

## Partie 2 : CSS ou feuille de Style

Maintenant que vous connaissez les bases du HTML... donnez du *style* à votre page grâce au CSS !

### Mettre en place le CSS

Après avoir passé toute une première partie du cours à ne travailler que sur le HTML, nous allons maintenant découvrir le CSS que j'avais volontairement mis à l'écart. Le CSS n'est pas plus compliqué que le HTML. Il vient le compléter pour vous aider à mettre en forme votre page web.

Dans ce premier chapitre sur le CSS, nous allons voir la théorie sur le CSS : qu'est-ce que c'est ? À quoi cela ressemble-t-il ? Où est-ce qu'on écrit du code CSS ? Ces aspects théoriques ne sont pas bien compliqués mais vous devez obligatoirement les connaître car c'est la base du CSS. C'est d'ailleurs la seule chose que je vous demanderai de retenir par cœur en CSS.

### La petite histoire du CSS

Je vous avais avertis dès le début de ce cours : nous allons apprendre deux langages. Nous avons déjà bien entamé notre découverte du HTML, même s'il reste encore de nombreuses choses à apprendre (nous y reviendrons dans quelques chapitres). En revanche, il est temps maintenant de nous intéresser au CSS.

CSS (*Cascading Style Sheets*), c'est cet autre langage qui vient compléter le HTML. Vous vous souvenez de son rôle ? *Gérer la mise en forme de votre site.*

### Petit rappel : à quoi sert CSS ?

CSS ? C'est lui qui vous permet de choisir la couleur de votre texte. Lui qui vous permet de sélectionner la police utilisée sur votre site.  
Lui encore qui permet de définir la taille du texte, les bordures, le fond...

Et aussi, c'est lui qui permet de faire la mise en page de votre site. Vous pourrez dire : je veux que mon menu soit à gauche et occupe telle largeur, que l'en-tête de mon site soit calé en haut et qu'il soit toujours visible, etc.

HTML  
(pas de CSS)



HTML + CSS



Grâce au HTML, nous avons pu rédiger le contenu de notre site mais il est brut. Le CSS vient compléter ce code pour mettre en forme tout cela et donner au contenu l'apparence que l'on souhaite.

## CSS : des débuts difficiles

Il faut savoir qu'aux débuts du Web, CSS n'existait pas. En fait, il n'y avait initialement que le langage HTML. Le HTML est né en 1991 et CSS en 1996. Alors, vous vous dites sûrement : comment faisait-on la mise en forme de 1991 à 1996 ? Eh bien, uniquement en HTML ! Il y avait en effet des balises HTML dédiées à la mise en forme. `<font color="#aabc3">`, par exemple, permettait de définir la couleur du texte.

Cependant, les pages HTML commençaient à devenir assez complexes. Il y avait de plus en plus de balises et c'était un joyeux mélange entre le fond et la forme, qui rendait la mise à jour des pages web de plus en plus complexe. C'est pour cela que l'on a créé le langage CSS.

## CSS : la prise en charge des navigateurs

Tout comme le HTML, le CSS a évolué. Je vous avais indiqué qu'il y avait quatre versions importantes de CSS :

- CSS 1 ;
- CSS 2.0 ;
- CSS 2.1 ;
- CSS 3.

Ce sont les navigateurs web qui font le travail le plus complexe : ils doivent *lire* le code CSS et *Comprendre* comment afficher la page.

*Que faut-il retenir de tout cela ?* Que les navigateurs ne connaissent pas toutes les propriétés CSS qui existent. Plus le navigateur est vieux, moins il connaît de fonctionnalités CSS.

## Où écrit-on le CSS ?

Vous avez le choix car on peut écrire du code en langage CSS à trois endroits différents :

- Dans un fichier `.css` (*méthode la plus recommandée*) ;
- Dans l'en-tête `<head>` du fichier HTML ;
- Directement dans les balises du fichier HTML via un attribut `style` (*méthode la*

*moins recommandée*). Je vais vous présenter ces trois méthodes mais sachez d'ores et déjà que la première... est la meilleure.

### **Dans un fichier. ccs (recommandé)**

Comme je viens de vous le dire, on écrit le plus souvent le code CSS dans un fichier spécial ayant l'extension `.css` (contrairement aux fichiers HTML qui ont l'extension `.html`). C'est la méthode la plus pratique et la plus souple. Cela nous évite de tout mélanger dans un même fichier. J'utiliserai cette technique dans toute la suite de ce cours.

Nous allons partir du fichier HTML suivant :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" type="text/css" href="style.css" />
    <title>Premiers tests du CSS</title>
  </head>

  <body>
    <h1>Mon super site</h1>

    <p>Bonjour et bienvenue sur mon site !</p>
    <p>Pour le moment, mon site est un peu <em>vide</em>.
    Patientez encore un peu !</p>
  </body>
</html>
```

Vous noterez le contenu de la ligne 5, `<link rel="stylesheet" href="style.css" />` : C'est elle qui indique que ce fichier HTML est associé à un fichier appelé `style.css` et chargé de la Mise en forme.

Maintenant, créez un *nouveau* fichier vide dans votre éditeur de texte (par exemple Notepad++) et copiez-y ce bout de code CSS (rassurez-vous, je vous expliquerai tout à l'heure ce qu'il veut dire) :

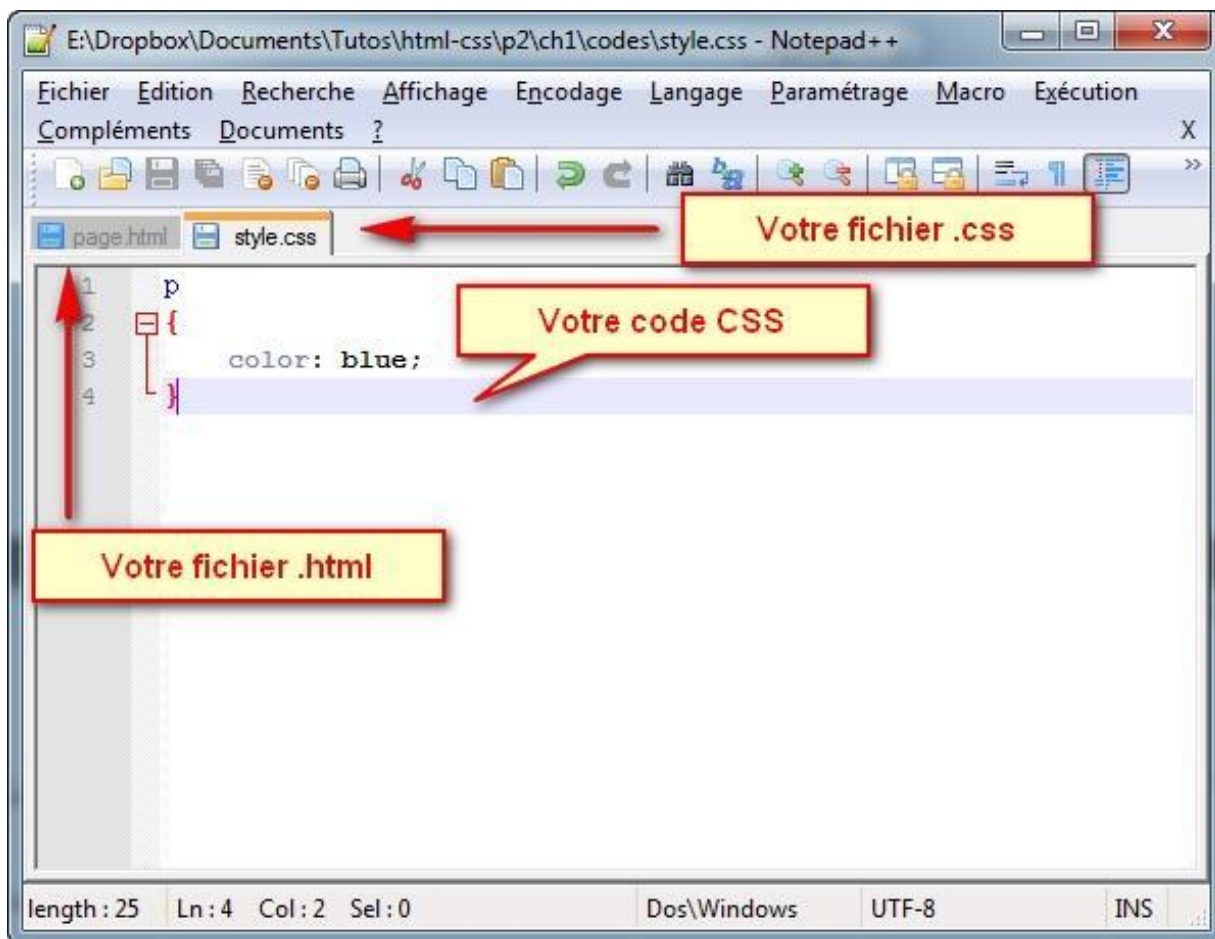
Code : CSS

```
p {
  color: blue;
}
```

Pour activer la coloration du code dans Notepad++, passez par les menus **Langage > C > CSS**.

Enregistrez le fichier en lui donnant un nom qui se termine par `.css`, comme `style.css`. Placez ce fichier `.css` dans le même dossier que votre fichier `.html`.

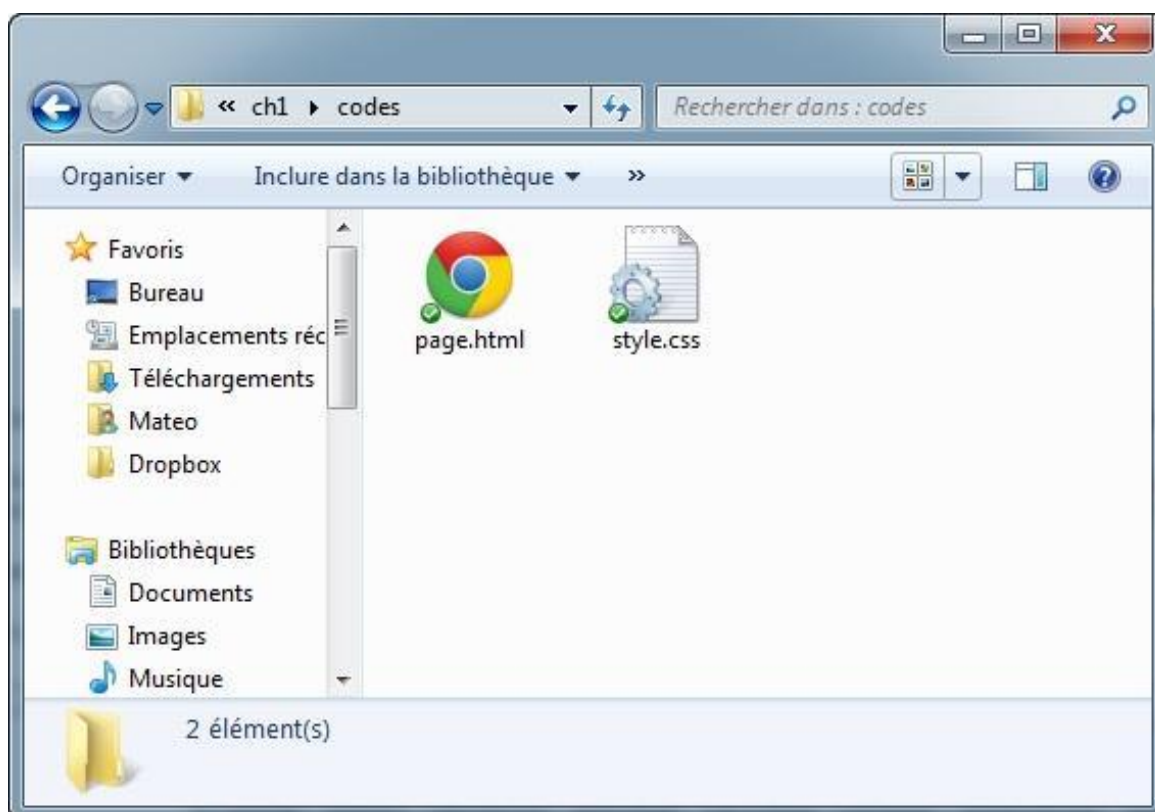
Dans Notepad++, vous devriez observer quelque chose de similaire à la figure suivante.



Fichiers HTML

et CSS dans Notepad++

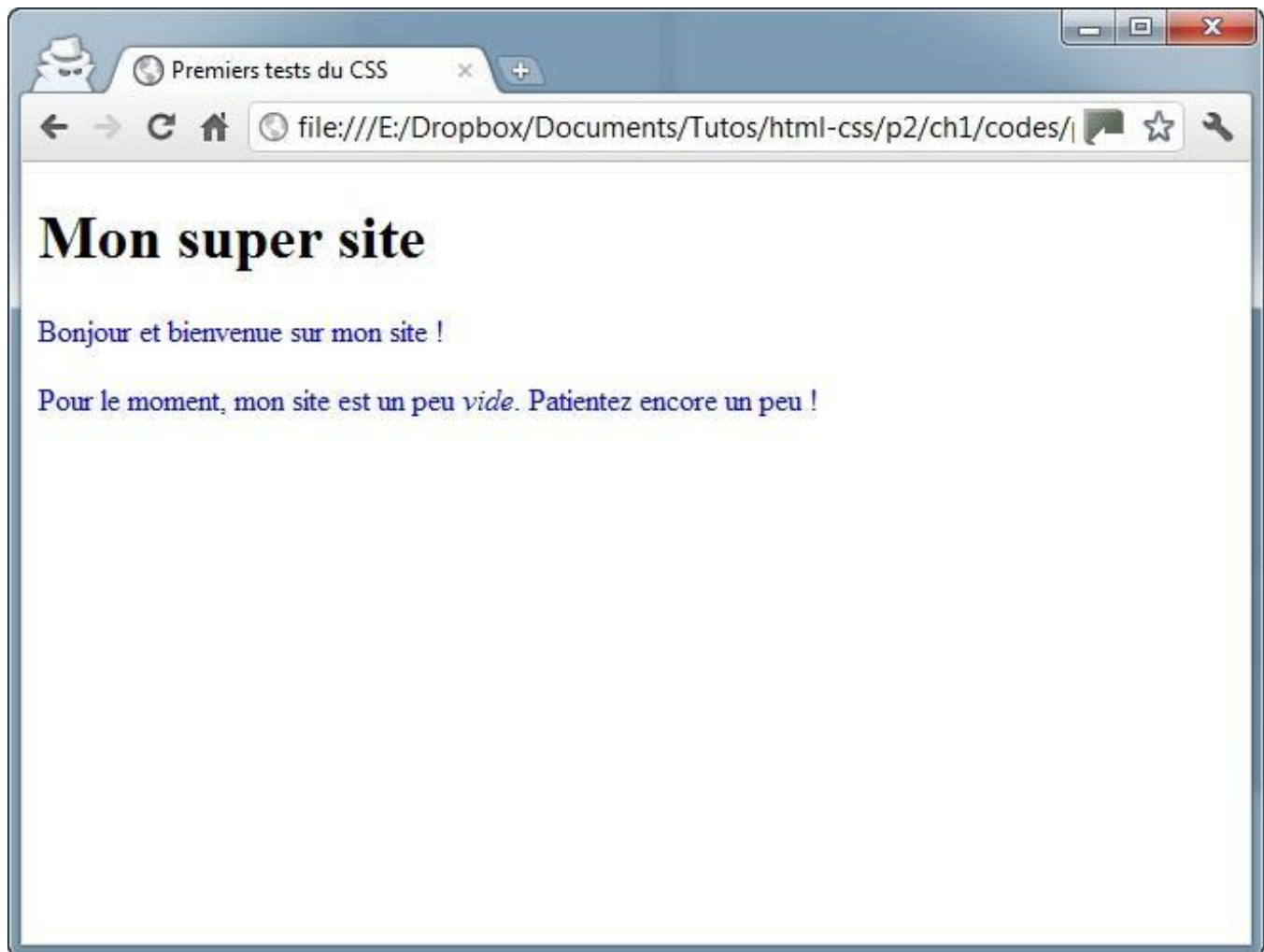
Dans votre explorateur de fichiers, vous devriez les voir apparaître côte à côte. D'un côté le **.html**, de l'autre le **.css**, comme à la figure suivante.



Fichiers HTML et

CSS dans l'explorateur de fichiers

Ouvrez maintenant votre fichier `page.html` dans votre navigateur pour le tester, comme vous le faites d'habitude. Regardez, c'est magique : vos paragraphes sont écrits en bleu, comme dans la figure suivante !



texte est écrit en bleu

### Dans l'en-tête `<head>` du fichier HTML

Il existe une autre méthode pour utiliser du CSS dans ses fichiers HTML : cela consiste à insérer le code CSS directement dans une balise `<style>` à l'intérieur de l'en-tête `<head>`.

Voici comment on peut obtenir exactement le même résultat avec un seul fichier `.html` qui contient le code CSS :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <style>
      p
      {
        color: blue;
      }
    </style>
    <title>Premiers tests du CSS</title>
  </head>
  <body>
    <h1>Mon super site</h1>
    <p>Bonjour et bienvenue sur mon site !</p>
    <p>Pour le moment, mon site est un peu Patientez encore un peu !</p>
  </body>
</html>
```

Testez, vous verrez que le résultat  
est le même.



---

## Directement dans les balises (non recommandé)

Dernière méthode, à manipuler avec précaution : vous pouvez ajouter un attribut `style` à n'importe quelle balise. Vous insérez votre code CSS directement dans cet attribut :

Code : HTML

```
<!DOCTYPE
html>
<html>
  <head
  >
    <meta charset="utf-8" />
    <title>Premiers tests du CSS</title>
  </head>
  >
  <body
  >
    <h1>Mon super site</h1>
    <p style="color: blue;">Bonjour et bienvenue sur mon site
!</p>
  >
    <p>Pour le moment, mon site est un peu
PatienteZ encore un peu !</p>
  </body>
</html>
```

Cette fois, seul le texte du premier paragraphe, dont la balise contient le code CSS, sera coloré en bleu.



## Quelle méthode choisir ?

### Appliquer un style : sélectionner une balise

Maintenant que nous savons où placer le code CSS, intéressons-nous au code.

Code : CSS

```
p
{
    color: blue;
}
```

Dans un code CSS comme celui-ci, on trouve trois éléments différents :

- **Des noms de balises ou selecteurs** : on écrit les noms des balises dont on veut modifier l'apparence. Par exemple, si je veux modifier l'apparence de tous les paragraphes `<p>`, je dois écrire `p`.
- **Des propriétés CSS** : les « effets de style » de la page sont rangés dans des propriétés. Il y a par exemple la propriété `color` qui permet d'indiquer la couleur du texte, `font-size` qui permet d'indiquer la taille du texte, etc. Il y a beaucoup de propriétés CSS et,
- **Les valeurs** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété `color`, il faut indiquer le nom de la couleur. Pour `font-size`, il faut indiquer quelle taille on veut, etc.

Schématiquement, une feuille de style CSS ressemble donc à cela :

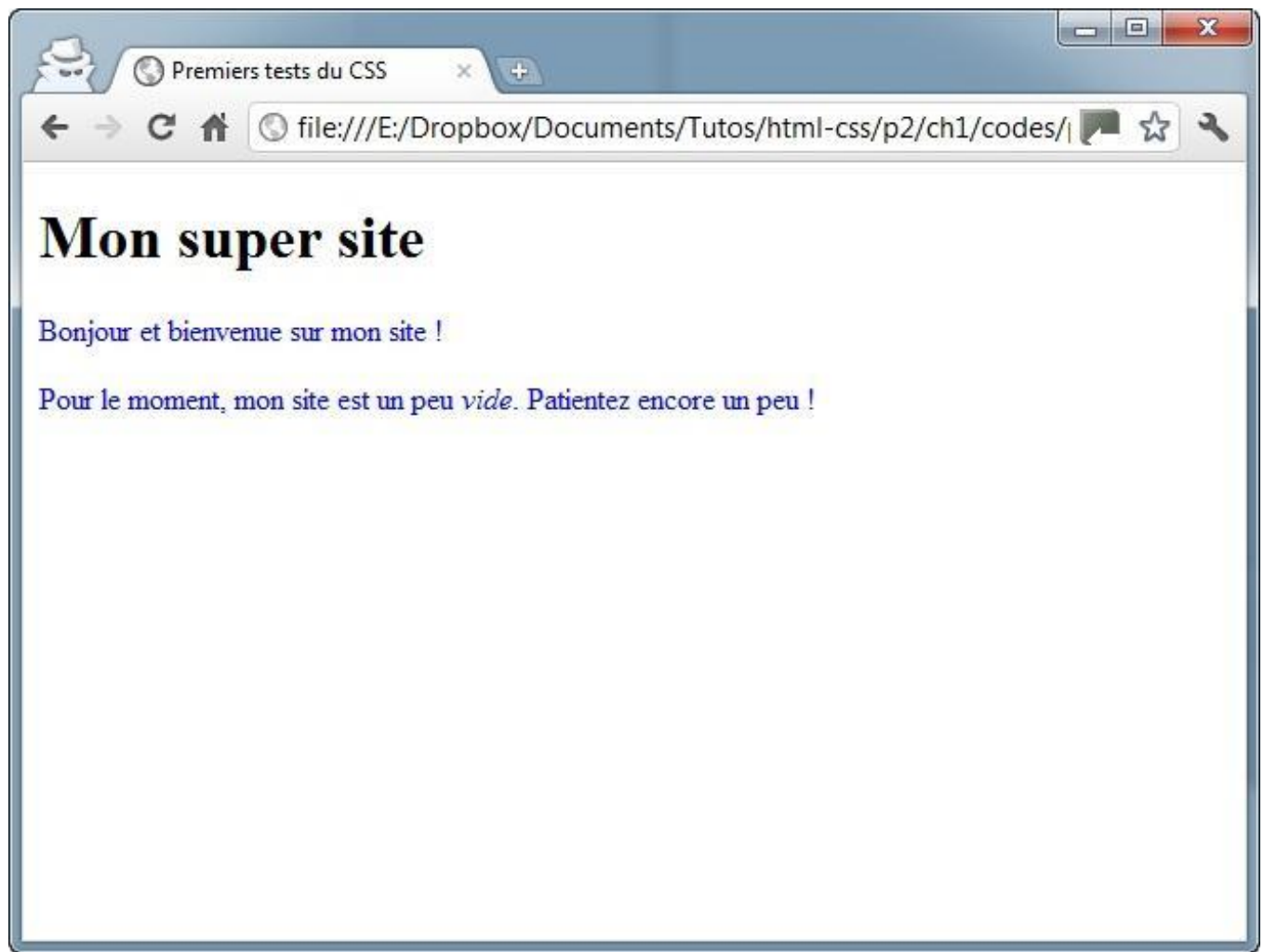
Code : CSS

**balise1**

```
{  
    propriete1: valeur1;  
    propriete2: valeur2;  
    propriete3: valeur3;  
}
```

**balise2**

```
{  
    propriete1: valeur1;  
    propriete2: valeur2;  
    propriete3: valeur3;  
    propriete4: valeur4;  
}
```



Paragraphe écrit en bleu

Essayez de changer le nom de la balise affectée par le code CSS. Par exemple, si j'écris `h1`, c'est le titre qui sera écrit en bleu. Modifiez votre fichier `style.css` comme ceci :

Code : CSS

```
h1
{
  color: blue;
}
```

## Des commentaires dans du CSS

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, par exemple pour vous retrouver dans un fichier CSS.

Notez qu'il est possible de créer plusieurs fichiers CSS pour votre site si vous ressentez le besoin de séparer un peu votre code CSS.

Donc, pour faire un commentaire, c'est facile . Tapez `/*`, suivi de votre commentaire, puis `*/` pour terminer votre commentaire. Vos commentaires peuvent être écrits sur une ou plusieurs lignes . Par exemple :

Code : CSS

```
/*
style.css
-----

Par Mathieu Nebra
*/

p
{
    color: blue; /* Les paragraphes seront en bleu */
}
```

## Appliquer un style : class et id

Ce que je vous ai montré jusqu'ici a quand même un défaut : cela implique par exemple que tous les paragraphes possèdent la même présentation (ici, ils seront donc tous écrits en bleu).

Comment faire pour que certains paragraphes seulement soient écrits d'une manière différente ? On pourrait placer le code CSS dans un attribut **style** sur la balise que l'on vise (c'est la technique que je vous ai présentée un peu plus tôt) mais, comme je vous l'ai dit, ce n'est pas recommandé (il vaut mieux utiliser un fichier CSS externe).

Pour résoudre le problème, on peut utiliser ces attributs spéciaux *qui fonctionnent sur toutes les balises* :

- l'attribut **class** ;
- l'attribut **id**.

Que les choses soient claires dès le début : les attributs **class** et **id** sont quasiment identiques.

Il y a seulement une petite différence que je vous dévoilerai plus bas.

Pour le moment, et pour faire simple, on ne va s'intéresser qu'à l'attribut **class**.

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

Code : HTML

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

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

En fait, vous devez écrire un nom qui sert à identifier la balise. Ce que vous voulez, du moment que le nom commence par une lettre.

Par exemple, je vais associer la classe **introduction** à mon premier paragraphe :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Premiers tests du CSS</title>
  </head>
  <body>
    <h1>Mon super site</h1>

    <p class="introduction">Bonjour et bienvenue sur mon site
!</p>
    <p>Pour le moment, mon site est un peu .
      Patientez encore un peu !</p>
  </body>
</html>
```

Maintenant que c'est fait, votre paragraphe est identifié. Il a un nom : **introduction**. Vous allez pouvoir réutiliser ce nom dans le fichier CSS pour dire : « *Je veux que seules les balises qui ont comme nom 'introduction' soient affichées en bleu* ».

Pour faire cela en CSS, indiquez le nom de votre classe en commençant par un point, comme ci-dessous :

Code : CSS

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

Et l'attribut **id** alors ?

Lui, il fonctionne exactement de la même manière que **class**, à un détail près : il ne peut être utilisé *qu'une fois* dans le code.

Quel intérêt ? Il y en a assez peu pour tout vous dire, cela vous sera utile si vous faites du JavaScript plus tard pour reconnaître certaines balises. D'ailleurs, nous avons déjà vu l'attribut **id** dans le chapitre sur les formulaires. En pratique, nous ne mettrons des **id** que sur des éléments qui sont uniques dans la page, comme par exemple le logo :

Code : HTML

```

```

Si vous utilisez des **id**, lorsque vous définirez leurs propriétés dans le fichier CSS, il faudra faire précéder le nom de l'**id** par un dièse (#) :

Code : CSS

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

## Les balises universelles

Il arrivera parfois que vous ayez besoin d'appliquer une **class** (ou un **id**) à certains mots qui, à l'origine, ne sont pas entourés par des balises.

En effet, le problème de **class**, c'est qu'il s'agit d'un attribut. Vous ne pouvez donc en mettre que sur une balise. Si, par exemple, je veux modifier uniquement « bienvenue » dans le paragraphe suivant :

Code : HTML

```
<p>Bonjour et bienvenue sur mon site !</p>
```



Cela serait facile à faire s'il y avait une balise autour de « bienvenue » mais, malheureusement il n'y en a pas. Par chance, on a inventé... la balise-qui-ne-sert-à-rien.

En fait, on a inventé deux balises dites **universelles**, qui n'ont aucune signification particulière (elles n'indiquent pas que le mot est important, par exemple). Il y a une différence minime (mais significative !) entre ces deux balises :

- **<span> </span>** : c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement. Les balises **<strong>** et **<em>** sont de la même famille. Cette balise s'utilise donc au milieu d'un paragraphe et c'est celle dont nous allons nous servir pour colorer « bienvenue ».
- **<div> </div>** : c'est une balise de type **block**, qui entoure un bloc de texte. Les balises **<p>**, **<h1>**, etc. sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau « bloc » dans la page et provoquent donc obligatoirement un retour à la ligne. **<div>** est une balise fréquemment utilisée dans la construction d'un design, comme nous le verrons plus tard.

Pour le moment donc, nous allons utiliser plutôt la balise **<span>**. On la met autour de « bienvenue », on lui ajoute une classe  
(du nom qu'on veut), on crée le CSS et c'est gagné !

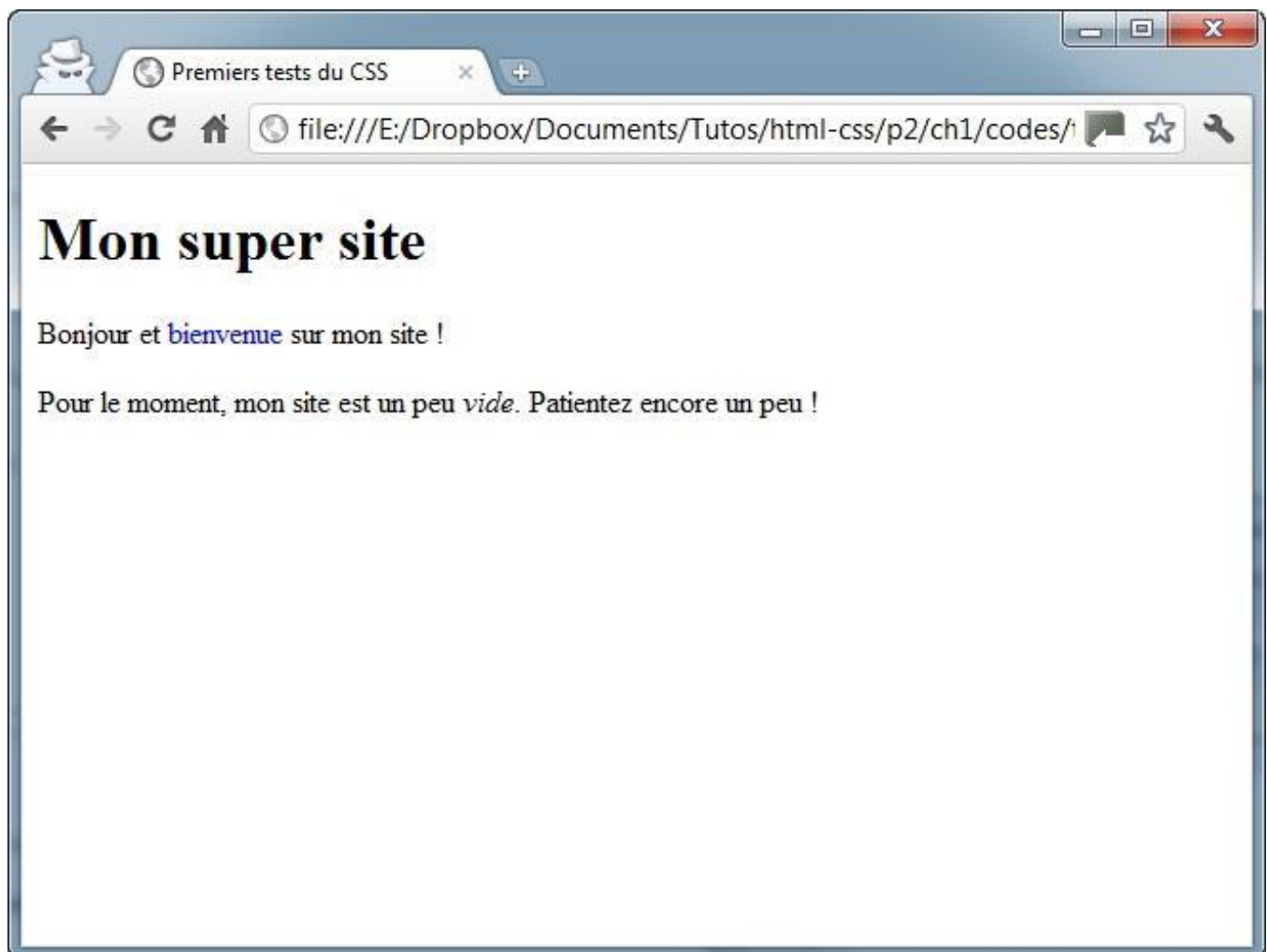
Code : HTML

```
<p>Bonjour et <span class="salutations">bienvenue</span> sur mon  
site !</p>
```

Code : CSS

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

Vous pouvez voir le résultat à la figure suivante.



## Appliquer un style : les sélecteurs avancés

En CSS, le plus difficile est de savoir cibler le texte dont on veut changer la forme. Pour cibler les éléments de la page à modifier, on utilise ce qu'on appelle des **sélecteurs**.

*\* : sélecteur universel*

**Code : CS**

**S**

\*

{

}

Sélectionne toutes les balises sans exception. On l'appelle le sélecteur universel.

### *A B : une balise contenue dans une autre*

Code : CSS

```
h3 h5
{
}
```

Sélectionne toutes les balises `<h5>` situées à l'intérieur d'une balise `<h3>`. Notez qu'il n'y a pas de virgule entre les deux noms de balises.

Exemple de code HTML correspondant :

Code : HTML

```
<h3>Titre avec <h5>texte important</h5></h3>
```

### *A + B : une balise qui en suit une autre*

Code : CSS

```
h3 + p
{
}
```

Sélectionne la première balise `<p>` située après un

titre `<h3>`. Exemple :

Code : HTML

```
<h3>Titre</h3>
<p>Paragraphe</p>
```

*A[attribut] : une balise qui possède un attribut*

Code : CSS

```
a[title]
{
}
```

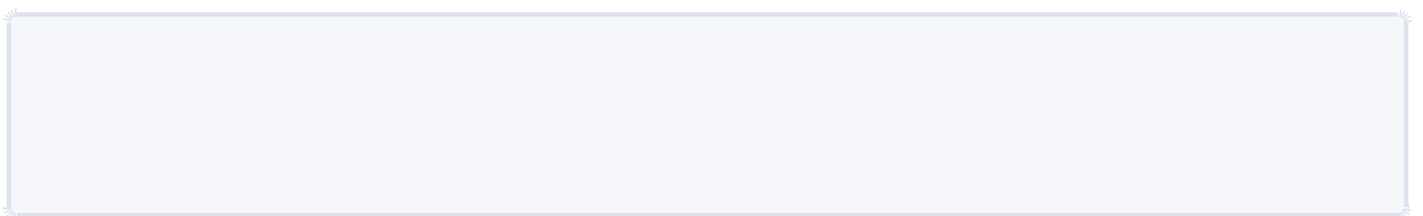
Sélectionne tous les liens `<a>` qui possèdent un attribut `title`. Exemple :

Code : HTML

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

## D'autres sélecteurs existent !

Je ne vous ai présenté ici qu'une partie des sélecteurs CSS mais sachez qu'il en existe beaucoup d'autres. Si vous voulez une liste complète, vous pouvez vous renseigner directement à la source : sur le [site du W3C](#) ! C'est très complet.



## Formatage du texte

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...

### 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
- Indiquer une **taille relative** : en pourcentage cette technique a l'avantage d'être plus souple.

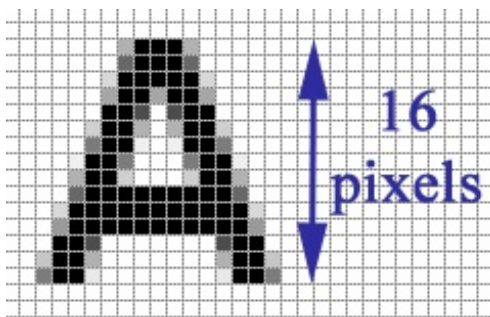
### Une taille absolue

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

Code : CSS

```
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) :

Code : CSS

```
p
{
    font-size: 14px; /* Paragraphes de 14 pixels */
}
h1
{
    font-size: 40px; /* Titres de 40 pixels */
}
```

## La police

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

Code : CSS

```
p
{
  font-family: verdana;
}
```

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 :

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.



Ainsi, si j'écris :

Code : CSS

```
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*) ».

## Italique, gras, souligné...

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

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).

Code : CSS

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

## Mettre 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). Voici par exemple comment écrire les titres en gras :

Code : CSS

```
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.
- **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** :

Code : CSS

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

Et le résultat est visible à la figure suivante.



## 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 :

Code : CSS

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

p
{
    text-align: justify;
}

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

## En résumé

- On modifie la taille du texte avec la propriété CSS **font-size**. On peut indiquer la taille en pixels (16px), en pourcentage (110%), etc.
- On change la police du texte avec **font-family**. Attention, seules quelques polices sont connues par tous les ordinateurs.
- 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**.

# La couleur et le fond

## Couleur du texte

Vous connaissez déjà la propriété qui permet de modifier la couleur du texte : il s'agit de **color**. Nous allons nous intéresser aux différentes façons d'indiquer la couleur, car il y en a plusieurs.

Pour passer tous les titres en marron, on peut donc écrire :

Code : CSS

```
h1
{
    color: maroon;
}
```

## Couleur de fond

Pour indiquer une couleur de fond, on utilise la propriété CSS **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.

Pour indiquer la couleur de fond de la page web, il faut travailler sur la balise **<body>**. Eh oui, **<body>** correspond à l'ensemble de la page web, c'est donc en modifiant sa couleur de fond que l'on changera la couleur d'arrière-plan de la page.

Regardez très attentivement ce fichier CSS :

Code : CSS

```
/* On travaille sur la balise body, donc sur TOUTE la page */
body
{
    background-color: black; /* Le fond de la page sera noir */
    color: white; /* Le texte de la page sera blanc */
}
```



Texte en blanc sur fond noir

## Appliquer une 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 :

Code : CSS

```
body
{
    background-image: url("neige.png");
}
```

Ce qui nous donne la figure suivante.





## Options disponibles pour l'image de fond

### *background-position : position du fond*

On peut indiquer où doit se trouver l'image de fond avec **background-position**. Cette propriété n'est intéressante que si elle est combinée avec **background-repeat: no-repeat** ; (un fond qui ne se répète pas).

Vous devez donner à **background-position** deux valeurs en pixels pour indiquer la position du fond par rapport au coin supérieur gauche de la page (ou du paragraphe, si vous appliquez le fond à un paragraphe). Ainsi, si vous tapez :

Code : CSS

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

... votre fond sera placé à 30 pixels de la gauche et à 50 pixels du haut. Il est aussi possible d'utiliser ces valeurs en anglais :

- **top** : en haut ;
- **bottom** : en bas ;
- **left** : à gauche ;
- **center** : centré ;
- **right** : à droite.

Il est possible de combiner ces mots . Par exemple, pour aligner une image en haut à droite, vous taperez :

**Code : CSS**

```
background-position: top right;
```

Ainsi, si je veux afficher un soleil en image de fond (figure suivante), en un unique exemplaire (**no-repeat**), toujours visible (**fixed**) et positionné en haut à droite (**top right**), je vais écrire ceci :

**Code : CSS**

```
body
{
    background-image: url("soleil.png");
    background-position: top right; /* Le fond sera placé en haut à
droite */
}
```



soleil placé en image de fond en haut à droite

Un

## Les bordures et les ombres

### Bordures standard

Pour **border** on peut utiliser jusqu'à trois valeurs pour modifier l'apparence de la bordure :

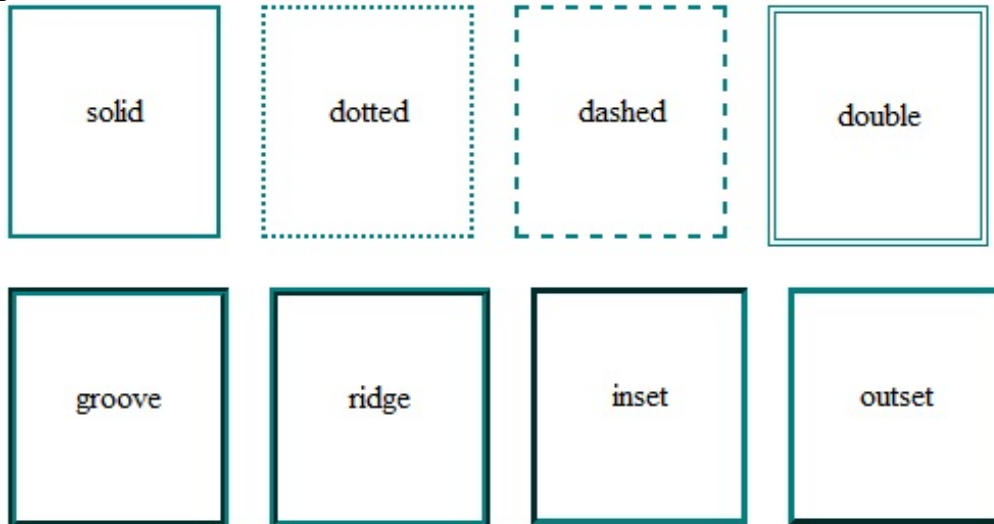
- **La largeur** : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme 2px).
- **La couleur** : c'est la couleur de votre bordure. Utilisez, comme on l'a appris, soit un nom de couleur (**black**, **red**,...), soit une valeur hexadécimale (#FF0000),.
- **Le type de bordure** : là, vous avez le choix. 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.

Ainsi, pour avoir une bordure bleue, en tirets, épaisse de 3 pixels autour de mes titres, je vais écrire :

Code : CSS

```
h1
{
  border: 3px blue dashed;
}
```

La figure suivante vous présente les différents styles de bordures que vous pouvez utiliser.



## Bordures arrondies

Les bordures arrondies, c'est un peu le Saint Graal attendu par les webmasters depuis des millénaires . CSS3 est arrivé, il est enfin possible d'en créer facilement !

La propriété **border-radius** va nous permettre d'arrondir facilement les angles de n'importe quel élément. Il suffit d'indiquer la taille (« l'importance ») de l'arrondi en pixels :

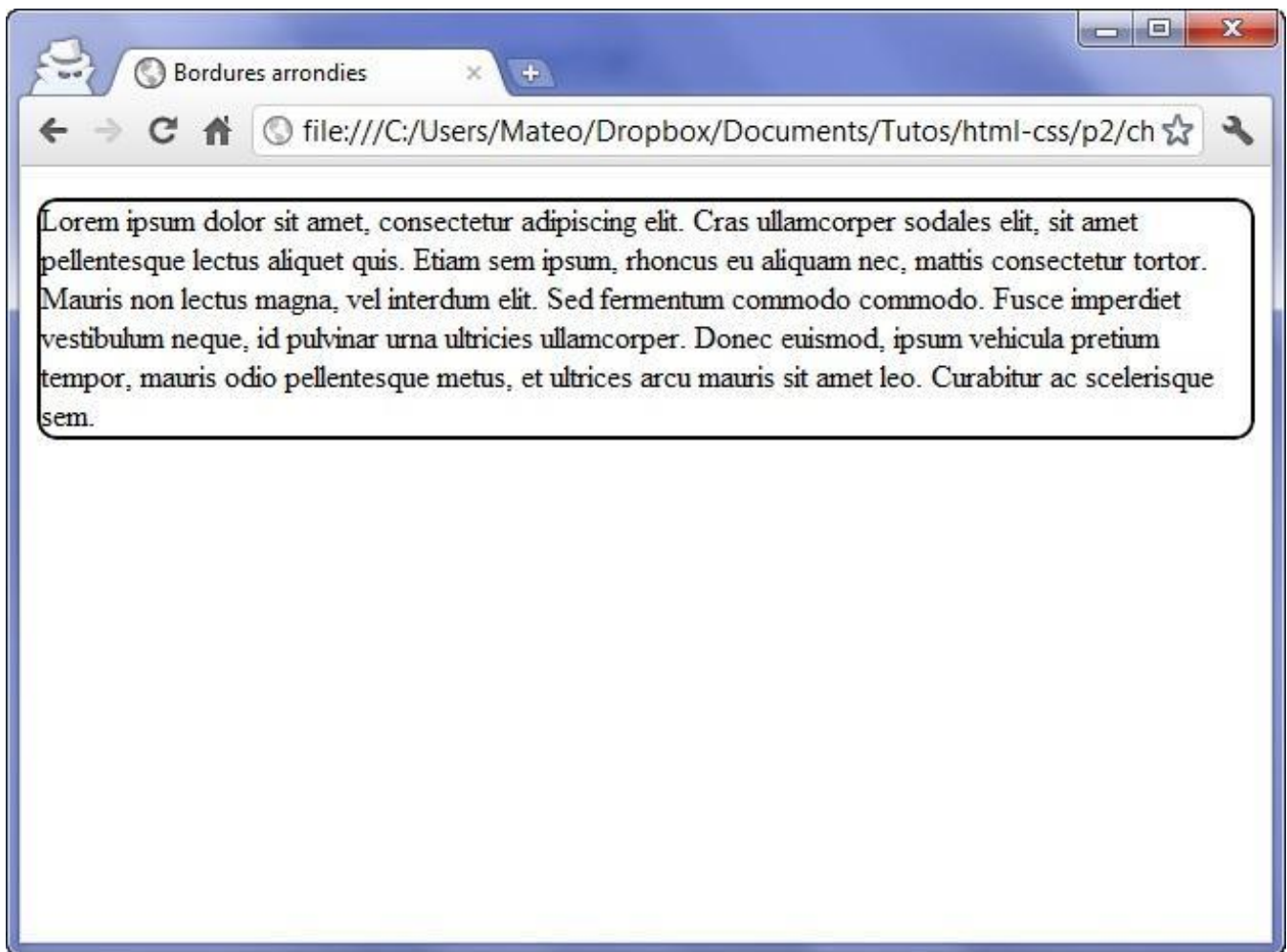
Code : CSS

```
p
{
    border-radius: 10px 5px 10px 5px;
}
```

Les valeurs correspondent aux angles suivants dans cet ordre :

1. en haut à gauche ;
2. en haut à droite ;
3. en bas à droite ;
4. en bas à gauche.

comme sur la figure suivante.



Des

bordures arrondies

## **MENU HORIZONTAL ET VERTICAL**

### **1. MENU HORIZONTAL**

### **2. MENU VERTICAL**

**FIN !!!**