**Formater du texte**

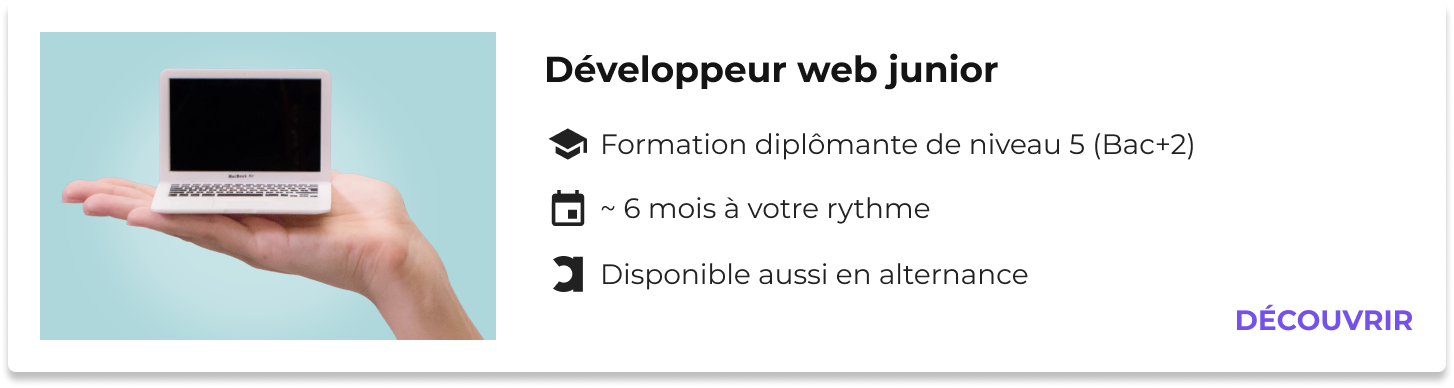
Nous arrivons maintenant à un chapitre qui devrait beaucoup vous intéresser.

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 »).

Pas de surprise particulière : nous sommes toujours dans le CSS, et nous allons réutiliser ce que nous venons d'apprendre dans le chapitre précédent. Nous allons donc travailler directement au sein du fichier .css  que nous avons créé.

Ce chapitre va être l'occasion de découvrir de nombreuses propriétés CSS : nous allons voir comment modifier la taille du texte, changer la police, aligner le texte…

Pour valider vos compétences avec un diplôme bac+2, suivez la [**formation Développeur web**](https://openclassrooms.com/fr/paths/185-developpeur-web) avec OpenClassrooms. 6 mois en moyenne sont nécessaires pour finaliser cette [**formation à distance**](https://openclassrooms.com/fr/), qui peut aussi être suivie en [**alternance ou contrat d’apprentissage**](https://openclassrooms.com/fr/apprenticeship).

[](https://openclassrooms.com/fr/paths/185-developpeur-web)

**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 :

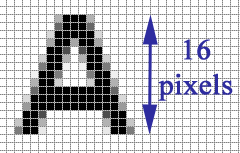
* indiquer une **taille absolue** : en pixels, en centimètres ou millimètres. Cette méthode est très précise, mais il est conseillé de ne l'utiliser que si c'est absolument nécessaire, car on risque d'indiquer une taille trop petite pour certains lecteurs ;
* indiquer une **taille relative** : en pourcentage, « 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

Si vous le souhaitez, vous pouvez également définir des tailles en centimètres ou millimètres. Remplacez « px » par « cm » ou « mm ». Ces unités sont cependant moins bien adaptées aux écrans.

**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*: 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;

}

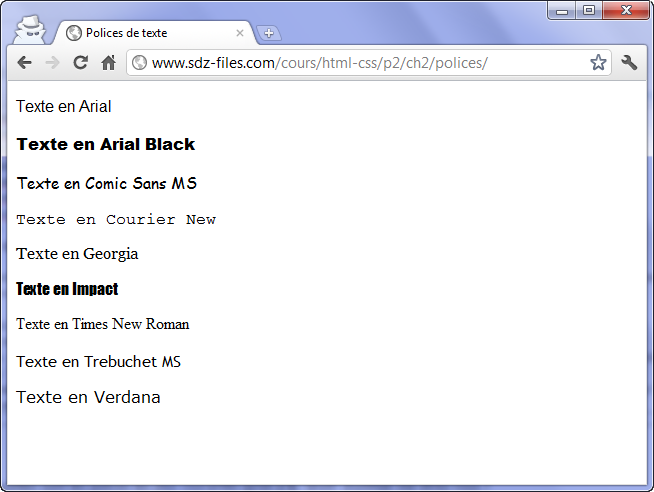
Le navigateur essaiera d'abord d'utiliser la police1  . S'il ne l'a pas, il essaiera la police2  . S'il ne l'a pas, il passera à la police3  , et ainsi de suite.  
En général, on indique en tout dernier serif  , ce qui correspond à une police par défaut (qui ne s'applique que si aucune autre police n'a 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 ».

**Utiliser une police personnalisée avec @font-face**

Je trouve le choix des polices trop limité.  
Comment puis-je utiliser ma police préférée sur mon site web ?

Pendant longtemps, cela n'était pas possible. Aujourd'hui, avec CSS 3, il existe heureusement un moyen d'utiliser n'importe quelle police sur son site. Cela fonctionne bien avec la plupart des navigateurs.

Mais attention, il y a des défauts (ce serait trop beau, sinon) :

* il faudra que le navigateur de vos visiteurs *télécharge* automatiquement le fichier de la police, dont le poids peut atteindre, voire dépasser 1 Mo… ;
* la plupart des polices sont soumises au droit d'auteur, il n'est donc *pas légal* de les utiliser sur son site. Heureusement, il existe des sites comme [fontsquirrel.com](http://www.fontsquirrel.com/) et [dafont.com](https://www.dafont.com/fr/) qui proposent en téléchargement un certain nombre de polices libres de droits. Je recommande en particulier [fontsquirrel.com](http://www.fontsquirrel.com/), car il permet de télécharger des packs prêts à l'emploi pour CSS 3 ;
* il existe *plusieurs formats* de fichiers de polices, et ceux-ci ne fonctionnent pas sur tous les navigateurs.

Voici les différents formats de fichiers de polices qui existent et qu'il faut connaître :

* .ttf  : *TrueType Font*. Fonctionne sur IE9 et tous les autres navigateurs ;
* .eot  : *Embedded OpenType*. Fonctionne sur Internet Explorer uniquement, toutes versions. Ce format est propriétaire, produit par Microsoft ;
* .otf  : *OpenType Font*. Ne fonctionne pas sur Internet Explorer ;
* .svg  : *SVG Font*. Le seul format reconnu sur les iPhone et iPad pour le moment ;
* .woff  : *Web Open Font Format*. Nouveau format conçu pour le Web, qui fonctionne sur IE9 et tous les autres navigateurs.

En CSS, pour définir une nouvelle police, vous devez la déclarer comme ceci :

@font-face {

font-family: 'MaSuperPolice';

src: url('MaSuperPolice.eot');

}

Le fichier de police (ici *MaSuperPolice.eot*) doit ici être situé dans le même dossier que le fichier CSS (ou dans un sous-dossier, si vous utilisez un chemin relatif).

Je croyais qu'il y avait plusieurs formats de police ?

Oui, d'ailleurs les .eot ne fonctionnent que sur Internet Explorer. L'idéal est de proposer plusieurs formats pour la police : le navigateur téléchargera celui qu'il sait lire. Voici comment indiquer plusieurs formats :

@font-face {

font-family: 'MaSuperPolice';

src: url('MaSuperPolice.eot') format('eot'),

url('MaSuperPolice.woff') format('woff'),

url('MaSuperPolice.ttf') format('truetype'),

url('MaSuperPolice.svg') format('svg');

}

Pour tester le fonctionnement, je vous propose de télécharger une police sur Font Squirrel, par exemple [CAC Champagne](https://www.fontsquirrel.com/fonts/CAC-Champagne). Cliquez sur « Webfont Kit », cela vous permettra de télécharger un kit prêt à l'emploi avec tous les formats pour cette police. N'oubliez pas de cocher toutes les cases (WOFF, TTF, EOT et SVG).

La police CAC Champagne n'est plus disponible, mais vous pouvez télécharger une autre police que vous aimez bien ! ;)

Pour trouver les polices qui vous permettent de télécharger un "pack CSS" comme je le décris ici, activez le filtre "Webfont" dans le menu Font Filter.

Votre fichier CSS ressemblera au final à ceci :

@font-face { /\* Définition d'une nouvelle police nommée CAC Champagne \*/

font-family: 'cac\_champagneregular';

src: url('cac\_champagne-webfont.eot');

src: url('cac\_champagne-webfont.eot?#iefix') format('embedded-opentype'),

url('cac\_champagne-webfont.woff') format('woff'),

url('cac\_champagne-webfont.ttf') format('truetype'),

url('cac\_champagne-webfont.svg#cac\_champagneregular') format('svg');

}

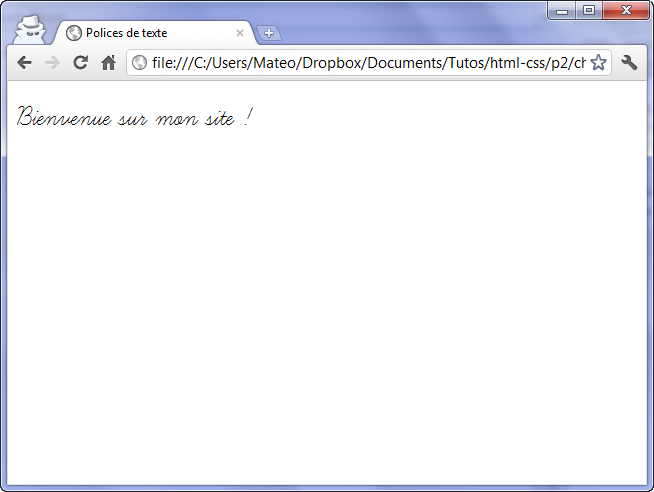
h1 /\* Utilisation de la police qu'on vient de définir sur les titres \*/

{

*font-family*: 'cac\_champagneregular', Arial, serif;

}

La première (grosse) section @font-face permet de définir un nouveau nom de police qui pourra être utilisé dans le fichier CSS. Ensuite, nous utilisons ce nom de police avec la propriété font-family  , que nous connaissons, pour modifier l'apparence des titres <h1>  . Vous pouvez voir le résultat à la figure suivante.

Affichage d'une police personnalisée

Vous noterez quelques bizarreries dans le CSS généré par le site Font Squirrel. Le but est de pallier certains bugs sur Internet Explorer, car les anciennes versions ne comprennent pas quand on définit plusieurs formats. Cela explique donc la présence d'un ?#iefix  dans le code.

**Italique, gras, souligné…**

Il existe en CSS une série de propriétés classiques de mise en forme du texte. Nous allons découvrir ici comment afficher le texte en gras, italique, souligné… et au passage nous verrons qu'il est même possible d'aller jusqu'à le faire clignoter !

**Mettre en italique**

Attends un peu là ! Je croyais que la balise <em>  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 <em>  était faite pour mettre le texte en italique (de même que je n'ai jamais dit que <strong>  était faite pour mettre en gras).

<em>  , 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 <em>  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 <strong>  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 ;
* blink  : clignotant. Ne fonctionne pas sur tous les navigateurs (Internet Explorer et Google Chrome, notamment) ;
* 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;

}

Et le résultat est visible à la figure suivante.

Différentes mises en forme du texte

**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;

}

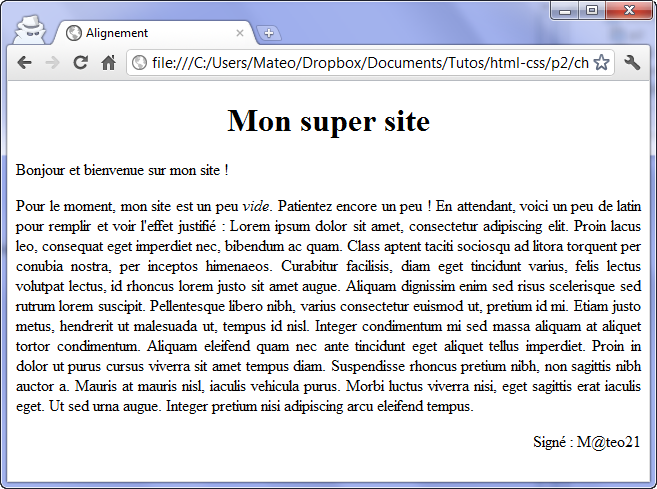
.signature

{

*text-align*: *right*;

}

Le résultat est visible sur la figure suivante.

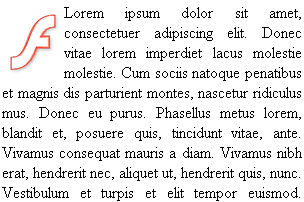
Alignements du texte

Vous ne pouvez pas modifier l'alignement du texte d'une balise *inline* (comme <span>  , <a>  , <em>  , <strong>  …). 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.

Une image flottante entourée par du texte

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><img src="flash.gif" class="imageflottante" alt="Image flottante" /> 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*;

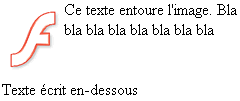
}

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 <br />  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.

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><img src="flash.gif" class="imageflottante" alt="Image flottante" /></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;

}

Et voilà le travail.

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

**En résumé**

* On modifie la taille du texte avec la propriété CSS font-size  . On peut indiquer la taille en pixels (16px), en « em » (1.3em), en pourcentage (110 %), etc.
* On change la police du texte avec font-family  . Attention, seules quelques polices sont connues par tous les ordinateurs. Vous pouvez cependant utiliser une police personnalisée avec la directive @font-face  : cela forcera les navigateurs à télécharger la police de votre choix.
* De nombreuses propriétés de mise en forme du texte existent : font-style  pour l'italique, font-weight  pour la mise en gras, text-decoration  pour le soulignement, etc.
* Le texte peut être aligné avec text-align  .
* On peut faire en sorte qu'une image soit habillée (« entourée ») de texte avec float  .