

CSS

Table des matières



I - Introduction	4
II - C'est quoi CSS ?	5
III - Où insérer du code CSS ?	7
1. Code CSS dans les balises	7
2. Code CSS dans l'en-tête	8
3. Code CSS dans fichier .css	8
IV - Commentaires en CSS	12
V - Les selecteurs	13
1. Selecteur par défaut	13
2. Sélecteur Class	14
3. Sélecteur Id	15
4. Sélecteurs avancés	16
VI - Les balises universelles	18
VII - Mettre en forme du texte	20
1. La taille du texte	20
2. La police du texte	23
3. Gras, Italique, souligner... ..	25
4. Alignement du texte	25
VIII - Flotter un éléments en CSS	27
IX - Couleur et Image en fond de la page	29
1. Couleur en CSS	29
2. Image de fond	30
3. Transparence du fond	33

X - Bordures et Ombres	35
1. Bordures	35
2. Ombres	36
XI - CSS dynamique	38
1. Au survol	38
2. Au clic	39
3. A la sélection	40
4. Déjà consulté	40

Introduction

I

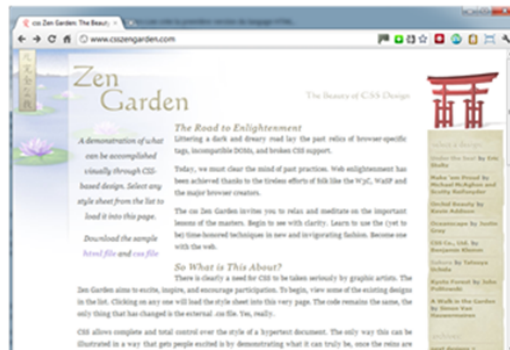
Vous avez déjà vu dans les chapitres précédents comment créer un site web uniquement en HTML, mais celui-ci ne sera pas très beau : l'information apparaîtra « brute ». C'est pour cela que le langage CSS vient toujours le compléter.

Pour vous donner une idée, la figure suivante montre ce que donne la même page sans CSS puis avec le CSS.



HTML (sans de CSS)

HTML avec CSS



Comme vous voyez dans les figures ci-dessus, le langage HTML définit le contenu "brut" de la page web (figure HTML sans CSS). Le CSS permet, lui, d'arranger le contenu et de définir la présentation : couleurs, image de fond, marges, taille du texte, etc. (figure HTML avec CSS).

Le CSS a besoin d'une page HTML pour fonctionner. C'est pour cela que nous avons consacré les deux derniers chapitre pour apprendre les bases du HTML avant de nous occuper de la décoration en CSS.

C'est quoi CSS ?

II

- Les feuilles de styles (en anglais "Cascading Style Sheets", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web.
- Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML.
- Le but de CSS est séparer la structure d'un document HTML et sa présentation.
- Il couvre toute la mise en forme : les polices, les couleurs, les marges, les arrière-plan...

Les versions de CSS

- CSS 1 : apparue en 1996. 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 : dès les 1999 la deuxième version du CSS a fait son apparition puis complétée par CSS 2.1, cette nouvelle version de CSS rajoute de nombreuses options. 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.

CSS et les navigateurs

- Lors d'affichage d'une page web, les navigateurs font le travail le plus complexe ; ils **lisent le code CSS** puis ils **comprennent comment afficher** la page.
- Les navigateurs ont connu une évolution considérable pour prendre en charge les différentes fonctionnalités de CSS
- Cependant, certain nombre de fonctionnalités de CSS qui ne marchent pas forcément sur tous les navigateurs.
- Le site www.canluse.com (nous avons déjà parlé de ce site dans le chapitre précédent) vous propose des tables de compatibilité des fonctionnalités de HTML et CSS sur différents navigateurs (et sur leurs différentes versions)

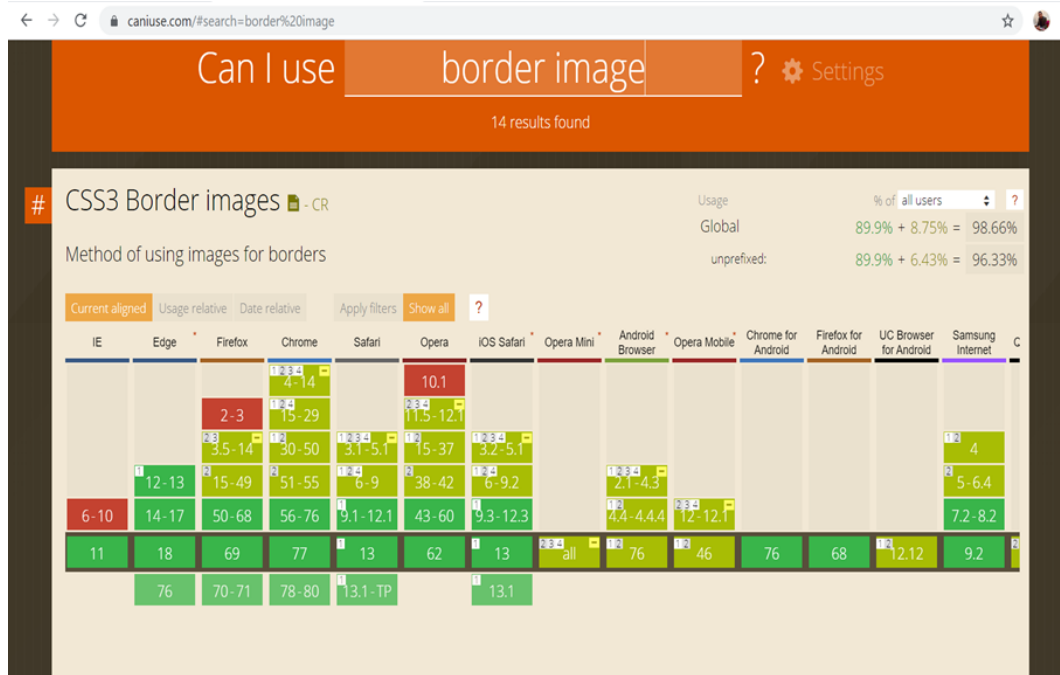


Table de compatibilité CSS sur www.caniuse.com

Où insérer du code CSS ?



Nous avons trois méthodes pour insérer du code en langage CSS dans une page HTML :

1. **Méthode 1** : Directement dans **les balises du fichier HTML** via un attribut style (méthode la moins recommandée).
2. **Méthode 2** : Dans l'**en-tête** <head>... </head> du fichier HTML
3. **Méthode 3** : Dans un **fichier «fichier.css»** (méthode la plus recommandée).

1. Code CSS dans les balises

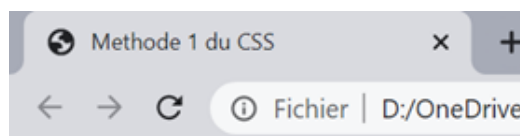
Cette première méthode permet d'appliquer directement un style CSS à n'importe quelle balise dans un document HTML avec l'attribut **style**

Dans ce cas on parle d'un style local ou interne.

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Methode 1 du CSS</title>
6   </head>
7   <body>
8     <h1>Ma page</h1>
9     <p style="color: red;">Bonjour et bienvenue sur mon site !</p>
10    <p>Ma page est encore de construction</p>
11  </body>
12 </html>

```



Ma page

Bonjour et bienvenue sur mon site !

Ma page est en construction

Comme vous voyez dans la figure, seul le texte du premier paragraphe (ligne 9), dont la balise contient le code CSS, sera coloré en rouge.

Cette méthode n'est pas recommandée et doit être manipulée, en cas de nécessité, avec précaution.

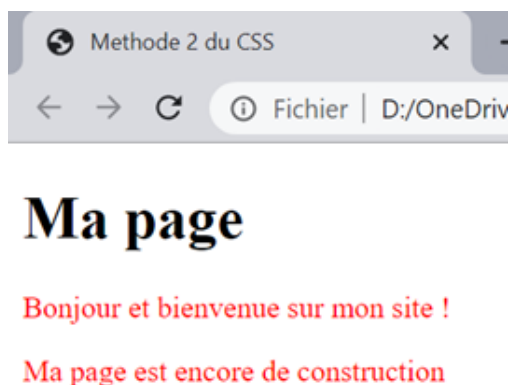
2. Code CSS dans l'en-tête

La deuxième méthode pour utiliser du CSS dans les fichiers HTML consiste à insérer le code CSS directement dans une balise `<style>` à l'intérieur de l'en-tête `<head>`.

Voici un exemple du code CSS dans le `<head>` du fichier html (lignes 6 à 11) :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Methode 1 du CSS</title>
6     <style>
7       p
8       {
9         color: red;
10      }
11    </style>
12  </head>
13  <body>
14    <h1>Ma page</h1>
15    <p>Bonjour et bienvenue sur mon site !</p>
16    <p>Ma page est encore de construction</p>
17  </body>
18 </html>
```

Le résultat de ce code est dans la figure suivante



Au contraire de la première méthode, tous les paragraphes `<p>` du fichier html sont colorés en rouge. C'est-à-dire que le code CSS est appliqué sur toutes les balises `<p>` du fichier html.

3. Code CSS dans fichier .css

Cette solution consiste à créer une feuille de style qui est un fichier externe d'extension «.css».

Ce fichier va contenir le code CSS et il sera appelé par les documents HTML.

Pour appeler la feuille de style, on ajoutera dans l'en-tête du document html (balise <head>) un lien de la forme suivante :

```
1 <link
2   rel="stylesheet"
3   href="monStyle.css"
4   type="text/css"
5 >
```

Pour implémenter cette méthode vous devez créer deux fichiers. Le premier est un fichier d'extension .html (maPage.html) qui va contenir le code html suivant :

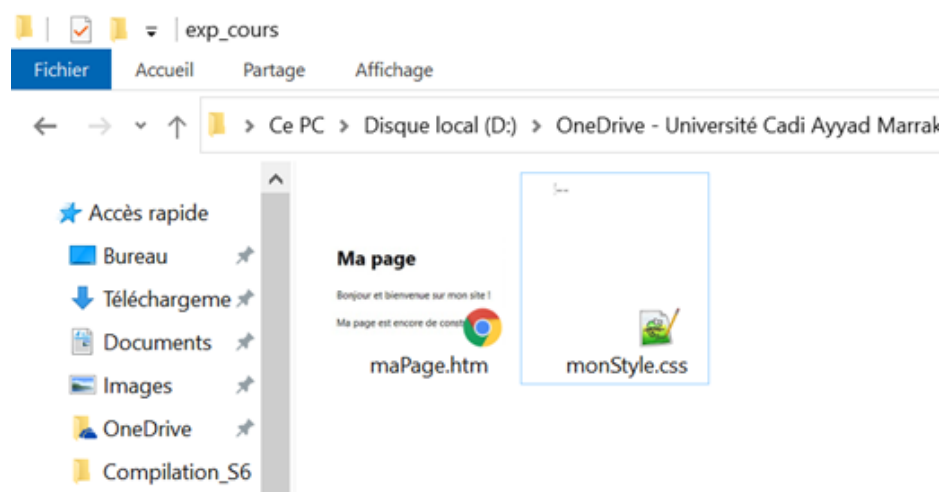
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Methode 3 du CSS</title>
6     <link rel="stylesheet" href="monStyle.css"/>
7   </head>
8   <body>
9     <h1>Ma page</h1>
10    <p >Bonjour et bienvenue sur mon site !</p>
11    <p>Ma page est encore de construction</p>
12  </body>
13 </html>
```

Le deuxième fichier va contenir le code css suivant .

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

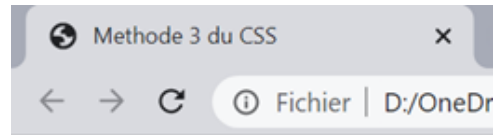
Ce fichier doit être enregistré sous l'extension .css (monStyle.css)

Dans votre explorateur de fichiers, vous devriez voir ces deux fichiers apparaître côte à côte. D'un côté le .html, de l'autre le .css, comme à la figure suivante.



Un double clic sur le fichier maPage.html permet de l'ouvrir dans votre navigateur pour le tester.

De même que le résultat de l'exemple de la méthode précédente, les paragraphes sont écrits en rouge, comme dans la figure suivante



Ma page

Bonjour et bienvenue sur mon site !

Ma page est encore de construction

Cette méthode est la plus pratique et la plus souple. Elle nous évite de mélanger l'HTML et le CSS dans un même fichier.

En plus, cette méthode permet la **réutilisation du code CSS** puisque nous pouvons appeler un même fichier CSS sur plusieurs pages HTML ainsi de **contrôler** et de **changer l'apparence** de la totalité des pages d'un site web par la modification d'un **seul fichier**. Donc, au lieu de copier/coller et modifier le code CSS dans toutes les pages html (voir figure1), on peut créer un seul fichier css qu'on peut le modifier et l'appliquer sur toutes les pages html (voir figure 2)

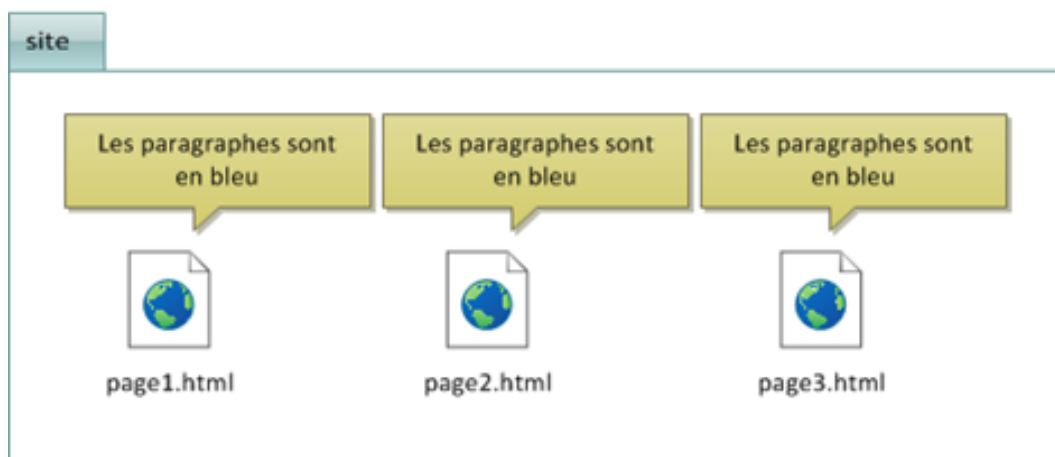


Figure 1 : Code CSS répété dans chaque fichier HTML

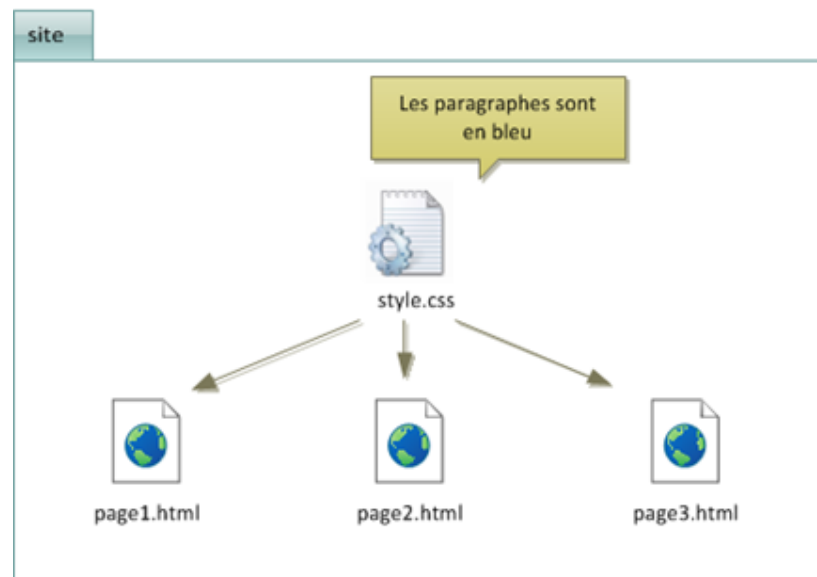


Figure 2 : Code CSS donné une fois pour toutes dans un fichier CSS et appelé dans toutes les pages de mon site

Commentaires en CSS

IV

Comme en HTML, il est possible de mettre des commentaires. Les commentaires ne seront pas affichés, ils servent simplement à indiquer des informations pour vous ou pour d'autre développeur.

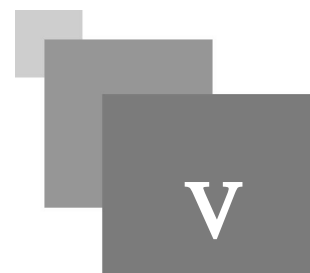
Pur faire un commentaire, c'est simple : Tapez **/***, **suivi de votre commentaire**, puis ***/** pour terminer votre commentaire.

Vos commentaires peuvent être écrits sur une ou plusieurs lignes.

Par exemple :

```
1 /*
2
3 Feuille de style : monStyle.css
4 Ecrite par : Moi Même
5
6 */
7
8 p
9 {
10     color: red; /* Les paragraphes seront en rouge*/
11 }
```

Les selecteurs



Pour appliquer un style CSS à un ou plusieurs éléments HTML, il va falloir tout d'abord les cibler via les **Sélecteurs**.

Un sélecteur est un moyen qui permet d'indiquer avec précision à quels éléments doivent s'appliquer les styles créés en CSS.

Nous pouvons distinguer différents types de sélecteurs :

- **Sélecteurs par défaut;**
- **Sélecteurs Class;**
- **Sélecteurs Id;**
- **Sélecteurs avancés.**

1. Selecteur par défaut

Tout élément HTML est un sélecteur par défaut par exemple <h1>, <h2>, <p>, <a>, etc. dont la mise en forme va être redéfinie par les **propriétés** et **valeurs**.

Une règle CSS est de la forme suivante :

```
1 Sélecteur_par_defaut {
2   Propriété1:Valeur1;
3   Propriété2:Valeur2;
4   .
5   .
6   .
7   PropriétéN:ValeurN;
8 }
```

- **Sélecteur** : ce n'est que le nom de balise, on écrit le nom de la balise dont on veut modifier l'apparence. Par exemple, si je veux modifier l'apparence de tous les paragraphes **<p>**, je dois écrire **p**.
- **Propriété** : 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. Il y a beaucoup de propriétés CSS que on n'est pas obligé à les connaître toutes par cœur.
- **Valeur** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété **color**, il faut indiquer le nom de la couleur par exemple : red, green, yellow , etc. Pour font-size, il faut indiquer quelle taille on veut, etc.

Par exemple :

```
1 p{
```

```

2  color: red;
3 }

```

Dans l'exemple précédent TOUS les paragraphes possèdent la même présentation (ici, ils seront donc tous écrits en rouge).

Si on cherche que certains paragraphes seulement soient écrits d'une manière différente ? nous pouvons placer le code CSS dans un attribut style sur la balise que l'on vise (c'est la première méthode que nous avons vue) mais, comme nous avons déjà dit, ce n'est pas recommandé (il vaut mieux utiliser un fichier CSS externe).

Pour résoudre le problème, nous pouvons utiliser des sélecteurs qui fonctionnent sur toutes les balises :

- Sélecteur class;
- Sélecteur id.

2. Sélecteur Class

Le **sélecteur class** permet de caractériser **plusieurs éléments** (identiques ou non) dans une page web. Il est spécifié en CSS par un point (.) et peut concerner tous les éléments HTML utilisant cette classe ou seulement l'un d'entre eux.

La syntaxe CSS d'un sélecteur class est la suivante :

```

1 .nom_de_classe {
2   Propriété1:Valeur1;
3   Propriété2:Valeur2;
4   .
5   .
6   .
7   PropriétéN:ValeurN;
8 }

```

Exemple de création d'un sélecteur class :

```

1 .style1{
2   color:red;
3 }

```

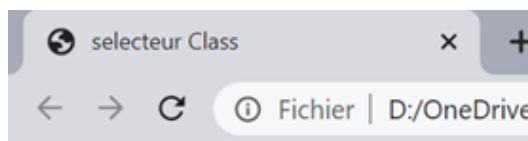
Le code HTML suivant présente une application l'utilisation du sélecteur class (lignes 10 et 12)

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>selecteur Class</title>
6     <link rel="stylesheet" href="monStyle.css"/>
7   </head>
8   <body>
9     <h1>Ma page</h1>
10    <p class="style1">Bonjour et bienvenue sur mon site !</p>
11    <p>Ma page est encore de construction</p>
12    <p class="style1">Je la terminerai prochainement</p>
13
14   </body>
15 </html>

```

Le résultat de l'application du code CSS sur la page HTML est illustré dans la figure suivante. Comme vous voyez dans cet exemple nous avons appliqué le style CSS via le sélecteur class sur deux paragraphes seulement (pas sur tous les paragraphes comme le cas du sélecteur par défaut)



Ma page

Bonjour et bienvenue sur mon site !

Ma page est encore de construction

Je la terminerai prochainement

3. Sélecteur Id

Le **sélecteur ID** (d'identifiant) ne peut être appliqué qu'à **un élément et un seul élément dans le code HTML**, par exemple un seul paragraphe. Il concerne donc les éléments uniques de la structure du document HTML, comme les blocs principaux (logo, en-tête, colonne(s), pied de page, etc).

Le sélecteur Id est spécifié dans un fichier CSS par un dieze # suivie de l'id.

La syntaxe CSS d'un sélecteur Id est la suivante :

```
1 #Selecteur_Id {
2     Propriété1:Valeur1;
3     Propriété2:Valeur2;
4     .
5     .
6     .
7     PropriétéN:ValeurN;
8 }
```

Exemple :

```
1 #style1{
2     color:red;
3 }
```

Pour appliquer ce style vous devez ajouter dans la balise de l'élément concerné l'attribut **id= "nom de l'identifiant"** comme indiqué dans l'exemple suivant (ligne 10) :

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="utf-8" />
5         <title>selecteur ID</title>
6         <link rel="stylesheet" href="monStyle.css"/>
7     </head>
8     <body>
```

```

9         <h1>Ma page</h1>
10        <p id="style1">Bonjour et bienvenue sur mon site !</p>
11        <p>Ma page est encore de construction</p>
12
13    </body>
14</html>

```

4. Sélecteurs avancés

En plus que les selecteurs par défauts, Class et ID , Il existe des dizaines d'autres façons de cibler des balises en CSS. Nous n'allons pas toutes les voir car il y en a beaucoup et certaines sont complexes, mais voici quelques uns qui vous permettent d'être plus efficaces en CSS :

- * : sélecteur universel

Il permet de sélectionner toutes les balises sans exception. On l'appelle le sélecteur universel.

```
1 *{
2     . . .
3 }
```

- **Balise1 Balise2** : une balise contenue dans une autre

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

Ce code permet de sélectionner toutes les balises `` situées à l'intérieur d'une balise `<h3>`.

Exemple de code HTML correspondant :

```
1 <h3>Titre avec <em>texte important</em></h3>
```

- **Balise1 + Balise2** : une balise qui en suit une autre

```
1 h3 + p
2 {
3   ...
4 }
5
```

Ce code est pour sélectionner la première balise `<p>` située après un titre `<h3>`.

Example :

```
1 <h3>Titre</h3>
2
3 <p>Paragraphe à selectionner</p>
```

- **Balise[attribut]** : une balise qui possède un attribut

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

Ce code permet de sélectionner tous les liens qui possèdent un attribut title.

Exemple :


```
1 <a href="http://site.com" title="Visiter">
```

- **Balise[attribut="Valeur"] : une balise, un attribut et une valeur exacte**

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

De même que l'exemple précédent sauf ici le titre doit contenir exactement la valeur : Cliquez ici

Exemple :

```
1 <a href="http://site.com" title="Cliquez ici">
```

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

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

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

Exemple :

```
1 <a href="http://site.com" title="Quelque part par ici">
```

Les balises universelles

VI

En HTML, la plupart des balises peuvent se ranger dans l'une ou l'autre de deux catégories :

- Les balises «**inline**» :
 - C'est le cas par exemple des liens `<a> .. `.
 - Cette catégorie de balise crée automatiquement un retour à la ligne avant et après.
 - Il suffit d'imaginer tout simplement un bloc. Votre page web sera en fait constituée d'une série de blocs les uns à la suite des autres.
- Les balises «**block**» :
 - C'est le cas par exemple des paragraphes `<p> .. </p>`.
 - Une balise de type inline se trouve obligatoirement à l'intérieur d'une balise block.
 - Une balise inline ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne (c'est pour cela que l'on parle de balise « en ligne »).

Les balises universelles « span » et « div » :

- Ce sont des balises qui n'ont aucun sens particulier. Contrairement à `<p>` qui veut dire «paragraphe», `` «important», etc.
- Le principal intérêt de ces balises est que l'on peut leur appliquer une class (ou un id) pour le CSS quand aucune autre balise ne convient.
- la balise ` .. ` est **inline**, tandis que `<div> .. </div>` est un **block**.
- «span» et «div» sont « pratiques » dans certains cas, certes, mais attention à ne pas en abuser.
- Exemple d'un **span inutile** : ``. Inutile puis qu'il existe la balise `` qui sert à indiquer l'importance !
- Exemple d'un **div inutile** : `<div class="titre">`. Ceci est complètement absurde puisqu'il existe des balises faites spécialement pour les titres (`<h1>`, `<h2>`...).

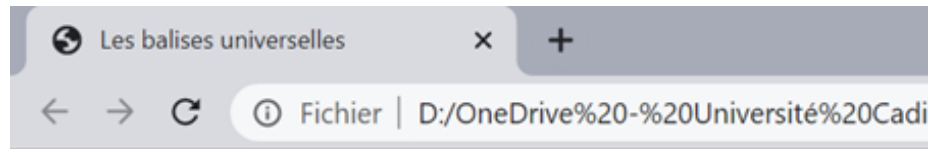
Exemple d'utilisation :

```
1 .partie1{
2   color:red;
3   font-weight: bold;
4 }
```

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" href="monstyle.css">
5     <title>Les balises universelles</title>
6   </head>
7
8   <body>
9
```

```
10 <p>Un paragraphe n'est pas de la partie 1, sauf  
11 <span class="partie1">ce mot</span> et pas celui-ci.  
12 </p>  
13  
14 </body>  
15 </html>
```

Comme vous allez remarqué dans la figure suivante, nous avons appliqué le style sur une partie d'un paragraphe en utilisant la balise `` (ligne 11).



Mettre en forme du texte

VII

Dans cette partie nous allons découvrir les propriétés CSS qui permettent de mettre en forme du texte.

La mise en forme du texte consiste à modifier la **taille du texte**, changer la **police**, **aligner** le texte, etc

1. La taille du texte

La propriété CSS **font-size** permet de modifier la taille du texte. Et pour attribuer une valeur à la propriété **font-size**, nous avons plusieurs techniques :

1. **Indiquer une taille absolue** : en **pixels**, en **centimètres** ou **millimètres**, etc. 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**.
 2. **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**.
- **Texte en taille absolue**

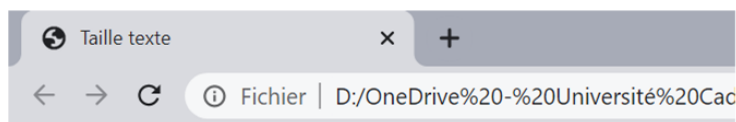
Pour indiquer une taille absolue, on utilise généralement les pixels.

Exemple:

```
1 .taille1{
2   font-size: 08px;
3 }
4
5 .taille2{
6   font-size: 15px;
7 }
8
9 .taille3{
10  font-size: 30px;
11 }
```

```
1 <h1>Ma page</h1>
2 <p class="taille1">Bonjour et bienvenue sur mon site !</p>
3 <p class="taille2">Ma page est encore de construction</p>
4 <p class="taille3">Je la terminerai prochainement</p>
```

Le résultat de ce code est présenté dans la figure suivante :



Ma page

Texte en 8 px

Bonjour et bienvenue sur mon site !

Texte en 15 px

Ma page est encore de construction

Texte en 30 px

Je la terminerai prochainement

- Texte en taille relative

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**: gigantesque

C'est la méthode la plus recommandée car elle permet au texte de s'adapter aux préférences de tous les visiteurs.

Exemple :

```

1 .taille1{
2   font-size: x-small;
3 }
4
5 .taille2{
6   font-size: medium;
7 }
8
9 .taille3{
10  font-size: x-large;
11 }
```



La limite de cette technique c'est qu'elle propose sept tailles seulement. Pour cela vous pouvez utiliser une autre solution (c'est la plus recommandée) qui consiste à indiquer la taille en « **em** ».

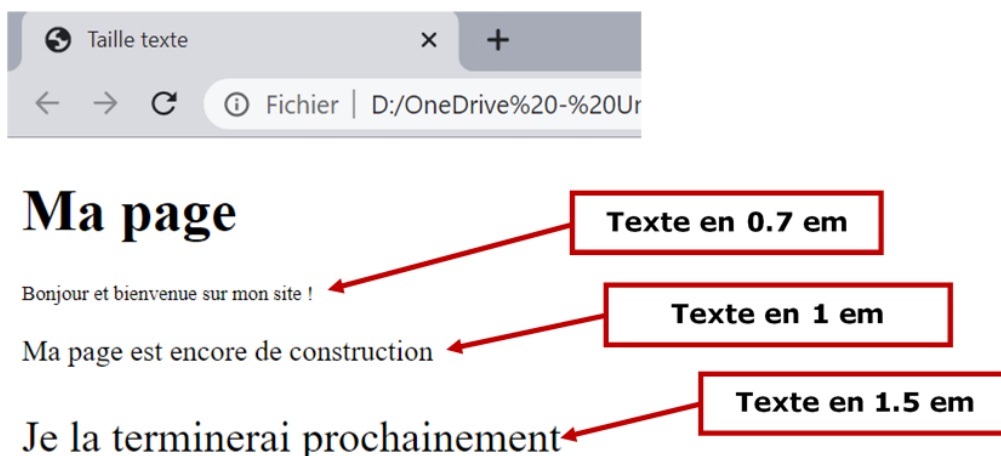
Si vous mettez 1em, le texte a une taille normale.

Si vous voulez agrandir le texte, vous pouvez écrire une valeur supérieure à 1, comme 1.5 em.

Si vous voulez réduire le texte, mettez une valeur inférieure à 1, comme 0.7 em.

Exemple :

```
1 .taille1{
2   font-size: 0.7em;
3 }
4
5 .taille2{
6   font-size: 1em;
7 }
8
9 .taille3{
10  font-size: 1.5em;
11 }
```



Pour savoir plus sur les tailles de texte vous pouvez consulter *ce lien*.

2. La police du texte

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

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

Pour éviter les problèmes si l'internaute n'a pas la même police que vous, vous précisez plusieurs noms de police, séparés par des virgules comme ceci :

```
1 balise
2 {
3     font-family: police1, police2, police3, police4;
4 }
```

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

En général, on indique en tout dernier **serif**, ce qui correspond à une **police par défaut**.

Exemple :

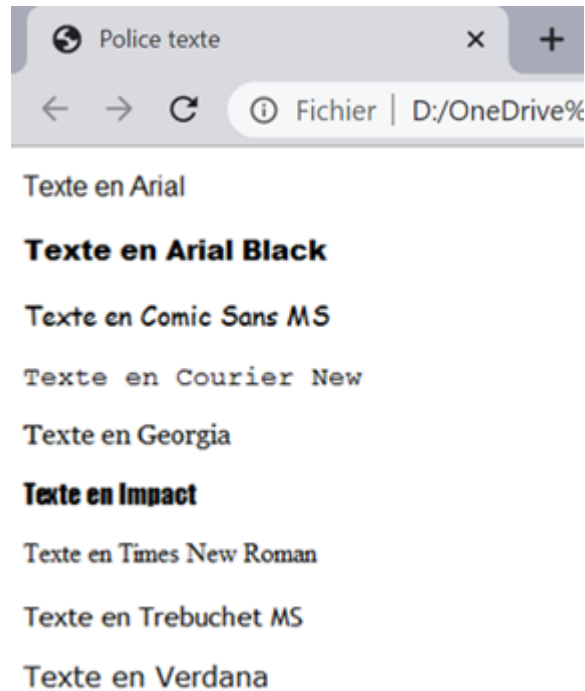
```
1 p{
2     font-family: Impact, "Arial Black", Arial, Verdana, sans-serif;
3 }
4
```

Le navigateur va utiliser tout d'abord la police "Impact". S'il ne l'a pas il va essayer la police "Arial Black" puis "Arial" et ainsi de suite.

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 présente à quoi ressemblent ces polices :



Personnaliser la police

Pour définir une nouvelle police, vous pouvez utiliser la propriété **@font-face** comme ceci :

```
1 @font-face {  
2   font-family: 'NouvellePolice';  
3   src: url('NouvellePolice.ttf') format('ttf'),  
4       url('NouvellePolice.eot') format('eot'),  
5       url('NouvellePolice.otf') format('otf'),  
6       url('NouvellePolice.svg') format('svg'),  
7       url('NouvellePolice.woff') format('woff');  
8 }
```

Dans cet exemple nous avons proposé plusieurs formats pour la police :

- .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 iPhones et iPads pour le moment.
- .woff: Web Open Font Format. Nouveau format conçu pour le Web, qui fonctionne sur IE9 et tous les autres navigateurs.

Cela permet au navigateur de télécharger la police qu'il sait lire.

Tous les fichiers NouvellePolice.ext doivent être placés au même dossier que le fichier CSS

La plupart des polices sont soumises au droit d'auteur, il n'est donc pas légal de les utiliser sur son site.

Cependant, il existe des sites comme *fontquirrel.com* qui proposent le téléchargement des packs de police libres de droits et prêts à l'emploi pour CSS3.

3. Gras, Italique, souligner...

Dans cette partie nous allons voir les propriétés classiques de mise en forme du texte. Nous allons découvrir comment afficher le texte en gras, italique, souligné...etc

Texte 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 (par défaut).

Texte en Italique

En CSS pour mettre un texte en italique, on utilise **font-style**. Cette propriété 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.

Soulignement de texte

La propriété CSS pour décorer un texte est **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).

Voici par exemple comment écrire les titres en gras , italique et souligné :

```
1 h1{
2   font-weight: bold;
3   text-decoration : underline;
4   font-style : italic;
5 }
```

4. Alignement du texte

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.

L'exemple suivant présente les différents alignements :



Flotter un éléments en CSS

VIII

Le CSS permet de faire flotter un élément autour du texte, ce que nous appelons aussi un « habillage ».

La figure suivante présente l'habillage du texte dans un logiciel de traitement de texte.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci. Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

En CSS la propriété **float** permet de faire cette action. Cette propriété peut prendre deux valeurs très simples :

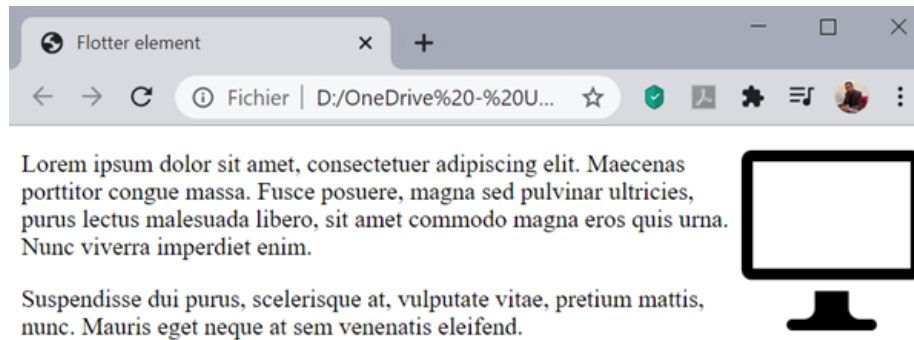
- **left**: l'élément flottera à gauche.
- **right**: l'élément flottera à droite.

L'utilisation des flottants est très simple, vous appliquez un float à l'élément concerné, puis vous continuez à écrire du texte à la suite normalement.

Exemple :

```
1 .imageflottante
2 {
3     float: right;
4 }
```

```
1 <p>
2   Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor
  congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada
  libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim.</p>
3   <p>Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis,
  nunc. Mauris eget neque at sem venenatis eleifend.</p>
```



Dans la figure précédente, vous remarquez que l'image est flotter aussi sur le deuxième paragraphe. Si vous voulez stopper le float sur le deuxième paragraphe vous devez utiliser 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;

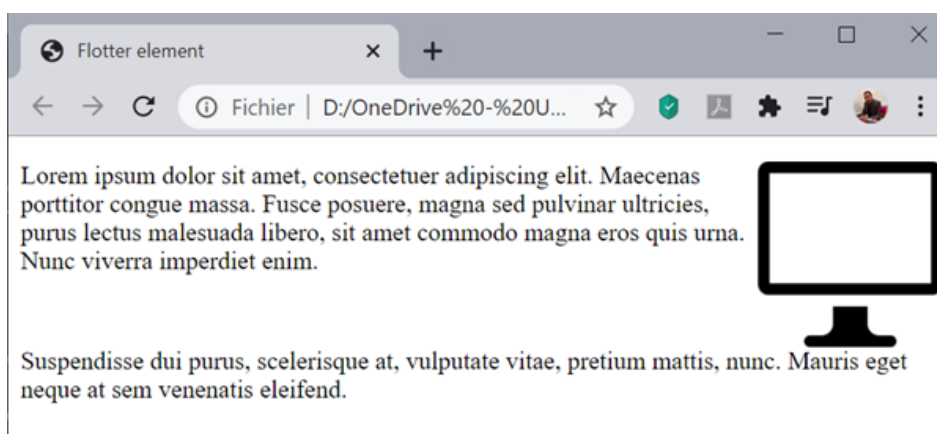
Exemple :

```
1 .imageflottante
2 {
3     float: right;
4 }
5
6 .dessous
7 {
8     clear: both;
9 }
```

Puis vous ajoutez la classe `.dessous` dans la balise `<p>` du deuxième paragraphe comme suit :

```
1 <p class="dessous">Suspendisse dui purus, scelerisque at, vulputate vitae,
   pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend.</p>
```

Le résultat sera comme suit :



Couleur et Image en fond de la page

IX

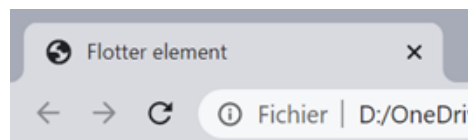
1. Couleur en CSS

En CSS la propriété **color** permet de changer la couleur. Cette propriété accepte trois types de valeur :

- **Nom de la couleur** en anglais par exemple : red, green, yellow , etc.
- **Code hexadécimale** de la couleur, exemple : #FF0000 pour le rouge
- **Notation RGB** comme rgb(Rouge, Vert, Bleu) par exemple : rgb(255,0,0) pour le rouge

Exemple d'utilisation des différentes méthodes de modification de la couleur :

```
1 h1{
2   color: blue;
3 }
4
5 h2{
6   color : #A34E03;
7 }
8
9 h3{
10  color : rgb(78,181,74);
11 }
```



Titre 1

Titre 2

Titre 3

Cette *page* vous permet de trouver facilement une couleur avec les différentes notations.

Couleur de fond

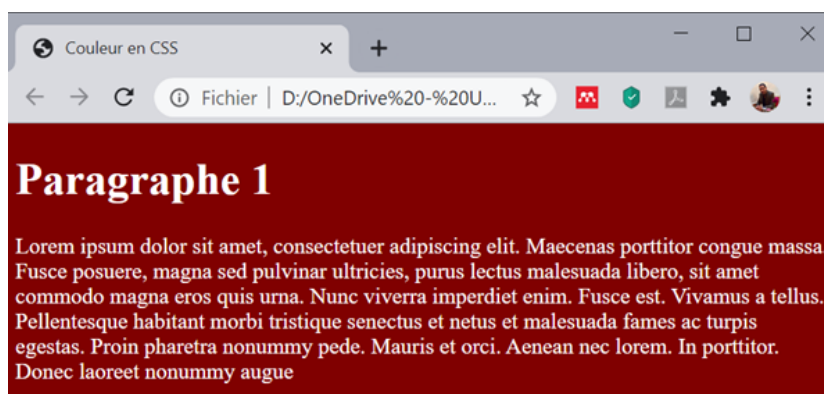
Pour modifier la couleur du fond de la page vous pouvez utiliser la propriété **background-color**. Elle s'utilise de la même manière que la propriété color, c'est-à-dire que vous pouvez taper le nom d'une couleur, l'écrire en notation hexadécimale ou encore utiliser la méthode RGB.

Pour modifier la couleur du fond de la page vous devez mettre cette propriété à l'intérieur de l'élément body.

Exemple :

```
1 body
2 {
3     background-color: maroon; /* Le fond de la page sera marron */
4     color: white; /* Le texte de la page sera blanc */
5 }
```

```
1 <body>
2     <h1> Paragraphe 1 </h1>
3     <p>
4         Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor
congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada
libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce
est. Vivamus a tellus. Pellentesque habitant morbi tristique senectus et netus et
malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.
Aenean nec lorem. In porttitor. Donec laoreet nonummy augue</p>
5 </body>
```



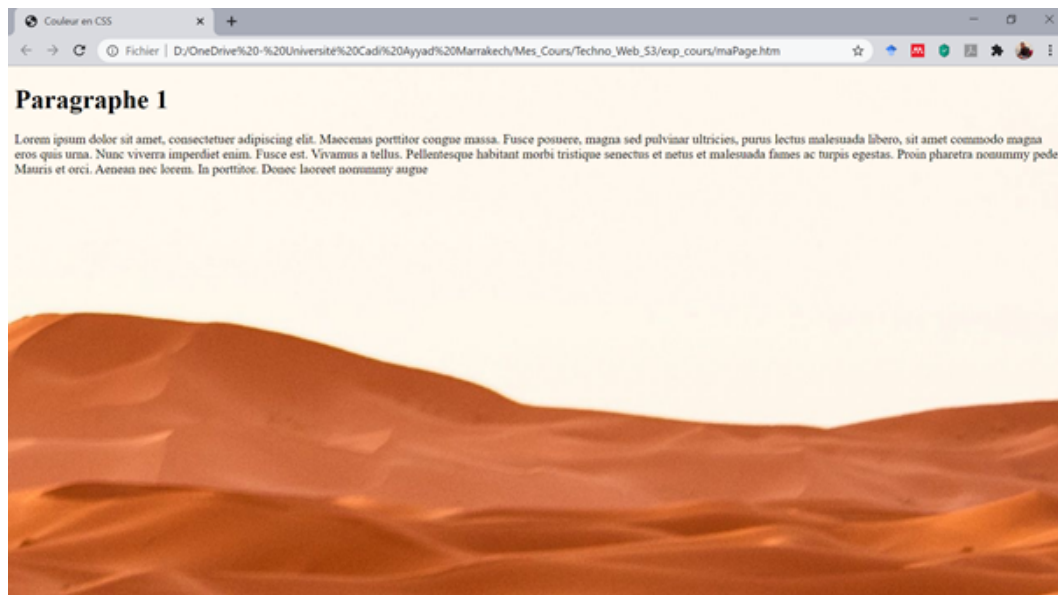
Vous remarquez que nous avons appliqué une couleur blanche sur le texte de la balise <body> et les autres balises de paragraphe<p> et de titre<h1>, ont pris le même style. C'est ce phénomène qu'on appelle l'**héritage** : on dit que les balises qui se trouvent à l'intérieur d'une autre balise « **héritent** » de ses propriétés.

2. 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")`. Par exemple :

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

Ce qui nous donne la figure suivante.



Options pour 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).
- **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).
- **background-position** : position du fond
Cette propriété permet l'endroit de l'image de fond dans la page. Il est aussi possible d'utiliser ces valeurs en anglais :
 - **top** : en haut ;
 - **bottom** : en bas ;
 - **left** : à gauche ;
 - **center** : centré ;
 - **right** : à droite.

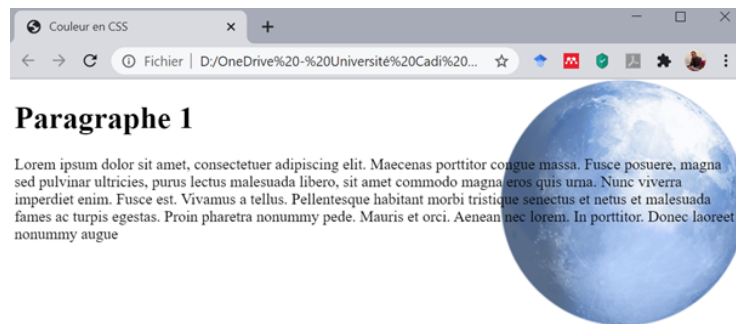
Comme il est possible de donner des coordonnées (x,y) en pixel . par exemple :

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

Notez bien que cette propriété n'est intéressante que si elle est combinée avec background-repeat: no-repeat; (un fond qui ne se répète pas).

Exemple de toutes les propriétés précédentes :

```
1 body
2 {
3   background-image: url("lune.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 }
```



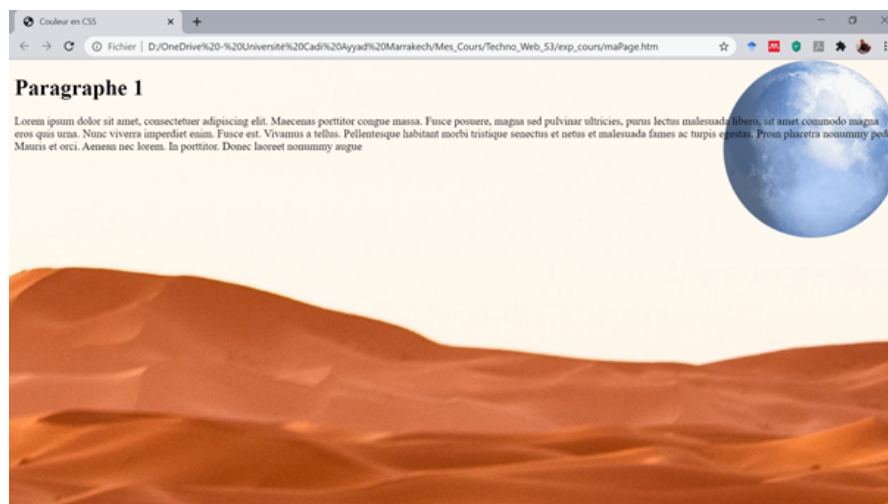
Un écriture plus simple du code précédent :

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

- Plusieurs images de fond :

Depuis CSS3, il est possible de donner plusieurs images de fond à un élément. Pour cela, il suffit de séparer les déclarations par une virgule, comme ceci :

```
1 body
2 {
3   background: url("lune.png") fixed no-repeat top right, url("desert.jpg") fixed
4 ;
5 }
```



3. Transparence du fond

La propriété **opacity** et la notation **RGBA** permettent de modifier les niveaux de transparence des éléments.

- **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.

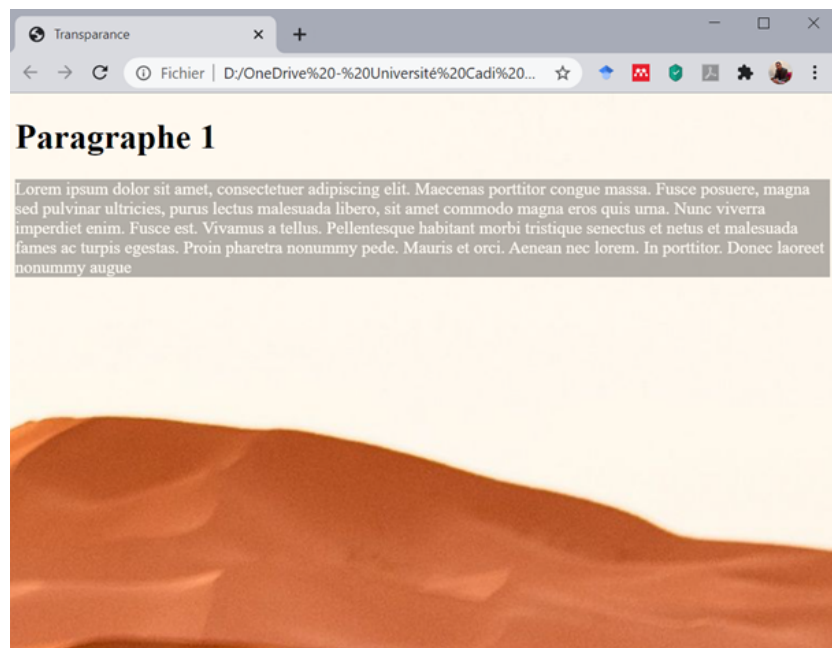
Il faut donc choisir une valeur comprise entre 0 et 1. Ainsi, avec une valeur de 0.6, votre élément sera opaque à 60 %

Voici comment on peut l'utiliser :

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

Un exemple complet :

```
1 body
2 {
3   background: url('desert.jpg');
4 }
5
6 p
7 {
8   background-color: black;
9   color: white;
10  opacity: 0.3;
11 }
```

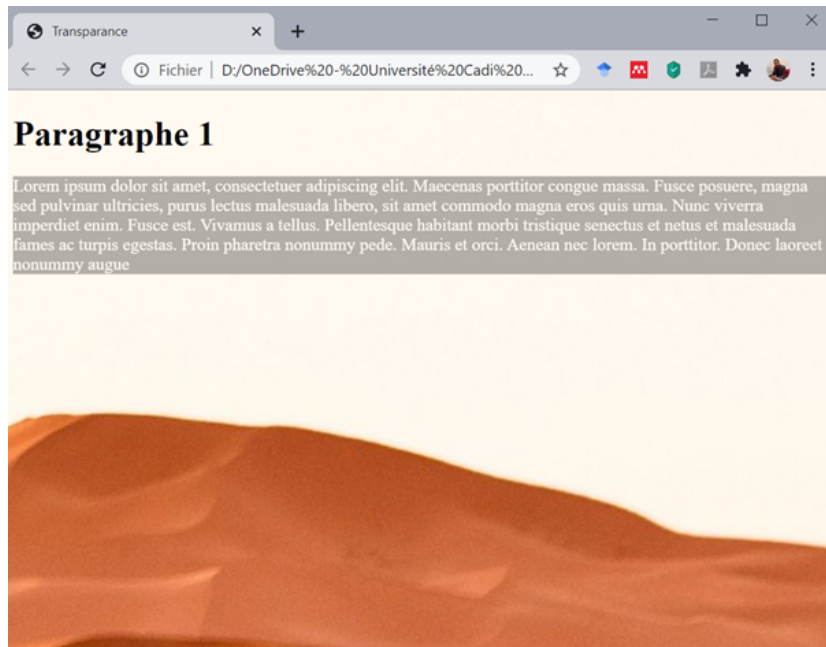


- **La notation RGBA :**

C'est une autre façon de jouer avec la transparence. Il s'agit en fait de la notation RGB, mais avec un quatrième paramètre : le niveau de transparence (appelé « canal alpha »). Avec une valeur de 1, le fond est complètement opaque. Avec une valeur inférieure à 1, il est transparent.

Example :

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



Bordures et Ombres



1. Bordures

En CSS la super-propriété **border** permet de jouer sur les bordures dans une page HTML.

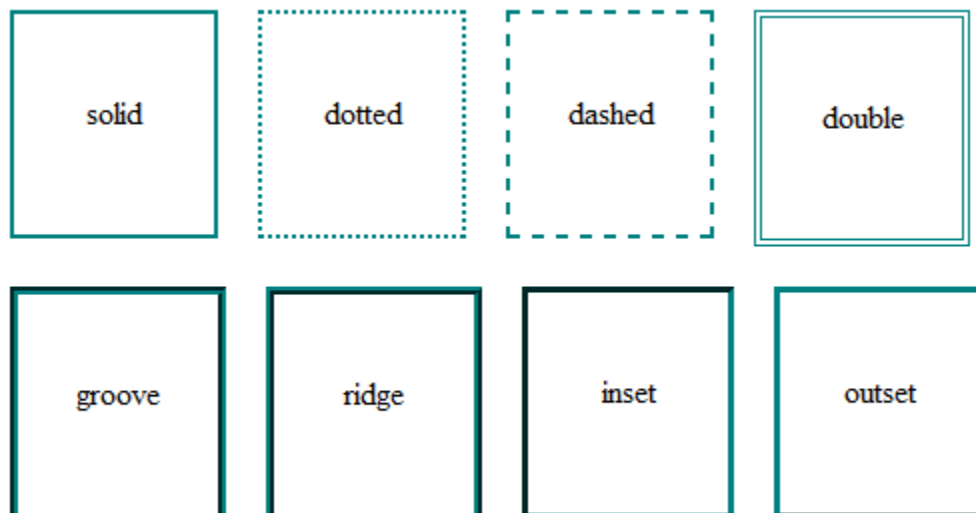
Vous pouvez associer trois valeurs à **border** pour modifier l'apparence de la bordure :

- **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** : votre bordure peut être un simple trait, ou des pointillés, ou encore des tirets, etc.

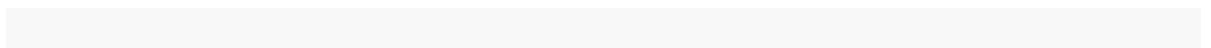
Voici les différentes valeurs disponibles :

- **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.

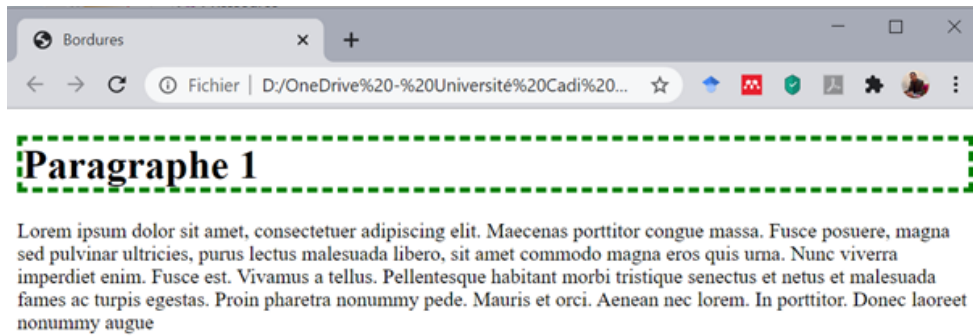
Les différents styles de bordure sont présentés dans la figure suivante :



Exemple d'utilisation de la super-propriété **border** :



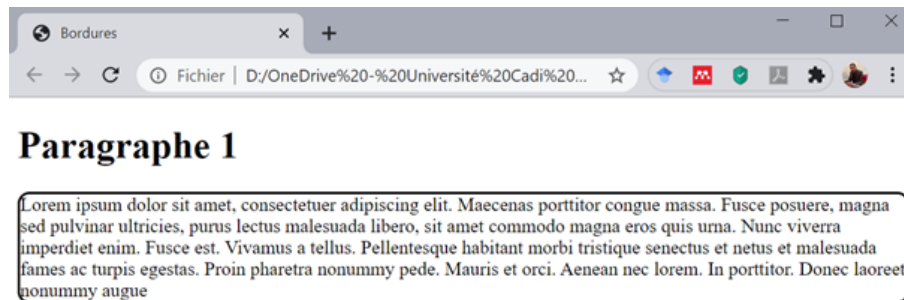
```
1 h1
2 {
3     border: 4px gren groove;
4 }
```



Si vous voulez arrondi la bordure vous pouvez utiliser la propriété **border-radius**.

Example :

```
p
{
border: 2px solid black;
border-radius: 10px;
}
```



2. Ombres

Nous pouvons distinguer deux types d'ombres :

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

Les ombres des boîtes

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

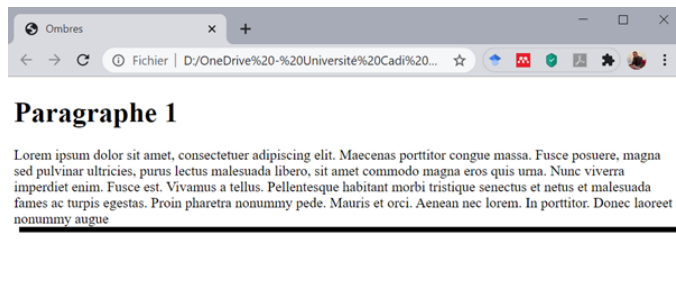
- Le décalage horizontal de l'ombre.
- Le décalage vertical de l'ombre.
- L'adoucissement du dégradé.
- La couleur de l'ombre.

Par exemple, pour insérer une ombre de couleur noire, 6 pixels, sans adoucissement, on écrira :

```

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

```



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.

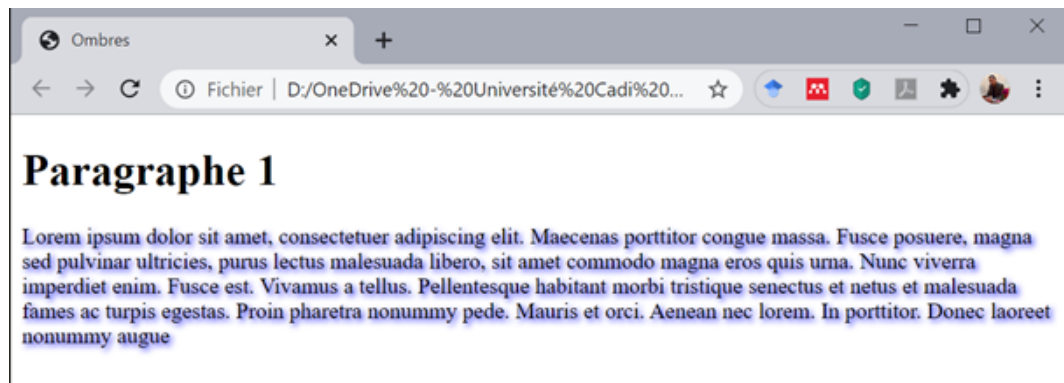
Exemple :

```

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

```

Le résultat est illustré à la figure suivante :



CSS dynamique

XI

En CSS, vous pouvez modifier aussi l'apparence des éléments de façon **dynamique**, c'est-à-dire que des éléments peuvent changer de forme une fois que la page a été chargée. Pour cela nous allons faire appel à une fonctionnalité puissante du CSS : les **pseudo-formats**.

Nous verrons dans cette partie comment changer l'apparence :

- au survol ;
- lors du clic ;
- lors du focus (élément sélectionné) ;
- lorsqu'un lien a été consulté.

1. Au survol

Le premier **pseudo-formats** que nous allons découvrir s'appelle **:hover**.

Comme tous les autres pseudo-formats que nous allons voir, c'est une information que l'on rajoute après le nom de la balise (ou de la classe) dans le CSS.

```
1 a:hover  
2 {  
3 }
```

:hover signifie « survoler ». a:hover peut donc se traduire par : « Quand la souris est sur le lien » (quand on pointe dessus).

Exemple :

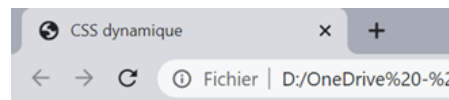
```
1 /* Liens par défaut (non survolés) */  
2 a{  
3   text-decoration: none;  
4   color: blue;  
5   font-style: italic;  
6 }  
7 a:hover /* Apparence au survol des liens */  
8 {  
9   text-decoration: underline;  
10  color: red;  
11 }
```

Dans ce code, nous avons défini ici deux versions des styles pour les liens :

- pour les liens par défaut (non survolés) ;

- pour les liens au survol.

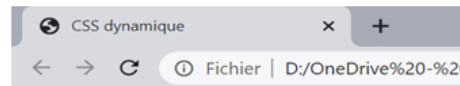
Les figures suivantes présentent les deux apparences du lien suite au survol.



Paragraphe 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc viverra imperdiet enim. Fusce Mauris et orci. Aenean nec lorem. In porttitor. Donec

[Cliquer ici](#)



Paragraphe 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc viverra imperdiet enim. Fusce Mauris et orci. Aenean nec lorem. In porttitor. Donec

[Cliquer ici](#)

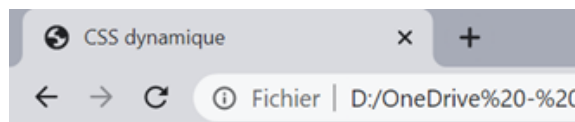
2. Au clic

Le pseudo-format **:active** permet d'appliquer un style particulier au moment du clic. En pratique, il n'est utilisé que sur les liens.

Le lien gardera cette apparence très peu de temps : en fait, le changement intervient lorsque le bouton de la souris est enfoncé. En clair, ce n'est pas forcément toujours bien visible.

Dans cet exemple nous allons changer la couleur de fond du lien en noir et le texte en blanc lorsque le visiteur clique dessus :

```
1 a:active /* Quand le visiteur clique sur le lien */
2 {
3   background-color: #000000;
4   color: white;
5 }
```



Paragraphe 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc viverra imperdiet enim. Fusce Mauris et orci. Aenean nec lorem. In porttitor. Donec

[Cliquer ici](#)

3. A la sélection

Pour modifier l'apparence d'un élément lorsqu'il est sélectionné on utilise le pseudo-format **:focus**.

Exemple :

```
1 a:focus /* Quand le visiteur sélectionne le lien */
2 {
3   background-color: purple;
4   color: white;
5 }
6
```

Le résultat se trouve dans la figure suivante.



4. Déjà consulté

Il est possible d'appliquer un style à un lien vers une page qui a déjà été vue.

Par défaut, le navigateur colore le lien en un violet.

Vous pouvez changer cette apparence avec **:visited** (qui signifie « visité »). En pratique, sur les liens consultés, on ne peut pas changer beaucoup de choses à part la couleur.

Exemple :

```
1 a:visited /* Quand le visiteur a déjà vu la page concernée */
2 {
3   color: #AAA; /* Appliquer une couleur grise */
4 }
```



