



## II. Les balises :

Les pages HTML sont remplies de ce qu'on appelle des balises. Celles-ci sont invisibles à l'écran pour vos visiteurs, mais elles permettent à l'ordinateur de comprendre ce qu'il doit afficher.

Les balises se repèrent facilement. Elles sont entourées de « chevrons », c'est-à-dire des symboles<et>, comme ceci :<balise>. À quoi est-ce qu'elles servent ? Elles indiquent la nature du texte qu'elles encadrent. Elles veulent dire par exemple : « Ceci est le titre de la page », « Ceci est une image », « Ceci est un paragraphe de texte », etc. On distingue deux types de balises : les balises en paires et les balises orphelines.

### **Les balises en paires**

Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Voici à quoi elles ressemblent :

```
<titre>Ceci est un titre</titre>
```

On distingue une balise ouvrante (<titre>) et une balise fermante (</titre>) qui indique que le titre se termine. Cela signifie pour l'ordinateur que tout ce qui n'est *pas* entre ces deux balises... n'est pas un titre.

```
Ceci n'est pas un titre <titre>Ceci est un titre</titre> Ceci n'est pas un titre
```

### **Les balises orphelines**

Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis (par exemple une image). Il n'est pas nécessaire de délimiter le début et la fin de l'image, on veut juste dire à l'ordinateur « Insère une image ici ».

Une balise orpheline s'écrit comme ceci :

```
<image />
```

### **Les attributs**

Les attributs sont un peu les options des balises. Ils viennent les compléter pour donner des informations supplémentaires. L'attribut se place après le nom de la balise ouvrante et a le plus souvent une valeur, comme ceci :

```
<balise attribut="valeur">
```

À quoi cela sert-il ? Prenons la balise<image />que nous venons de voir. Seule, elle ne sert pas à grand chose. On pourrait rajouter un attribut qui indique le nom de l'image à afficher :

```
<image nom="photo.jpg" />
```

L'ordinateur comprend alors qu'il doit afficher l'image contenue dans le fichierphoto.jpg.

Dans le cas d'une balise fonctionnant « par paire », on ne met les attributs que dans la balise ouvrante et pas dans la balise fermante. Par exemple, ce code indique que la citation est de Neil Armstrong et qu'elle date du 21 Juillet 1969 :

```
<citation auteur="Neil Armstrong" date="21/07/1969">
```

C'est un petit pas pour l'homme, mais un bond de géant pour l'humanité.  
</citation>

### III. Structure d'une page html :

Dans Notepad++

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Titre</title>
6   </head>
7
8   <body>
9
10  </body>
11 </html>
```

Une fois enregistré « essai2.html », par exemple la coloration syntaxique se met en place.

Vous noterez que les balises s'ouvrent et se ferment dans un ordre précis. Par exemple, la balise `<html>` est la première que l'on ouvre et c'est aussi la dernière que l'on ferme (tout à la fin du code, avec `</html>`). *Les balises doivent être fermées dans le sens inverse de leur ouverture.* Un exemple :

- `<html><body></body></html>`: **correct**. Une balise qui est ouverte à l'intérieur d'une autre doit aussi être fermée à l'intérieur.
- `<html><body></html></body>`: **incorrect**, les balises s'entremêlent.

#### Le doctype

```
<!DOCTYPE html>
```

La toute première ligne s'appelle le **doctype**. Elle est indispensable car c'est elle qui indique qu'il s'agit bien d'une page web HTML.

Ce n'est pas vraiment une balise comme les autres (elle commence par un point d'exclamation). Vous pouvez considérer que c'est un peu l'exception qui confirme la règle.

#### La balise `</html>`

```
<html>
```

```
</html>
```

C'est la balise principale du code. Elle englobe tout le contenu de votre page. Comme vous pouvez le voir, la balise fermante `</html>` se trouve tout à la fin du code !

#### L'en-tête `<head>` et le corps `<body>`

Une page web est constituée de deux parties :

- L'en-tête `<head>`: cette section donne quelques informations générales sur la page comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc. Cette section est généralement assez

courte. Les informations que contient l'en-tête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes !

- Le corps<body>: c'est là que se trouve la partie principale de la page. Tout ce que nous écrirons ici sera affiché à l'écran. C'est à l'intérieur du corps que nous écrirons la majeure partie de notre code.

Pour le moment, le corps est vide (nous y reviendrons plus loin). Intéressons-nous par contre aux deux balises contenues dans l'en-tête...

### L'encodage (charset)

```
<meta charset="utf-8" />
```

Cette balise indique l'encodage utilisé dans votre fichier.html.

Sans rentrer dans les détails, car cela pourrait vite devenir compliqué, l'encodage indique la façon dont le fichier est enregistré. C'est lui qui détermine comment les caractères spéciaux vont s'afficher (accents, idéogrammes chinois et japonais, caractères arabes, etc.).

Il y a plusieurs techniques d'encodage portant des noms bizarres et utilisées en fonction des langues : ISO-8859-1, OEM 775, Windows-1253... Une seule cependant devrait être utilisée aujourd'hui autant que possible : UTF-8. Cette méthode d'encodage permet d'afficher sans aucun problème pratiquement tous les symboles de toutes les langues de notre planète ! C'est pour cela que j'ai indiqué utf-8 dans cette balise.

Il ne suffit pas de *dire* que votre fichier est en UTF-8. Il faut aussi que votre fichier soit bien *enregistré* en UTF-8. C'est heureusement le cas désormais par défaut dans la plupart des éditeurs de texte.

Si les accents s'affichent mal par la suite, c'est qu'il y a un problème avec l'encodage. Vérifiez que la balise meta indique bien UTF-8 et que votre fichier est enregistré en UTF-8 (sous Sublime Text, allez dans le menu File > Save with Encoding > UTF-8 pour vous assurer que votre fichier est enregistré en UTF-8.).

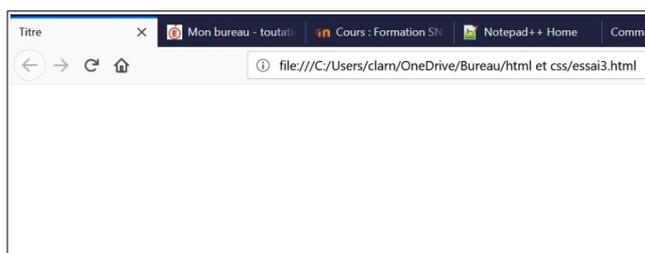
### Le titre principal de la page

```
<title>
```

C'est le titre de votre page, probablement l'élément le plus important ! Toute page doit avoir un titre qui décrit ce qu'elle contient.

Il est conseillé de garder le titre assez court (moins de 100 caractères en général).

Le titre ne s'affiche pas dans votre page mais en haut de celle-ci (souvent dans l'onglet du navigateur). Enregistrez votre page web et ouvrez-la dans votre navigateur. Vous verrez que le titre s'affiche dans l'onglet, comme sur la figure suivante.



Le titre de la page apparaît en haut du navigateur

Il faut savoir que le titre apparaît aussi dans les résultats de recherche, comme sur Google (figure suivante).

Le titre de la page apparaît dans les recherches Google

Autant vous dire que bien choisir son titre est important !

#### IV. Organiser son texte :

Nous allons voir successivement dans ce chapitre :

- comment rédiger des paragraphes ;
- comment structurer sa page avec les titres ;
- comment donner de l'importance à certains mots de son texte ;
- comment organiser les informations sous forme de listes.

#### Les paragraphes

La plupart du temps, lorsqu'on écrit du texte dans une page web, on le fait à l'intérieur de paragraphes. Le langage HTML propose justement la balise `<p>` pour délimiter les paragraphes.

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

- `<p>` signifie « Début du paragraphe » ;
- `</p>` signifie « Fin du paragraphe ».

Comme je vous l'ai dit au chapitre précédent, on écrit le contenu du site web entre les balises `<body></body>`. Il nous suffit donc de mettre notre paragraphe entre ces deux balises et nous aurons enfin notre première vraie page web avec du texte !

Je reprends donc exactement le même code qu'au chapitre précédent et j'y ajoute mon paragraphe :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Organiser</title>
6   </head>
7   <body>
8     <p> un premier paragraphe </p>
9   </body>
10 </html>
```

#### Sauter une ligne

En HTML, si vous appuyez sur la touche `Entrée`, cela ne crée pas une nouvelle ligne comme vous en avez l'habitude. Essayez donc ces code :

<pre> 1 &lt;!DOCTYPE html&gt; 2 &lt;html&gt; 3   &lt;head&gt; 4     &lt;meta charset="utf-8" /&gt; 5     &lt;title&gt;Organiser&lt;/title&gt; 6   &lt;/head&gt; 7   &lt;body&gt; 8     &lt;p&gt; un premier paragraphe 9       une ligne 10      deux lignes 11      trois lignes 12    &lt;/p&gt; 13  &lt;/body&gt; 14 &lt;/html&gt; 15 </pre>	<pre> 1 &lt;!DOCTYPE html&gt; 2 &lt;html&gt; 3   &lt;head&gt; 4     &lt;meta charset="utf-8" /&gt; 5     &lt;title&gt;Organiser&lt;/title&gt; 6   &lt;/head&gt; 7   &lt;body&gt; 8     &lt;p&gt; un premier paragraphe&lt;/br&gt; 9       une ligne&lt;/br&gt; 10      deux lignes&lt;/br&gt; 11      trois lignes&lt;/br&gt; 12    &lt;/p&gt; 13  &lt;/body&gt; 14 &lt;/html&gt; </pre>
---	--

On peut organiser son corps de page en plusieurs paragraphes

```

7 <body>
8   <p> un premier paragraphe</br>
9     une ligne</br>
10    deux lignes</br>
11    trois lignes</br>
12  </p>
13  <p> un deuxième paragraphe</br>
14    une ligne</br>
15    deux lignes</br>
16    trois lignes</br>
17  </p>
18 </body>

```

## Les titres

Lorsque le contenu de votre page va s'étoffer avec de nombreux paragraphes, il va devenir difficile pour vos visiteurs de se repérer. C'est là que les titres deviennent utiles.

En HTML, on est verni, on a le droit d'utiliser six niveaux de titres différents. Je veux dire par là qu'on peut dire « Ceci est un titre très important », « Ceci est un titre un peu moins important », « Ceci est un titre encore moins important », etc. On a donc six balises de titres différentes :

- `<h1> </h1>`: signifie « titre très important ». En général, on s'en sert pour afficher le titre de la page au début de celle-ci.
- `<h2> </h2>`: signifie « titre important ».
- `<h3> </h3>`: pareil, c'est un titre un peu moins important (on peut dire un « sous-titre » si vous voulez).
- `<h4> </h4>`: titre encore moins important.
- `<h5> </h5>`: titre pas important.
- `<h6> </h6>`: titre vraiment, mais alors là vraiment pas important du tout.

Attention : ne confondez pas avec la balise<title>! La balise<title>affiche le titre de la page dans la barre de titre du navigateur comme nous l'avons vu. Les titres<h1>et compagnie, eux, servent à créer des titres qui seront affichés *dans* la page web.

Ne choisissez pas votre balise de titre en fonction de la taille qu'elle applique au texte ! Il faut impérativement bien structurer sa page en commençant par un titre de niveau 1 (<h1>), puis un titre de niveau 2 (<h2>), etc. Il ne devrait pas y avoir de sous-titre sans titre principal !

Si vous voulez modifier la taille du texte, sachez que nous apprendrons à faire cela en CSS un peu plus tard.

Essayez de créer une page web avec des titres pour voir ce que cela donne :

```
8 <body>
9   <h1>Bienvenue sur mon site</h1>
10  <p>Mon joli site<br />
11  SNT</p>
12  <h2>Initiation au langage HTML</h2>
13  <p>Les titres, ça s'organise</p>
14  <h2>Le Langage CSS</h2>
15  <p>Coming next</br>
16  chaque chose en son temps</p>
17 </body>
```

## La mise en valeur

Au sein de vos paragraphes, certains mots sont parfois plus importants que d'autres et vous aimeriez les faire ressortir. HTML vous propose différents moyens de mettre en valeur le texte de votre page.

### Mettre un peu en valeur

Pour mettre *un peu* en valeur votre texte, vous devez utiliser la balise<em> </em>.

Son utilisation est très simple : encadrez les mots à mettre en valeur avec ces balises.

Comme vous pouvez le voir, utiliser la balise<em>a pour conséquence de mettre le texte en *italique*. En fait, c'est le navigateur qui choisit comment afficher les mots. On lui dit que les mots sont assez importants et, pour faire ressortir cette information, il change l'apparence du texte en utilisant l'italique.

### Mettre bien en valeur

Pour mettre un texte bien en valeur, on utilise la balise<strong>qui signifie « fort », ou « important » si vous préférez. Elle s'utilise exactement de la même manière que<em>:

Vous voyez sûrement le texte s'afficher en gras. Là encore, le gras n'est qu'une *conséquence*. Le navigateur a choisi d'afficher en gras les mots importants pour les faire ressortir davantage.

La balise<strong>ne signifie pas « mettre en gras » mais « important ». On pourra décider plus tard, en CSS, d'afficher les mots « importants » d'une autre façon que le gras si on le souhaite.

## Marquer le texte

La balise `<mark>` permet de faire ressortir visuellement une portion de texte. L'extrait n'est pas forcément considéré comme important mais on veut qu'il se distingue bien du reste du texte. Cela peut être utile pour faire ressortir un texte pertinent après une recherche sur votre site par exemple.

Par défaut, `<mark>` a pour effet de surligner le texte. On pourra changer l'affichage en CSS (décider de surligner dans une autre couleur, d'encadrer le texte, etc.). C'est le même principe que ce que je vous disais pour les balises précédentes : elles indiquent le *sens* des mots et non pas comment ceux-ci doivent s'afficher.

## Test sur la couleur et la taille d'un texte (juste un test, on oublie ensuite) :

Vous pouvez essayer cet exemple :

```
<body>
  <h1>Bienvenue sur mon site</h1>
  <p><FONT color="#ff0000">Mon joli site</FONT><br />
  SNT</p>
  <h2><FONT color="#00ff00"><FONT size=8>Initiation au langage html</FONT></FONT></h2>
  <p>Les <em>titres</em>, ça s'organise</p>
  <h2><FONT color="#00ff00"><FONT FACE="Calibri">Le langage CSS</FONT></FONT></h2>
  <p>Coming next</br>
  <strong>chaque chose en son temps</strong></p>
</body>
```

Ca fonctionne, alors pourquoi l'oublier ? Il est assez pénible de devoir définir toutes les propriétés d'un texte, de devoir le recopier dans chaque page du site, un jour vous voudrez peut-être modifier le design entier de votre site. Tout se fera dans le CSS pour l'ensemble de votre site, reprenez :

- le HTML définit le fond (contenu, logique des éléments) ;
- le CSS définit la forme (apparence).

On oublie donc.

## Les listes :

Une liste non ordonnée ressemble à ceci :

- Web
- Microbit
- Photo

C'est un système qui nous permet de créer une liste d'éléments sans notion d'ordre (il n'y a pas de «premier» ni de « dernier »). Créer une liste non ordonnée est très simple. Il suffit d'utiliser la balise<ul>que l'on referme un peu plus loin avec</ul>.

Commencez donc à taper ceci :

```
<ul>
  <li>Web</li>
  <li>Microbit</li>
  <li>Photo</li>
</ul>
```

Retenez donc ces deux balises :

- <ul></ul>délimite toute la liste ;
- <li></li>délimite un élément de la liste (une puce).

### Liste ordonnée

Une liste ordonnée fonctionne de la même façon, seule une balise change : il faut remplacer<ul></ul>par<ol></ol>.

À l'intérieur de la liste, on ne change rien : on utilise toujours des balises<li></li>pour délimiter les éléments.

## V. Des liens :

### Un lien vers une autre page de son site

Créer deux pages (« index.html » et « web.html »), que vous enregistrez dans le même dossier.

Page Index	Page web
<pre>1 &lt;!DOCTYPE html&gt; 2 &lt;html&gt; 3   &lt;head&gt; 4     &lt;meta charset="utf-8" /&gt; 5     &lt;title&gt;Liens&lt;/title&gt; 6   &lt;/head&gt; 7   &lt;body&gt; 8     &lt;a href="web.html"&gt;Mon lien vers la page "web"&lt;/a&gt; 9   &lt;/body&gt; 10 &lt;/html&gt; 11</pre>	<pre>1 &lt;!DOCTYPE html&gt; 2 &lt;html&gt; 3   &lt;head&gt; 4     &lt;meta charset="utf-8" /&gt; 5     &lt;title&gt;Liens&lt;/title&gt; 6   &lt;/head&gt; 7   &lt;body&gt; 8     &lt;a href="index.html"&gt;Mon lien vers la page "index"&lt;/a&gt; 9   &lt;/body&gt; 10 &lt;/html&gt;</pre>

### Un lien vers un autre site

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Liens</title>
6   </head>
7   <body>
8     <a href="http://www.google.fr">Mon lien externe</a>
9   </body>
10 </html>
```

### Une ancre

Une ancre est une sorte de point de repère que vous pouvez mettre dans vos pages HTML lorsqu'elles sont très longues.

En effet, il peut alors être utile de faire un lien amenant plus bas dans la même page pour que le visiteur puisse sauter directement à la partie qui l'intéresse.

Pour créer une ancre, il suffit de rajouter l'attribut *id* à une balise qui va alors servir de repère. Ce peut être n'importe quelle balise, un titre par exemple.

Utilisez l'attribut *id* pour donner un nom à l'ancre. Cela nous servira ensuite pour faire un lien vers cette ancre. Par exemple :

```
<h2 id="mon_ancre">Titre</h2>
```

Ensuite, il suffit de créer un lien comme d'habitude, mais cette fois l'attribut *href* contiendra un dièse (#) suivi du nom de l'ancre. Exemple :

```
<a href="#mon_ancre">Aller vers l'ancre</a>
```

Normalement, si vous cliquez sur le lien, cela vous amènera plus bas dans la même page (à condition que la page comporte suffisamment de texte pour que les barres de défilement se déplacent automatiquement).

```
<body>
  <h1>Bienvenue sur mon site</h1>
  <ul>
    <li><a href="#ancre1">chapitre 1</a></li>
    <li><a href="#ancre2">chapitre 2</a></li>
    <li><a href="#ancre3">chapitre 3</a></li>
    <li><a href="#ancre4">chapitre 4</a></li>
    <li><a href="#ancre5">chapitre 5</a></li>
  </ul>

  <h2 id="ancre1">Chapitre 1</h2>
  <p> bla bla </br>
    bla bla </br>
    bla bla </br>
</p>
  <h2 id="ancre2">Chapitre 2</h2>
  <p> bla bla </br>
    bla bla </br>
    bla bla </br>
</p>
  <h2 id="ancre3">Chapitre 3</h2>
  <p> bla bla </br>
    bla bla </br>
    bla bla </br>
</p>
```

Il est souvent intéressant de mettre dans chaque chapitre une ancre de retour en haut de page.

### Lien vers une ancre située dans une autre page

L'idée, c'est de faire un lien qui ouvre une autre page ET qui amène directement à une ancre située plus bas sur cette page.

En pratique c'est assez simple à faire : il suffit de taper le nom de la page, suivi d'un dièse (#), suivi du nom de l'ancre.

Par exemple : <a href="page2.html#ancre\_dans\_la\_page\_2">

### Un lien qui ouvre une nouvelle fenêtre (target) avec infobulle (title)

Sur notre page 1

<a href="web.html" title="On change" target="\_blank">Mon lien vers la page "web"</a>

### Un lien pour envoyer un e-mail

Si vous voulez que vos visiteurs puissent vous envoyer un e-mail, vous pouvez utiliser des liens de type mailto. Rien ne change au niveau de la balise, vous devez simplement modifier la valeur de l'attribut href comme ceci :

<p><a href="mailto:votrenom@bidule.com">Envoyez-moi un e-mail !</a></p>

### Un lien pour télécharger un fichier

Beaucoup d'entre vous se demandent comment cela se passe pour le téléchargement d'un fichier... En fait, il faut procéder exactement comme si vous faisiez un lien vers une page web, mais en indiquant cette fois le nom du fichier à télécharger.

Par exemple, supposez que vous vouliez faire téléchargermonfichier.zip. Placez simplement ce fichier dans le même dossier que votre page web (ou dans un sous-dossier) et faites un lien vers ce fichier :

<p><a href="monfichier.zip">Télécharger le fichier</a></p>

## VI. Les images :

### Il existe un format adapté à chaque image

voici quel format adopter en fonction de l'image que vous avez :

- **Une photo** : utilisez un JPEG.
- **N'importe quel graphique avec peu de couleurs** (moins de 256) : utilisez un PNG 8 bits ou éventuellement un GIF.
- **N'importe quel graphique avec beaucoup de couleurs** : utilisez un PNG 24 bits.
- **Une image animée** : utilisez un GIF animé.

### Les erreurs à éviter

#### Bannissez les autres formats

Les autres formats non cités ici, comme le format BITMAP (\*.bmp) sont à bannir car bien souvent ils ne sont pas compressés, donc trop gros. Ils ne sont pas du tout adaptés au Web. On peut en mettre sur son site mais le chargement sera vraiment *extrêmement* long !

#### Choisissez bien le nom de votre image

Si vous voulez éviter des problèmes, prenez l'habitude d'enregistrer vos fichiers avec des noms en minuscules, sans espace ni accent, par exemple : mon\_image.png.

Vous pouvez remplacer les espaces par le caractère *underscore* (« \_ »).

## Insérer une image

Revenons maintenant au code HTML pour découvrir comment placer des images dans nos pages web.

### Insertion d'une image

La balise qui va nous permettre d'insérer une image est `<img />`!

C'est une balise de type **orpheline** (comme `<br />`). Cela veut dire qu'on n'a pas besoin de l'écrire en deux exemplaires comme la plupart des autres balises que nous avons vues jusqu'ici. En effet, nous n'avons pas besoin de délimiter une portion de texte, nous voulons juste insérer une image à un endroit précis.

La balise doit être accompagnée de deux attributs obligatoires :

- `src` : il permet d'indiquer où se trouve l'image que l'on veut insérer. Vous pouvez soit mettre un chemin absolu (ex. : `http://www.site.com/bateau.png`), soit mettre le chemin en relatif (ce qu'on fait le plus souvent). Ainsi, si votre image est dans un sous-dossier `images`, vous devrez taper :  
`src="images/bateau.png"`
- `alt` : cela signifie « texte alternatif ». On doit *toujours* indiquer un texte alternatif à l'image, c'est-à-dire un court texte qui décrit ce que contient l'image. Ce texte sera affiché à la place de l'image si celle-ci ne peut pas être téléchargée (cela arrive), ou dans les navigateurs de personnes handicapées (non-voyants) qui ne peuvent malheureusement pas « voir » l'image. Cela aide aussi les robots des moteurs de recherche pour les recherches d'images. Pour le bateau, on mettrait par exemple :  
`alt="Un bateau"`.

Les images doivent se trouver obligatoirement à l'intérieur d'un paragraphe (`<p></p>`). Voici un exemple d'insertion d'image :

```
<p>  
    Un joli bateau:<br />  
      
</p>
```

### Ajouter une infobulle

L'attribut permettant d'afficher une bulle d'aide est le même que pour les liens : il s'agit de `title`. Cet attribut est facultatif (contrairement à `alt`).

Voici ce que cela peut donner :

```
<p>  
    Un joli bateau:<br />  
      
</p>
```

Survolez la photo avec la souris pour voir l'infobulle apparaître.

## Miniature cliquable

Si votre image est très grosse, il est conseillé d'en afficher la miniature sur votre site. Ajoutez ensuite un lien sur cette miniature pour que vos visiteurs puissent afficher l'image en taille originale.

De nombreux sites permettent de redimensionner des images, comme [ResizImage.net](http://ResizImage.net) par exemple.

La miniature et son image d'origine

Je les place toutes les deux dans un dossier appelé par exemple images. J'affiche la version bateau\_mini.jpg sur ma page et je fais un lien vers bateau.jpg pour que l'image agrandie s'affiche lorsqu'on clique sur la miniature.

Voici le code HTML que je vais utiliser pour cela :

```
<p>
    Vous souhaitez voir l'image dans sa taille d'origine ? Cliquez
dessus !<br />
    <a href="img/bateau.jpg"></a>
</p>
```

Nous savons maintenant faire plein de choses, reste à appliquer des styles.

## VII. Le CSS :

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.

Dans le *head* de vos pages insérez la ligne suivante :

```
<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 et copiez-y ce bout de code CSS .

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

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, puis ouvrez votre fichier.html dans un navigateur.

Maintenant que nous savons où placer le code CSS, intéressons-nous de plus près à ce code. Je vous ai donné, sans vous l'expliquer, un premier bout de code CSS :

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

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

- **Des noms de balises** : 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 propriétécolorqui permet d'indiquer la couleur du texte,font-sizequi permet d'indiquer la taille du texte, etc. Il y a beaucoup de propriétés CSS et, comme je vous l'ai dit, je ne vous obligerai pas à les connaître toutes par cœur.
- **Les valeurs** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriétécolor, il faut indiquer le nom de la couleur. Pourfont-size, il faut indiquer quelle taille on veut, etc.

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

```
balise1
{
    propriete1: valeur1;
    propriete2: valeur2;
    propriete3: valeur3;
}
```

```
balise2
{
    propriete1: valeur1;
    propriete2: valeur2;
    propriete3: valeur3;
    propriete4: valeur4;
}
```

```
balise3
```

```
{  
    propriete1: valeur1;  
}
```

Vous repérez dans cet extrait de code les balises, propriétés et valeurs dont je viens de vous parler.

Comme vous le voyez, on écrit le nom de la balise (par exemple h1) et on ouvre des accolades pour, à l'intérieur, mettre les propriétés et valeurs que l'on souhaite. On peut mettre autant de propriétés que l'on veut à l'intérieur des accolades. Chaque propriété est suivie du symbole « deux-points » ( : ) puis de la valeur correspondante. Enfin, chaque ligne se termine par un point-virgule ( ; ).

### Appliquer un même style à plusieurs balises

Rien de plus simple :

```
h1, h2  
{  
    color: blue;  
}
```

### Appliquer un style : class

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 un attribut spécial *qui fonctionne sur toutes les balises* :

- l'attribut class;

c'est un attribut que l'on peut mettre sur n'importe quelle balise, aussi bien titre que paragraphe, image, etc.

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

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

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

Oui 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 :

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 :

.introduction

Le fichier HTML	Le fichier CSS
<pre>1 &lt;!DOCTYPE html&gt; 2 &lt;html&gt; 3   &lt;head&gt; 4     &lt;meta charset="utf-8" /&gt; 5     &lt;link rel="stylesheet" href="style2.css" /&gt; 6     &lt;title&gt;Liens&lt;/title&gt; 7   &lt;/head&gt; 8   &lt;body&gt; 9     &lt;p class="introduction"&gt;Bonjour et bienvenue sur mon site !&lt;/p&gt; 10    &lt;p&gt; 11      Mon joli texte...en bleu 12    &lt;/p&gt; 13  &lt;/body&gt; 14 &lt;/html&gt; 15</pre>	<pre>1 p 2 { 3   color: blue; 4 } 5 .introduction 6 { 7   color: red; 8 }</pre>

Testez le résultat

### Les balises universelles

Il arrivera parfois que vous ayez besoin d'appliquer une class 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 :

```
<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.

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 !

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

## Les sélecteurs avancés

**\* : sélecteur universel**

```
*  
  
{  
  
}
```

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

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

`h3 em`

```
{  
  
}
```

Sélectionne toutes les balises `<em>` 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 :

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

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

`h3 + p`

```
{  
  
}
```

Sélectionne la première balise `<p>` située après un titre `<h3>`.

Exemple :

```
<h3>Titre</h3>
```

```
<p>Paragraphe</p>
```

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

`a[title]`

```
{  
}
```

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

Exemple :

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

**A[attribut="Valeur"] : une balise, un attribut et une valeur exacte**  
a[title="Cliquez ici"]

```
{  
}
```

Idem, mais l'attribut doit en plus avoir exactement pour valeur « Cliquez ici ».

Exemple :

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

**A[attribut\*="Valeur"] : une balise, un attribut et une valeur**  
a[title\*="ici"]

```
{  
}
```

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

Exemple :

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

## VIII. Formatage du texte :

### **La taille**

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

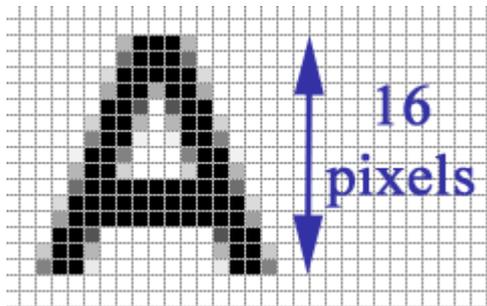
- Indiquer une **taille absolue** : en pixels, en centimètres ou millimètres. Cette méthode est très précise mais il est conseillé de ne l'utiliser que si c'est absolument nécessaire, car on risque d'indiquer une taille trop petite pour certains lecteurs.
- Indiquer une **taille relative** : en pourcentage,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

## Une taille absolue

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

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

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



Une lettre de 16 pixels de hauteur

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

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

## 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 :

### 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 `lapolice1`. S'il ne l'a pas, il essaiera `lapolice2`. S'il ne l'a pas, il passera à `lapolice3`, 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).

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.

### **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**

`<em>`, 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;  
}
```

## 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;
```

}

## 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.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vitae lorem imperdiet lacus molestie molestie. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec eu purus. Phasellus metus lorem, blandit et, posuere quis, tincidunt vitae, ante. Vivamus consequat mauris a diam. Vivamus nibh erat, hendrerit nec, aliquet ut, hendrerit quis, nunc. Vestibulum et turpis et elit tempor euismod.

Une image flottante entourée par du texte

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

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

L'utilisation des flottants est très simple :

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

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

### Faire flotter une image

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

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

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

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

```
.imageflottante
```

```
{
```

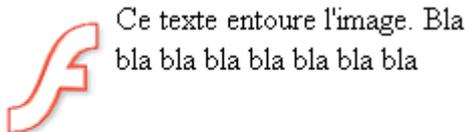
```
float: left;
}
```

Amusez-vous aussi à faire flotter l'image à droite, c'est tout bête : il suffit d'indiquer la valeur right.

### 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.



Texte écrit en-dessous

Le texte sous l'image ignore la propriété float

Il existe en fait une propriété CSS qui permet de dire : « *Stop, ce texte doit être en-dessous du flottant et non plus à côté* ». C'est la propriété `clear`, qui peut prendre ces trois valeurs :

- left: le texte se poursuit en-dessous après un `float: left;`
- right: le texte se poursuit en-dessous après un `float: right;`
- both: le texte se poursuit en-dessous, que ce soit après un `float: left;` ou après un `float: right;`.

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

```
<p></p>
```

```
<p>Ce texte est écrit à côté de l'image flottante.</p>
```

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

Et ce code CSS :

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

## Ajoutez de la couleur et un fond

Continuons notre tour d'horizon des propriétés CSS existantes. Nous allons nous intéresser ici aux propriétés liées de près ou de loin à la couleur. Nous verrons entre autres :

- comment changer la couleur du texte ;
- comment mettre une couleur ou une image d'arrière-plan ;
- comment ajouter des ombres ;
- comment jouer avec les niveaux de transparence.

### Couleur du texte

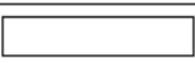
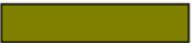
Vous connaissez déjà la propriété qui permet de modifier la couleur du texte : il s'agit de `color`. Nous allons nous intéresser aux différentes façons d'indiquer la couleur, car il y en a plusieurs.

### Indiquer le nom de la couleur

La méthode la plus simple et la plus pratique pour choisir une couleur consiste à taper son nom (*in english, of course*).

Le seul défaut de cette méthode est qu'il n'existe que seize couleurs dites « standard ». D'autres couleurs officieuses existent mais, comme elles ne fonctionneront pas forcément de la même manière sur tous les navigateurs, je vais éviter de vous les montrer.

La figure suivante vous montre les seize couleurs que vous pouvez utiliser en tapant simplement leur nom.

white		red		yellow		fuchsia	
silver		maroon		olive		purple	
gray		lime		blue		aqua	
black		green		navy		teal	

Les seize noms de couleurs utilisables en CSS

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

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

## La notation hexadécimale

Seize couleurs, c'est quand même un peu limite quand on sait que la plupart des écrans peuvent en afficher seize *millions*.

D'un autre côté, remarquez, s'il avait fallu donner un nom à chacune des seize millions de couleurs...

Heureusement, il existe en CSS plusieurs façons de choisir une couleur parmi toutes celles qui existent. La première que je vais vous montrer est la notation hexadécimale. Elle est couramment utilisée sur le Web mais il existe aussi une autre méthode que nous verrons plus loin.

Un nom de couleur en hexadécimal, cela ressemble à : #FF5A28. Pour faire simple, c'est une combinaison de lettres et de chiffres qui indiquent une couleur.

On doit toujours commencer par écrire un dièse (#), suivi de six lettres ou chiffres allant de 0 à 9 et de A à F. Ces lettres ou chiffres fonctionnent deux par deux. Les deux premiers indiquent une quantité de rouge, les deux suivants une quantité de vert et les deux derniers une quantité de bleu. En mélangeant ces quantités (qui sont les composantes Rouge-Vert-Bleu de la couleur) on peut obtenir la couleur qu'on veut.

Ainsi, #000000 correspond à la couleur noire et #FFFFFF à la couleur blanche.

Certains logiciels de dessin, comme Photoshop, [Gimp](#) et [Paint.NET](#), vous indiquent les couleurs en hexadécimal. Il vous est alors facile de copier-coller le code hexadécimal d'une couleur dans votre fichier CSS.

Voici par exemple comment on fait pour appliquer aux paragraphes la couleur blanche en hexadécimal :

```
p
{
    color: #FFFFFF;
}
```

Notez qu'il existe une notation raccourcie : on peut écrire une couleur avec seulement trois caractères. Par exemple : #FA3 équivaut à écrire #FFAA33.

## La méthode RGB

Que signifie RGB ? En anglais, Rouge-Vert-Bleu s'écrit *Red-Green-Blue*, ce qui s'abrège en « RGB ». Comme avec la notation hexadécimale, pour choisir une couleur, on doit définir une quantité de rouge, de vert et de bleu.

```
p
{
    color: rgb(240, 96, 204);
}
```

## 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 ou encore utiliser la méthode RGB.

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 :

```
/* 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

En CSS, si vous appliquez un style à une balise, toutes les balises qui se trouvent à l'intérieur prendront le même style.

C'est en fait simple à comprendre et intuitif. La balise<body>, vous le savez, contient entre autres les balises de paragraphe<p>et de titre<h1>.

Si j'applique une couleur de fond noire et une couleur de texte blanche à la balise<body>, tous mes titres et paragraphes auront eux aussi un arrière-plan de couleur noire et un texte de couleur blanche... 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.

C'est d'ailleurs de là que vient le nom « CSS », qui signifie « *Cascading Style Sheets* », c'est-à-dire « Feuilles de style en cascade ». Les propriétés CSS sont héritées en cascade : si vous donnez un style à un élément, tous les sous-éléments auront le même style.

Cela veut dire que TOUT le texte de ma page web sera forcément écrit en blanc ?

Non, pas obligatoirement. Si vous dites par la suite que vous voulez vos titres en rouge, ce style aura la priorité et vos titres seront donc en rouge. En revanche, si vous n'indiquez rien de particulier (comme on l'a fait tout à l'heure), alors vos titres hériteront de la couleur blanche.

Cela ne fonctionne pas uniquement pour la couleur, entendons-nous bien. Toutes les propriétés CSS seront héritées : vous pouvez par exemple demander une mise en gras dans la balise<body>et tous vos titres et paragraphes seront en gras.

### **Exemple d'héritage avec la balise<mark>**

On a tendance à croire qu'on ne peut modifier que la couleur de fond de la page. C'est faux : vous pouvez changer le fond de n'importe quel élément : vos titres, vos paragraphes, certains mots... Dans ce cas, ils apparaîtront surlignés (comme si on avait mis un coup de marqueur dessus).

Vous vous souvenez par exemple de la balise<mark>qui permet de mettre en valeur certains mots ? Utilisons-la à nouveau ici :

```
<h1>Qui a éteint la lumière ?</h1>
```

```
<p>Brr, il fait tout noir sur ce site, c'est un peu  
<mark>inquiétant</mark> comme ambiance non vous trouvez pas ?</p>
```

Par défaut, le texte s'affiche sur un fond jaune. Vous pouvez changer ce comportement en CSS :

body

```
{  
    background-color: black;  
    color: white;  
}
```

mark

```
{  
    /* La couleur de fond prend le pas sur celle de toute la page */  
    background-color: red;  
    color: black;  
}
```

Sur le texte de la balise<mark>, c'est la couleur de fond rouge qui s'applique. En effet, même si le fond de la page est noir, c'est la propriété CSS de l'élément le plus précis qui a la priorité (figure suivante).

Un texte surligné en rouge sur un fond noir

Le même principe vaut pour toutes les balises HTML et toutes les propriétés CSS ! Si vous dites :

- mes paragraphes ont une taille de 1.2 em ;
- mes textes importants (<strong>) ont une taille de 1.4 em ;

... on pourrait penser qu'il y a un conflit. Le texte important fait partie d'un paragraphe, quelle taille lui donner ? 1.2 em ou 1.4 em ? Le CSS décide que c'est la déclaration la plus précise qui l'emporte : comme<strong>correspond à un élément plus précis que les paragraphes, le texte sera écrit en 1.4 em.

## Images de fond

Dans les exemples qui suivent, je vais modifier l'image de fond de la page. Cependant, tout comme pour la couleur de fond, n'oubliez pas que l'image de fond ne s'applique pas forcément à la page entière. On peut aussi mettre une image de fond derrière les titres, paragraphes, etc.

### Appliquer une image de fond

La propriété permettant d'indiquer une image de fond est `background-image`. Comme valeur, on doit renseigner l'url("nom\_de\_l\_image.png"). Par exemple :

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

Bien entendu, votre fond n'est pas forcément en PNG, il peut aussi être en JPEG ou en GIF.

L'adresse indiquant où se trouve l'image de fond peut être écrite en absolu (`http://...`) ou en relatif (`fond.png`).

Attention lorsque vous écrivez une adresse en relatif dans le fichier CSS ! L'adresse de l'image doit être indiquée *par rapport au fichier.css* et non pas par rapport au fichier.html. Pour simplifier les choses, je vous conseille de placer l'image de fond dans le même dossier que le fichier.css(ou dans un sous-dossier).

### Options disponibles pour l'image de fond

On peut compléter la propriété `background-image` que nous venons de voir par plusieurs autres propriétés qui permettent de changer le comportement de 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).

```
body
{
    background-image: url("neige.png");
    background-attachment: fixed; /* Le fond restera fixe */
}
```

```
}
```

### **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).

Exemple d'utilisation :

```
body
```

```
{
```

```
background-image: url("soleil.png");
```

```
background-repeat: no-repeat;
```

```
}
```

### **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 :

```
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 :

```
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 :

```
body
{
    background-image: url("soleil.png");
    background-attachment: fixed; /* Le fond restera fixe */
    background-repeat: no-repeat; /* Le fond ne sera pas répété */
    background-position: top right; /* Le fond sera placé en haut à droite */
}
```

Combiner les propriétés

Si vous utilisez beaucoup de propriétés en rapport avec le fond (comme c'est le cas sur ce dernier exemple), vous pouvez utiliser une sorte de « super-propriété » appelée background dont la valeur peut combiner plusieurs des propriétés vues précédemment :background-image, background-repeat, background-attachment et background-position.

On peut donc tout simplement écrire :

```
body
{
    background: url("soleil.png") fixed no-repeat top right;
}
```

### 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 :

```
body
{
    background: url("soleil.png") fixed no-repeat top right, url("neige.png") fixed;
}
```

La première image de cette liste sera placée par-dessus les autres (figure suivante). Attention donc, l'ordre de déclaration des images a son importance.

À noter que les images de fond multiples fonctionnent sur tous les navigateurs sauf sur les anciennes versions d'Internet Explorer, qui ne reconnaît cette fonctionnalité qu'à partir de la version 9 (IE9).

Une dernière chose avant d'en terminer avec les images de fond : dans tous ces exemples, j'ai appliqué un fond à la page entière (body). Mais cela ne doit pas vous faire oublier qu'on peut appliquer un fond à n'importe quel élément (un titre, un paragraphe, certains mots d'un paragraphe, etc.).

Je vous conseille donc, pour vous entraîner, d'essayer d'appliquer un fond à vos titres ou paragraphes. Si vous avez un peu de goût (contrairement à moi !) vous arriverez certainement à donner une très belle allure à votre page web.

## La transparence

Le CSS nous permet de jouer très facilement avec les niveaux de transparence des éléments ! Pour cela, nous allons utiliser des fonctionnalités de CSS3 : la propriété `opacity` et la notation `RGBA`.

### 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%... et on verra donc à travers !

Voici comment on peut l'utiliser :

```
p
{
    opacity: 0.6;
}
```

Voici un exemple qui va nous permettre d'apprécier la transparence. Vous en trouverez le rendu à la figure suivante.

```
body
{
    background: url('neige.png');
}
```

```
p
{
```

```
background-color: black;
color: white;
opacity: 0.3;
}
```

Notez que la transparence fonctionne sur tous les navigateurs récents, y compris Internet Explorer à partir de IE9.

Si vous appliquez la propriété `opacity` à un élément de la page, *tout* le contenu de cet élément sera rendu transparent (même les images, les autres blocs à l'intérieur, etc.). Si vous voulez juste rendre la couleur de fond transparente, utilisez plutôt la notation `RGBA` que nous allons découvrir.

### La notation `RGBA`

CSS3 nous propose une autre façon de jouer avec la transparence : la notation `RGBA`. Il s'agit en fait de la notation `RGB` que nous avons vue précédemment, mais avec un quatrième paramètre : le niveau de transparence (appelé « canal alpha »). De la même façon que précédemment, avec une valeur de 1, le fond est complètement opaque. Avec une valeur inférieure à 1, il est transparent.

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

C'est aussi simple que cela. Vous pouvez obtenir exactement le même effet qu'avec `opacity` juste en jouant avec la notation `RGBA`, essayez !

Cette notation est connue de tous les navigateurs récents, y compris Internet Explorer (à partir de IE9). Pour les navigateurs plus anciens, il est recommandé d'indiquer la notation `RGB` classique en plus de `RGBA`. Pour ces navigateurs, le fond ne sera alors pas transparent mais, au moins, il y aura bien une couleur d'arrière-plan.

```
p
{
    background-color: rgb(255,0,0); /* Pour les navigateurs anciens */
    background-color: rgba(255,0,0,0.5); /* Pour les navigateurs plus
récents */
}
```

### Bordures et Ombres

## Bordures standard

Le CSS vous offre un large choix de bordures pour décorer votre page. De nombreuses propriétés CSS vous permettent de modifier l'apparence de vos bordures :border-width,border-color,border-style...

Pour aller à l'essentiel, je vous propose ici d'utiliser directement la super-propriété border qui regroupe l'ensemble de ces propriétés. Vous vous souvenez de la super-propriété background? Cela fonctionne sur le même principe : on va pouvoir combiner plusieurs valeurs.

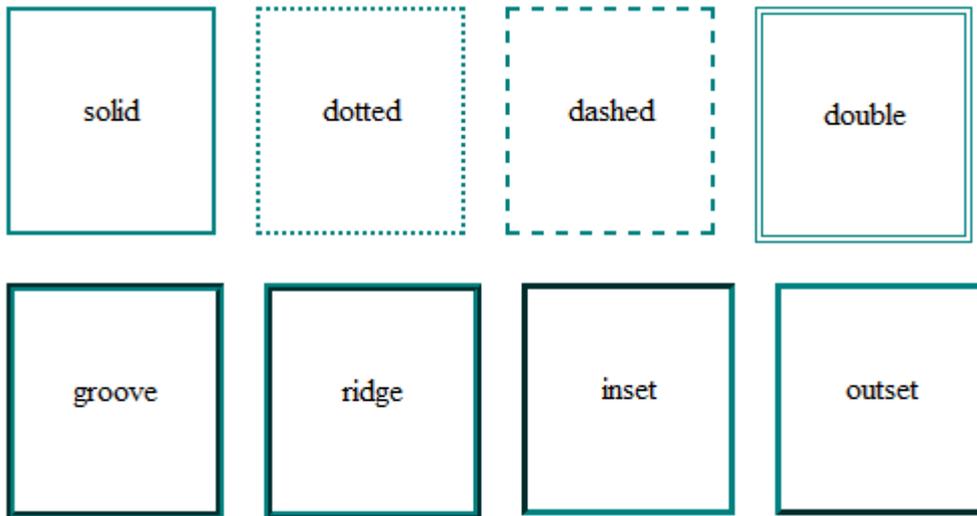
Pour border on peut utiliser jusqu'à trois valeurs pour modifier l'apparence de la bordure :

- **La largeur** : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme2px).
- **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** : 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 :

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

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



### En haut, à droite, à gauche, en bas...

Qui a dit que vous étiez obligés d'appliquer la même bordure aux quatre côtés de votre élément ? Taratata, si vous voulez mettre des bordures différentes en fonction du côté (haut, bas, gauche ou droite), vous pouvez le faire sans problème. Dans ce cas, vous devrez utiliser ces quatre propriétés :

- `border-top`: bordure du haut ;
- `border-bottom`: bordure du bas ;
- `border-left`: bordure de gauche ;
- `border-right`: bordure de droite.

Il existe aussi des équivalents pour paramétrer chaque détail de la bordure si vous le désirez : `border-top-width` pour modifier l'épaisseur de la bordure du haut, `border-top-color` pour la couleur du haut, etc.

Ce sont aussi des super-propriétés, elles fonctionnent comme `border` mais ne s'appliquent donc qu'à un seul côté.

Pour ajouter une bordure uniquement à gauche et à droite des paragraphes, on écrira donc :

```
p
{
  border-left: 2px solid black;
  border-right: 2px solid black;
}
```

On peut modifier les bordures de n'importe quel type d'élément sur la page. Nous l'avons fait ici sur les paragraphes mais on peut aussi modifier la bordure des images, des textes importants comme `<strong>`, etc.

### Bordures arrondies

Les bordures arrondies, c'est un peu le Saint Graal attendu par les webmasters depuis des millénaires (ou presque). Depuis que 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 :

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

**On peut aussi préciser la forme de l'arrondi pour chaque coin. Dans ce cas, indiquez quatre valeurs :**

```
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.

Enfin, il est possible d'affiner l'arrondi de nos angles en créant des courbes elliptiques (figure suivante). Dans ce cas, il faut indiquer deux valeurs séparées par une barre oblique (*slash*, caractère `/`). Le mieux est certainement de tester pour voir l'effet :

```
p
{
    border-radius: 20px / 10px;
}
```

### Les ombres

Les ombres font partie des nouveautés récentes proposées par CSS3. Aujourd'hui, il suffit d'une seule ligne de CSS pour ajouter des ombres dans une page !

Nous allons ici découvrir deux types d'ombres :

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

### **box-shadow: les ombres des boîtes**

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

1. le décalage horizontal de l'ombre ;

2. le décalage vertical de l'ombre ;

3. l'adoucissement du dégradé ;

4. la couleur de l'ombre.

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

```
p
{
    box-shadow: 6px 6px 0px black;
}
```

On peut aussi rajouter une cinquième valeur facultative :inset. Dans ce cas, l'ombre sera placée à l'intérieur du bloc, pour donner un effet enfoncé :

```
p
{
    box-shadow: 6px 6px 6px black inset;
}
```

Avec text-shadow, vous pouvez ajouter une ombre directement sur les lettres de votre texte ! Les valeurs fonctionnent exactement de la même façon quebox-shadow: décalage, adoucissement et couleur.

```
p
{
    text-shadow: 2px 2px 4px black;
}
```

## **Créez des apparences dynamiques**

C'est une de ses forces : le CSS nous permet aussi de modifier 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. Nous allons faire appel à une fonctionnalité puissante du CSS : les pseudo-formats.

Nous verrons dans ce chapitre comment changer l'apparence :

- au survol ;
- lors du clic ;

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

Vous allez voir que le langage CSS n'a pas fini de nous étonner !

### Au survol

Nous allons découvrir dans ce chapitre plusieurs pseudo-formats CSS. Le premier que je vais vous montrer 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, comme ceci :

```
a:hover
```

```
{
```

```
}
```

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

À partir de là, c'est à vous de définir l'apparence que doivent avoir les liens lorsqu'on pointe dessus. Laissez libre cours à votre imagination, il n'y a pas de limite.

Voici un exemple de présentation des liens, mais n'hésitez pas à inventer le vôtre :

```
a /* Liens par défaut (non survolés) */
```

```
{
```

```
text-decoration: none;
```

```
color: red;
```

```
font-style: italic;
```

```
}
```

```
a:hover /* Apparence au survol des liens */
```

```
{
```

```
text-decoration: underline;
```

```
color: green;
}
```

On a défini ici deux versions des styles pour les liens :

- pour les liens par défaut (non survolés) ;
- pour les liens au survol.

Même si on l'utilise souvent sur les liens, vous pouvez modifier l'apparence de n'importe quel élément. Par exemple, vous pouvez modifier l'apparence des paragraphes lorsqu'on pointe dessus :

```
p:hover /* Quand on pointe sur un paragraphe */
{

}
```

### **Au clic et lors de la sélection**

Vous pouvez interagir encore plus finement en CSS. Nous allons voir ici que nous pouvons changer l'apparence des éléments lorsque l'on clique dessus et lorsqu'ils sont sélectionnés !

#### **:active: au moment du 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.

On peut par exemple changer la couleur de fond du lien lorsque l'on clique dessus :

```
a:active /* Quand le visiteur clique sur le lien */
{
    background-color: #FFCC66;
}
```

## Lorsque le lien a déjà été 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 assez laid.

Vous pouvez changer cette apparence avec `:visited`. En pratique, sur les liens consultés, on ne peut pas changer beaucoup de choses à part la couleur (figure suivante).

```
a:visited /* Quand le visiteur a déjà vu la page concernée */
```

```
{
```

```
  color: #AAA; /* Appliquer une couleur grise */
```

```
}
```

***A vous de jouer et de réaliser votre petit site web pour la spécialité SNT !***

***<https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/1604192-decouvrez-le-fonctionnement-des-sites-web>***