

Initiation à HTML/CSS

Une première page

Pour faire une page web, il suffit d'utiliser un éditeur de texte et de créer un fichier HTML. Pour cela, nous allons utiliser Notepad++.

EXERCICE 1 : Ouvrez Notepad++, créer un fichier `premiere_page.html` et mettez `Bonjour !` dedans. Sauvegardez le fichier et ouvrez le avec Firefox, soit en double-cliquant sur le fichier dans l'explorateur de fichiers, soit en faisant CTRL+O dans Firefox.

Bravo, vous avez fait votre première page web.

Cette page est vraiment minimaliste et n'est pas vraiment du HTML (HyperText Markup Language). Ce langage permet de "décrire la page". On ne programme pas en HTML. On ne fait qu'indiquer quelle est la structure du document. Chaque élément est défini à l'aide de **balises**. Les balises sont de deux formes : `<nom_de_balise>` pour ouvrir une balise et `</nom_de_balise>` pour la fermer. Tout ce qui se trouve entre ces deux balises aura les propriétés correspondantes.

Quand on ouvre plusieurs balises à la suite, il faut fermer la dernière ouverte avant de fermer la première, comme pour des parenthèses. En clair :

- Oui : `<balise1><balise2>...</balise2></balise1>`
- Non : `<balise1><balise2>...</balise1></balise2>`

Il est également possible de rajouter des paramètres dans la balise ouvrante, comme nous le verrons plus tard.

Nous allons adopter une structure plus classique. Une page HTML est généralement composée de deux parties : l'entête `<head>...</head>` et le corps `<body>...</body>`. On indique également que le fichier contient du HTML à l'aide de la première ligne `<!DOCTYPE html>` et le reste du document est entre deux balises `<html>...</html>`.

EXERCICE 2 : Enlever le contenu de votre fichier et le remplacer par le suivant :

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    Bonjour !
  </body>
</html>
```

Rechargez la page. Le résultat devrait être le même, mais au moins, la structure de la page va permettre d'aller plus loin.

Une tête bien pleine

L'**entête** de la page permet de définir certaines propriétés de la page, qui ne correspondent pas vraiment au contenu affiché dans la fenêtre :

- le nom de la page (celui qui s'affiche en haut de la fenêtre du navigateur) avec la balise `<title>` ;
- le type d'encodage utilisé avec la balise `<meta>`. C'est une balise spéciale qui est ouvrante et fermante. Elle permet d'éviter que les caractères accentués soient mal affichés ;

- la feuille de style, pour changer l'aspect de la page, mais nous verrons ça plus tard.
- Quasiment toutes nos pages contiendront l'entête suivant :

```
<head>
  <title>Titre de la page</title>
  <meta charset="utf-8">
</head>
```

EXERCICE 3 : Complétez l'entête de votre page pour qu'il affiche le titre "Une page incroyable!".

Sur un corps sain

Le **corps** est la partie la plus importante puisqu'il contient tout ce qui sera affiché par le navigateur. Dans cette partie, vous ne modifierez votre fichier en ne changeant que ce qui se passe entre les deux balises `<body>` et `</body>`.

Une page "basique" s'articule à l'aide de titres et de blocs de texte. Les titres sont définis à l'aide des balises `<h1>` (gros titre), `<h2>` (un peu moins gros titre),... à `<h6>` (tout petit titre).

```
<body>
  <h1>Un gros titre</h1>
  <h2>Un moins gros titre</h2>
  <h3>Un titre moyen</h3>
  <h4>Un petit titre</h4>
  <h5>Un très petit titre</h5>
  <h6>Un très très petit titre</h6>
  Bonjour !
</body>
```

EXERCICE 4 : Remplacez le corps de votre page avec celui ci-dessus et testez-le.

Nous allons maintenant voir comment structurer du texte.

```
<body>
  <h1>Bonjour tout le monde</h1>
  J'adore faire du HTML.

  C'est trop facile.
</body>
```

EXERCICE 5 : Remplacez le corps de votre page avec celui ci-dessus et testez-le.

On peut remarquer que les deux lignes de textes se retrouvent à côté l'une de l'autre. C'est parce que dans le code HTML, les sauts de lignes du fichier ne sont pas répercutés dans la page. Si on veut faire deux paragraphes différents, il faut utiliser la balise `<p>`.

```
<body>
  <h1>Bonjour tout le monde</h1>
  <p>J'adore faire du HTML.</p>
  <p>C'est trop facile.</p>
</body>
```

EXERCICE 6 : Remplacez le corps de votre page avec celui ci-dessus et testez-le.

On peut également rajouter des saut de lignes avec la simple balise `
` qui est **auto-fermante**. Mais cette balise n'est pas conseillée.

```
Une ligne<br>Une autre ligne
```

Il existe une autre balise “générique” qui permet de former un “bloc” autour de plusieurs éléments. C'est la balise `<div>` qui sert à structurer le code et à appliquer un style particulier à une partie de la page, comme nous le verrons après.

```
<div>
  <h1>Titre dans le bloc</h1>
  <p>Texte dans le bloc</p>
</div>
<p>Texte hors du bloc</p>
```

Les blocs se mettent les uns en dessous des autres. Si au contraire, on veut rester sur la même ligne, tout en mettant un style particulier, on peut utiliser la balise ``.

```
<p>Début de la ligne <span>toujours la même ligne</span> fin de la
  ligne</p>
```

Si on veut structurer un peu mieux la page, il existe d'autres balises de type bloc comme `<header>` pour le haut du document, `<section>` pour le contenu et `<footer>` pour le bas de la page. Pour que ces blocs soient pertinents, il faut utiliser du code CSS, comme nous le verrons plus loin.

Des listes de listes

Parfois, il est pratique de faire des listes d'éléments. Pour cela il faut utiliser `` pour commencer la liste et `` pour chaque éléments.

```
<ul>
  <li>Premier élément</li>
  <li>Deuxième élément</li>
  <li>Troisième élément</li>
  <ul>
    <li>Sous-élément</li>
    <li>Encore un</li>
  </ul>
</ul>
```

EXERCICE 7 : Rajoutez le code ci-dessus dans le corps de votre page.

On peut aussi faire des listes numérotées en remplaçant `` par ``.

EXERCICE 8 : Remplacez `` par `` dans votre page pour tester le résultat.

Je ne vois pas le lien...

La base d'HTML, c'est ce qu'on appelle l'**hypertext**. L'idée, c'est que du texte ne soit pas que du texte. Il est possible en cliquant sur une partie de ce texte, appelé **lien**, d'aller sur une autre page. Pour cela on utilise la balise `<a>` dont la syntaxe est la suivante :

```
<a href="adresse de la page cible">Texte qui sera affiché</a>
```

Cela permet d'écrire des choses comme :

```
<ul>
  <li><a href="http://www.wikipedia.fr">www.wikipedia.fr</a></li>
  <li><a href="http://www.google.fr">Une question ?</a></li>
  <li><a href="http://www.qwant.fr">www.google.fr</a></li>
</ul>
```

EXERCICE 9 : Rajoutez dans le corps de votre page les liens ci-dessus et testez-les.

Vous pouvez remarquer qu'il n'y a pas forcément de lien entre le texte affiché et la page cible. En regardant l'adresse qui s'affiche en bas à gauche de votre navigateur lorsque vous passez la souris sur un lien, vous pouvez voir où il va réellement vous envoyer.

Il est également possible de faire des liens vers d'autres pages. Si la page se trouve dans le même dossier, il suffit de mettre le nom de la page comme paramètre de href.

EXERCICE 10 :

1) Créez un nouveau fichier `deuxieme_page.html` et copier le code ci-dessous :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Une deuxième page !</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Et de 2 !</h1>
    <a href="premiere_page.html">
      Retour à la première page</a>
  </body>
</html>
```

Ouvrez la page et vérifiez que le lien envoie bien vers votre première page.

2) Rajoutez un lien dans la première page pointant vers la deuxième.

Il est également possible de faire des liens vers des pages se trouvant dans d'autres dossiers.

EXERCICE 11 :

1) Créez un dossier `secret` dans le dossier où se trouvent les autres fichiers.

2) Créez un nouveau fichier `troisieme_page.html` dans ce dossier et copier le code ci-dessous :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Troisième page</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Rien ne nous arrête</h1>
    <a href="../premiere_page.html">
      Retour à la première page</a>
  </body>
</html>
```

3) Rajoutez le lien suivant dans `premiere_page.html` :

```
<a href="secret/troisieme_page.html">
  En route vers la page 3</a>
```

4) Testez que les liens permettent bien de passer d'une page à l'autre.

Pour les sous-dossiers, on utilise des chemins relatifs à partir de la page courante. C'est-à-dire qu'il faut donner le chemin depuis la page courante. Ainsi pour aller de la première page à la troisième, il faut rentrer dans le dossier `secret`. Au contraire, pour aller de la troisième à la première, il faut sortir du dossier. Pour cela on utilise `..`.

Il est tout à fait possible de rentrer dans plusieurs dossiers ou d'en quitter plusieurs. À chaque fois, on utilise `/` pour séparer les noms de dossiers ou de fichiers.

EXERCICE 12 :

1) Rajoutez le lien suivant dans `troisieme_page.html` :

```
<a href=" ../mystere/mysterieux/quatrieme_page.html ">
  Est-ce que ça marche ?</a>
```

2) Créez les dossiers nécessaires et le fichier `quatrieme_page.html` au bon endroit. Vous pouvez mettre le code que vous voulez dedans.

3) Rajoutez les liens nécessaires dans les autres pages pour qu'on puisse passer de n'importe laquelle des 4 pages à n'importe quelle autre.

Le poids des mots, le choc des photos

Le texte c'est bien, mais les images, c'est mieux. Pour insérer une image, il faut utiliser la balise auto-fermante `` qui nécessite un certain nombre d'attributs. La syntaxe est :

```

```

L'adresse de l'image suit les mêmes règles que pour les liens. L'image peut se trouver sur un autre site, et dans ce cas on met l'adresse complète, ou sur le même ordinateur que la page, et dans ce cas, on donne le chemin pour y accéder.

EXERCICE 13 :

1) Rajoutez le code suivant dans votre page :

```

```

Rechargez la page et vérifiez que le texte alternatif s'affiche.

2) Allez chercher l'image `image_chien.jpg` qui se trouve dans `Echange` et copiez-la dans le même dossier que vos fichiers HTML.

3) Rechargez la page qui doit maintenant afficher l'image.

Par défaut, l'image s'affiche en ligne, c'est-à-dire qu'elle peut se mettre au milieu du texte. Si on veut la mettre à part, il faut rajouter une balise `<div>` autour.

Un peu de style, ça ne fait pas de mal

Tout ce qu'on vient de faire manque de fantaisie. Il existe diverses balises permettant de changer l'aspect du texte. Voici un exemple de ce qu'il est possible de faire :

```

<p>
  On peut avoir du texte <i>en italique</i>, <b>en gras</b>.<br>
  On peut le <u>souligné</u>.<br>
  On peut faire du <small>petit</small>
  ou du <big>grand</big>.<br>
  On peut faire des indices : 100101<sub>2</sub>
  ou des exposants 2<sup>3</sup>.<br>
  On peut même <b><i><u>cumuler</u></i></b> !
</p>

```

EXERCICE 14 : Testez le code ci-dessus.

Il y a d'autres balises génériques qui font sensiblement la même chose, mais le comportement peut changer d'un navigateur à l'autre et il est possible de les redéfinir pour changer le style du document.

```

<p>
  <em>Mettre en avant</em> quelque chose.<br>
  <strong>Renforcer</strong> l'effet.<br>
  La fonction <code>print("mon message").</code><br>
  La variable <var>x</var> en math.
</p>

```

EXERCICE 15 : Testez le code ci-dessus.

CSS : Décorateur d'intérieur

Si on veut vraiment changer le style d'une page, il faut utiliser le CSS : *Cascading Style Sheet*. Il permet de modifier totalement l'aspect d'une page : couleur du fond, ajout de cadres, changement de la police, alignement du texte...

On peut rajouter du code CSS dans le code HTML, mais il est préférable de le mettre à part dans un autre fichier.

EXERCICE 16 :

1) Créez un fichier `une_belle_page.html` et mettez-y le code suivant :

```

<!DOCTYPE html>
<html>
  <head>
    <title>Une bien belle page</title>
    <link rel="stylesheet" href="feuille_de_style.css">
    <meta charset="utf-8">
  </head>
  <body>
    <header>
      <h1>Un titre qui claque</h1>
    </header>
    <section>
      <h1>Un titre qui rend bien</h1>
      <p>Des trucs à dire très intéressants</p>
      
    </section>
  </body>
</html>

```

```

        </section>
        <section>
            <h1>Et un autre pour la route</h1>
            <p>Un premier paragraphe</p>
            <p>Un deuxième paragraphe</p>
            <p>Un paragraphe <strong>costaud</strong></p>
        </section>
    </body>
</html>

```

- 2) Chargez la page et vérifiez que le résultat n'est pas si terrible que ça.
- 3) Créez un fichier `feuille_de_style.css` et mettez-y le code suivant :

```

/* pour définir des propriétés globales dans tous le document */
body {
    background-color: lavender;
    margin: 0;
}

/* on crée une balise juste pour le titre */
header {
    background-color: red;
    padding: 20px;
    margin: 20px;
    border: 5px solid black;
}

/* pour le titre dans header
 * les autres h1 ne sont pas affectés */
header h1 {
    text-align: center;
    background-color: greenyellow;
    color: blue;
    margin: 10px;
    padding: 30px;
    border: 5px dotted black;
}

section {
    padding: 20px;
    margin: 20px;
    background-color: lightsteelblue;
}

```

- 4) Rechargez la page HTML et admirez le résultat.

Comme vous pouvez le voir, on définit des attributs qui sont affecté à chaque balise. La syntaxe générale est :

```

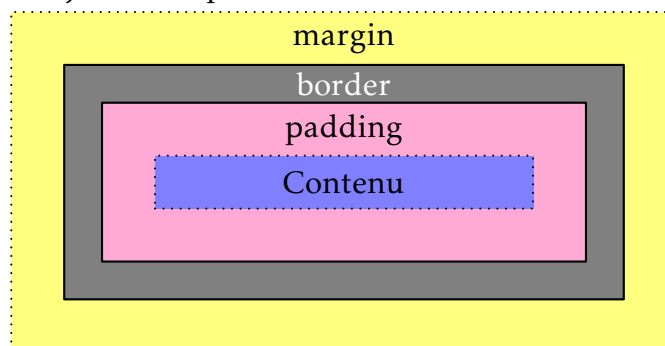
balise {
    attribut1: valeur1;
    attribut2: valeur2;
}

```

Il existe de nombreux attributs, qui peuvent s'appliquer à toutes les balises ou juste à certaines.

- **text-align** permet de déterminer la justification du texte. Les valeurs possibles sont **left**, **right**, **center** et **justify**.
- **color** permet de choisir la couleur du texte.
- **background-color** permet de choisir la couleur du fond.
- **border** permet de faire un encadrement autour du bloc, avec les attributs **taille motif couleur**. Il existe de nombreux motifs possibles. On peut aussi mettre **none** pour ne pas afficher le cadre.
- **padding** permet de modifier l'espace entre l'objet et l'encadrement (qui peut être de taille nulle). On peut donner de une à 4 longueurs selon si on veut mettre la même pour toutes directions ou utiliser des valeurs différentes.
- **margin** fait la même chose, mais pour créer de l'espace entre le cadre et les autres éléments.

Les espaces autour de l'objet sont représentés ainsi :



Les longueurs en HTML/CSS peuvent s'exprimer en pixels (20px) ou en pourcentage de la place disponible (30%). On peut aussi utiliser **em** qui correspond à la taille de la police.

Pour les couleurs, on peut utiliser :

- le nom : **red**, **blue**, **white**, **black**, **darkorchid**, **moccasin**. . . ;
- les composantes RGB : **rgb(r, v, b)**, avec chacune des valeurs entre 0 (pas de cette couleur) et 255 (couleur au max) ;
- les composantes rouge, vert, bleu et opacité : **rgba(r, v, b, alpha)**, avec **alpha** entre 0 (transparent) et 1 (opaque) ;
- l'hexadécimal : **#4F210B**, chaque paire de symboles représentant une couleur.

Il existe de nombreux sites permettant de trouver la bonne couleur et indiquant comment l'écrire en HTML. En voici deux :

- <https://www.rapidtables.com/web/color/html-color-codes.html>
- https://www.w3schools.com/colors/colors_picker.asp

Pour les textes, les principaux attributs sont :

- **font-family** pour changer de police. On peut en mettre plusieurs au cas où les premières ne soient pas trouvées par le navigateur.
- **font-weight** pour changer l'épaisseur du texte : **normal**, **light**, **bold** et **bolder**
- **font-size** pour changer la taille du texte. Par exemple **2em** double la taille. On peut aussi mettre **xx-small**, **x-small**, **small**, **normal**, **large**, **x-large** et **xx-large**.
- **text-decoration** pour mettre une ligne sous, sur ou au dessus du texte : **none**, **underline** ou **line-through**.

Pour revenir à l'exemple, on peut remarquer que la couleur de fond est définie dans le corps du document mais aussi dans les sections. Lorsqu'un même attribut est défini plusieurs

fois, c'est celui de la balise le plus à l'intérieur qui est utilisé. Dans ce cas, les sections passent avant la définition du fond du corps.

On peut voir aussi qu'on peut changer les attributs `<h1>` qui se trouve uniquement dans `<header>`.

```
b1 b2 b3 {  
    attribut1: valeur1;  
    attribut2: valeur2;  
}
```

Avec la syntaxe ci-dessus, on peut modifier uniquement les attributs de la balise `<b3>` si elle est dans `<b2>`, elle même dans `<b1>`. Dans tous les autres cas, la balise `<b3>` gardera ses attributs de base, ou définis ailleurs.

EXERCICE 17 : Si on veut changer les attributs d'une balise seulement à certains endroits, on peut définir une classe.

1) Rajoutez la partie suivante à votre fichier CSS :

```
.interessant {  
    font-style: italic;  
    color: red;  
}
```

2) Modifiez le premier paragraphe de la page HTML avec :

```
<p class="interessant">Des trucs à dire très intéressants</p>
```

3) Rechargez la page et remarquez que ce paragraphe a changé.

Il est possible de donner plusieurs classes à une même balise ou d'utiliser cette classe dans d'autres balise. Si on veut faire un changement uniquement sur une seule balise dans le document, il faut lui donner une id unique :

```
<balise id="mon_id">...</balise>
```

Et ensuite, on peut définir :

```
#mon_id {  
    attribut1: valeur1;  
    attribut2: valeur2;  
}
```

EXERCICE 18 :

1) Rajoutez la partie suivante à votre fichier CSS :

```
#bob {  
    font-weight: bolder;  
}
```

2) Modifiez la ligne correspondante de la page HTML avec :

```
<p id="bob">Un deuxième paragraphe</p>
```

3) Rechargez la page et remarquez que ce paragraphe a changé.

Il est également possible de faire des modifications très ciblées dans le CSS :

```
b1#mon_id b2 .special{
    attribut1: valeur1;
    attribut2: valeur2;
}
```

Cela ne modifiera que les propriétés d'une balise de classe `special` dans une balise `<b2>` dans une balise `b1` avec l'id `mon_id`.

Enfin, on peut mettre l'image sur le côté du texte et pas en dessous.

EXERCICE 19 : Rajoutez le code suivant dans votre CSS et rafraîchissez la page :

```
img {
    float: right;
    margin: 20px;
}
```

Si on a plusieurs images, il peut être intéressant de passer par des classes plutôt que de modifier les attributs de la balise image.

Pour aller plus loin

EXERCICE 20 : Récupérez les fichiers `page_a_ne_pas_toucher.html` et `css_a_modifier.css` et en modifiant que le fichier CSS, donnez du style à cette page.

Si vous voulez en apprendre plus sur les possibilités de HTML et CSS, vous pouvez aller voir les pages suivantes :

- <https://developer.mozilla.org/fr> qui est en français.
- <https://www.w3schools.com/> en anglais. La partie HTML Tag Reference est très complète.