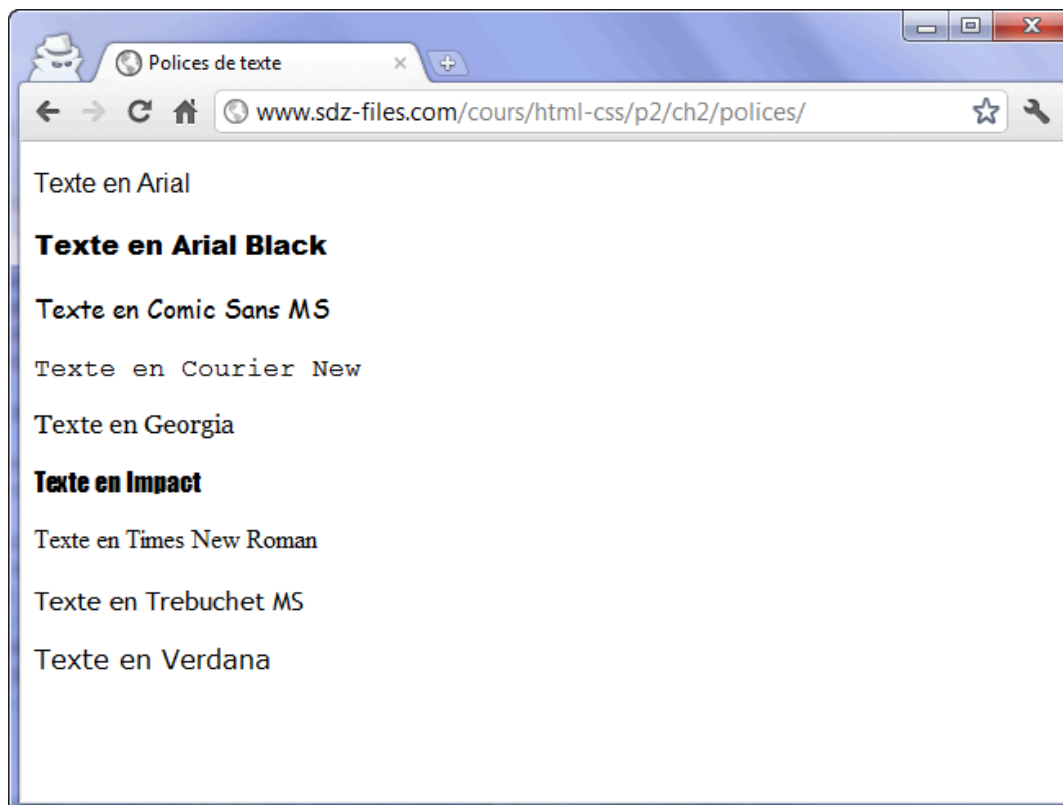
Il existe aussi une autre police par défaut appelée `sans-serif` . La différence entre les deux est la présence de petites pattes de liaison en bas des lettres, que la police `sans-serif` n'a pas. Oui, c'est subtil.

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 :

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

La figure suivante vous montre à quoi ressemblent ces polices.



Différentes polices

Ainsi, si j'écris :

```
p
{
    font-family: Impact, "Arial Black", Arial, Verdana, sans-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 (Sans Serif) ».

En général, il est bien d'indiquer un choix de trois ou quatre polices (+ Serif ou Sans 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, je conseille de l'entourer de guillemets, comme je l'ai fait pour « Arial Black ».

Mettre en italique

Attends un peu là ! Je croyais que la balise `` permettait de mettre un texte en italique ?!

Je n'ai jamais dit cela.

Retournez voir les chapitres précédents si vous avez des doutes, mais je n'ai *jamais* dit que la balise `` était faite pour mettre le texte en italique (de même que je n'ai jamais dit que `` était faite pour mettre en gras).

`` , mettez-vous bien cela dans la tête, est faite pour insister sur des mots. Cela veut dire que les mots qu'elle entoure sont assez importants.

Pour représenter cette importance, la plupart des navigateurs choisissent d'afficher le texte en italique, mais ce n'est pas une obligation.

Le CSS lui, permet de dire réellement : « Je veux que ce texte soit en italique ». Rien ne vous empêche, par exemple, de décider que tous vos titres seront en italique.

Concrètement, en CSS, pour mettre en italique, on utilise `font-style` qui peut prendre trois valeurs :

- `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. Par exemple, si vous voulez que les textes entre `` ne soient plus en italique, vous devrez écrire :

```
em
{
    font-style: normal;
}
```

Ainsi, dans l'exemple suivant, je me sers de `font-style` pour mettre en italique tous mes titres `<h2>` :

```
h2
{
    font-style: italic;
}
```

Mettre en gras

Et si nous passions à la mise en gras ?

Alors, là encore, n'oubliez pas que ce n'est pas `` qui permet de mettre en gras (son rôle est d'indiquer que le texte est important, *donc* le navigateur l'affiche généralement en gras). La mise en gras en CSS peut par exemple s'appliquer aux titres, à certains paragraphes entiers, etc. C'est à vous de voir.

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 (par défaut).

Voici par exemple comment écrire les titres en gras :

```
h1
{
    font-weight: bold;
}
```

Soulignement et autres décorations

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 ;
- `none` : normal (par défaut).

Ce CSS va vous permettre de tester les effets de `text-decoration` :

```
h1
{
    text-decoration: blink;
}
.souligne
{
    text-decoration: underline;
}
.barre
{
    text-decoration: line-through;
}
.ligne_dessus
{
    text-decoration: overline;
}
```

Essayez vous-même pour voir le résultat !

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.

Regardez les différents alignements sur cet exemple :

```
h1
{
    text-align: center;
}

p
{
    text-align: justify;
}

.signature
{
    text-align: right;
}
```

Le résultat est visible sur la figure suivante.



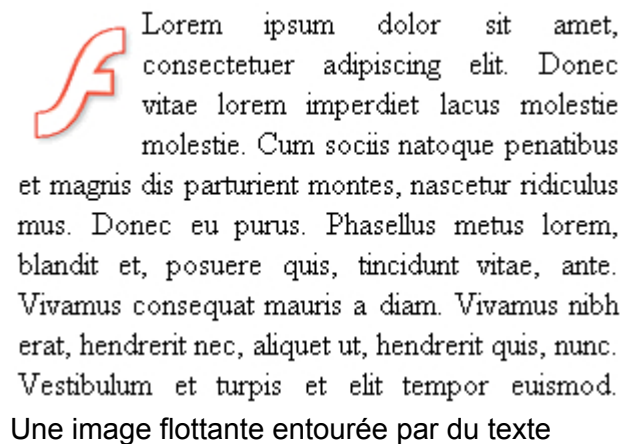
Alignements du texte

Vous ne pouvez pas modifier l'alignement du texte d'une balise *inline* (comme `` , `<a>` , `` , `` ...). L'alignement ne fonctionne que sur des balises de type *block* (`<p>` , `<div>` , `<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.

Les flottants

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

Pour que vous voyiez bien de quoi on parle, la figure suivante vous montre ce que nous allons apprendre à faire.



J'imagine que, maintenant, la question qui vous brûle les lèvres est : « *Mais quelle est donc la propriété magique qui fait flotter ?* ».

La réponse est... `float` (« flottant » en anglais). Cette propriété peut prendre deux valeurs très simples :

- `left` : l'élément flottera à gauche ;
- `right` : l'élément flottera... à droite ! Oui, bravo.

L'utilisation des flottants est très simple :

1. Vous appliquez un `float` à une balise.
2. Puis vous continuez à écrire du texte à la suite normalement.

On peut aussi bien utiliser la propriété `float` sur des balises *block* que sur des balises *inline*. Il est courant de faire flotter une image pour qu'elle soit habillée par du texte, comme dans l'exemple précédent.

Faire flotter une image

Nous allons apprendre ici à faire flotter une image. Voici le code HTML que nous devons taper dans un premier temps :

```
<p>
Ceci est un texte normal de paragraphe, écrit à la suite de l'image et qui
l'habillera car l'image est flottante.</p>
```

Vous devez placer l'élément flottant en premier dans le code HTML. Si vous placez l'image après le paragraphe, l'effet ne fonctionnera pas.

Voici le seul bout de code CSS qu'on ait besoin de taper, qui permet de faire flotter l'image à gauche :

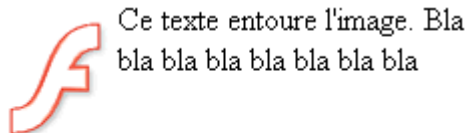
```
.imageflottante
{
    float: left;
}
```

Amusez-vous aussi à faire flotter l'image à droite, c'est tout bête : il suffit d'indiquer la valeur `right` , et le tour est joué !

Stopper un flottant

Quand vous mettez en place un flottant, le texte autour l'habille. Mais comment faire si vous voulez qu'au bout d'un moment le texte continue *en dessous* du flottant ? On pourrait enchaîner plusieurs `
` à la suite, mais cela ne serait ni élégant ni très propre...

En gros, on aimerait pouvoir obtenir le même résultat qu'à la figure suivante.



Texte écrit en-dessous

Le texte sous l'image ignore la propriété float

Il existe en fait 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;` .

Pour simplifier, on va utiliser tout le temps le `clear: both` , qui marche après un flottant à gauche et après un flottant à droite (cela fonctionne donc à tous les coups). Pour illustrer son fonctionnement, on va prendre ce code HTML :

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

Et ce code CSS :

```
.imageflottante
{
    float: left;
}
.dessous
{
    clear: both;
}
```

On applique un `clear: both;` au paragraphe que l'on veut voir continuer sous l'image flottante, et le tour est joué !

EN RÉSUMÉ : NOTEZ ICI LES POINTS IMPORTANTS DU CHAPITRE