

# HTML/CSS

## *Objectifs*

Je serai capable de :

- Editer une page HTML à l'aide d'un éditeur de texte et d'un navigateur
- Comprendre la syntaxe et les bases du langage HTML (notions : balise, attribut, valeur d'attribut et contenu)
- Comprendre la structure générale d'un document HTML (html, head et body)
- Composer une page HTML en utilisant à bon escient les balises :
  - de titres (title, h1 à h6)
  - de paragraphe (p, br, blockquote)
  - de mise en évidence de texte (em, span)
  - d'images (img)
  - de tableaux (table, tr, td)
  - de listes numérotées ou à puces (ol, ul, li)
  - de liens (a)
  - de sections (div)
- Formater une page HTML en créant et en appliquant des styles séparément du contenu
- Ecrire les styles en respectant la syntaxe (notions de sélecteur, propriété et valeur associée à une propriété)

## ***Le langage HTML***

HTML est l'abréviation de « **HyperText Mark-up Language** » ; ce terme décrit donc un langage (**Language**) formé de balises (<h1>, <h2>, <p>, <a>, <ol> et bien d'autres) qui permettent de marquer (**Mark-up**) les éléments d'un texte brut (**Text**) pour mettre en forme le texte et lui donner une cohérence au niveau de la présentation. Ces balises seront interprétées par un navigateur web. Le terme **Hyper** apporte une dimension supplémentaire : le document ainsi marqué peut-être lié à d'autres documents répartis sur les différents serveurs du monde à travers le réseau Internet. La personne qui consulte une information peut ainsi « sauter » (ou passer) d'une page à une autre en cliquant sur les liens disponibles sur chaque document.

Tim Berners-Lee est à l'origine de ce langage basé sur le SGML. Dès 1989, il travaille sur trois technologies qui sont à l'origine du web d'aujourd'hui : les adresses web, le http (protocole de communication web) et, bien sûr, le langage HTML. Il fonde en 1994 le W3C (world wide web consortium) qui permettra de définir les standards actuels, ainsi que de faire évoluer et d'améliorer le web.

## Syntaxe et bases du langage HTML<sup>1</sup>

### Les balises

HTML est un langage de balisage ; cela signifie que, pour être interprété par le navigateur qui affichera le document, chaque élément d'un document (texte, images, vidéo, etc.) doit être compris entre une balise d'ouverture et une balise de fermeture (à part quelques exceptions, de type `<br>`, `<img>`, etc., qui sont des balises « solitaires »).

Une balise HTML a la syntaxe suivante :



La balise d'ouverture est placée entre un chevron ouvrant « < » et un chevron fermant « > ».

Pour la balise de fermeture, un « / » est placé entre les chevrons, avant le nom de la balise.

Il existe une liste prédéfinie de balises HTML, dont voici quelques exemples simples :

Titre 1	<pre>&lt;h1&gt;</pre> <p>Ceci est un titre 1</p> <pre>&lt;/h1&gt;</pre>	<p><b>Ceci est un titre 1</b></p> <p>Ceci est un paragraphe</p>
Paragraphe	<pre>&lt;p&gt;</pre> <p>Ceci est un paragraphe</p> <pre>&lt;/p&gt;</pre>	

<sup>1</sup> Depuis sa première version, le HTML a été étendu et amélioré. Aujourd'hui, le langage en est à sa 5<sup>ème</sup> version, nommée « html5 ».

Des attributs peuvent être ajoutés à la plupart des balises d'ouverture. Ils ont la même fonction, indépendamment de la balise sur laquelle ils sont appliqués :

**id** : cet attribut permet d'identifier un élément d'une page de façon unique pour permettre à un script (javascript) d'y accéder (ce qui permettra par exemple de modifier cet élément). Cet identificateur peut aussi être utilisé pour appliquer un formatage particulier à l'élément en question grâce à une feuille de style CSS (voir plus loin).

**title** : cet attribut permet d'afficher une bulle quand l'utilisateur positionne le curseur de sa souris sur l'élément.

**class** : cet attribut permet de faire référence à une classe de style pour *la mise en page / le formatage* de l'élément.

**style** : cet attribut permet de définir « inline » un *style / format* pour l'élément donné.

Reprenons l'exemple ci-dessus en introduisant des attributs dans la balise <p> :

Paragraphe	<pre>&lt;h1&gt;     Ceci est un titre 1 &lt;/h1&gt; &lt;p id="p1" title="PARAGRAPHE 1"     style="font-family:Verdana;color:blue"&gt;     Ceci est un paragraphe &lt;/p&gt;</pre>	
------------	---	--

Vous découvrirez dans la suite du cours d'autres attributs – des attributs de gestion d'événements – qui sont utilisés notamment pour exécuter du code javascript, par exemple :

**onclick** : si l'utilisateur clique sur l'élément, alors un programme est exécuté.

**onmouseover** : si l'utilisateur passe le curseur de sa souris sur l'élément, alors un programme est exécuté.

## Structure de base d'un document HTML/page web



Une page HTML commence par le **<!DOCTYPE>** ; cette déclaration n'est pas une balise HTML, mais une instruction destinée aux navigateurs web qui indique la version du langage HTML dans laquelle la page est écrite. Par défaut, nous utiliserons la déclaration HTML5 : **<!DOCTYPE html>**.

Après cette déclaration, le code contient la balise d'ouverture **<html>** et se termine par la balise de fermeture **</html>** ; à l'intérieur de cette balise, deux autres balises principales composent le document :

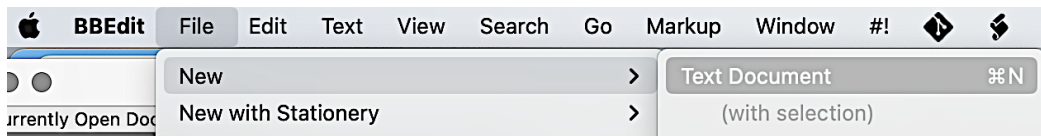
1. **<head></head>** : elle contient notamment le titre de la page qui apparaîtra sur la barre du navigateur, les chemins menant aux feuilles de styles, les scripts éventuels...
2. **<body></body>** : elle contient le corps de la page, ce qui s'affichera dans la fenêtre du navigateur.

Remarquez enfin que, pour faciliter la lecture du code HTML, on introduit un retour à la ligne après les balises principales du document et on « indente » (c'est-à-dire que l'on décale vers la droite) les balises imbriquées. Les retours à la ligne, l'indentation et les espaces – à l'exception de ceux situés entre les mots - ne sont pas pris en compte lors de l'affichage des textes sur la page web.

Il est possible d'ajouter des éléments d'information dans une page web sans que ceux-ci ne soient affichés à l'écran. Ceci est effectué grâce à un jeu de balises spécifiques, les balises de commentaires, de type `<!-- Voici un commentaire -->`.

Créez dans l'éditeur BBEdit un nouveau fichier de type html que vous nommerez 'essai.html'. Pour cela, suivez les étapes suivantes :

1. Ouvrez l'éditeur BBEdit.
2. Créez un nouveau fichier de type Text Document.



3. Copiez le code ci-dessous dans ce fichier.
4. Sauvez le fichier dans vos documents sous le nom 'essai.html'.

```
<!DOCTYPE html>
```

```
<html lang="fr">
```

```
  <head>
```

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

```
    <title>Titre de la page</title>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```

Cette structure est la structure de base d'un code html.

Pour la suite du cours, nous vous conseillons de créer plusieurs fichiers « modèles » que vous pourrez dupliquer et utiliser dès que vous souhaiterez vous lancer dans la rédaction d'un nouveau code. Les différents modèles utiles sont les suivants :

- un fichier *html\_modele.html* contenant le code de la page 6.
- un fichier *html\_css\_modele.html* comportant de plus une balise <link> permettant de lier le fichier html à une feuille de styles css (*cf pages suivantes*).
- un fichier *html\_js\_modele.html* comportant de plus une balise <script> permettant d'introduire du code javascript (*cf module 8*).
- un fichier *html\_canvas\_modele.html* comportant de plus une balise <canvas> et sa fonction d'initialisation afin de pouvoir introduire une représentation graphique dans la page web (*cf module 9*).

### Balises utiles pour la suite (non exhaustif)

Double-cliquez sur le fichier 'essai.html'. Il s'ouvrira dans le navigateur défini par défaut sur votre ordinateur (Safari, Firefox, Google Chrome, ...). Pour l'instant, la page web est « blanche » puisque qu'il n'y a pas de code à l'intérieur de la balise <body>.

Le fichier 'essai.html' est donc ouvert dans deux applications sur votre ordinateur :

- dans l'application BBEdit, grâce à laquelle vous pourrez **modifier** et **sauver** votre fichier.
- dans l'application Safari (ou Firefox, Google Chrome, etc.), grâce à laquelle vous pourrez **visualiser** le rendu de votre code sur une page web (ce qui apparaît lorsqu'on charge le fichier html dans un navigateur).

Pour passer d'une application à l'autre sur un Mac, il est conseillé d'utiliser le raccourci clavier **Command + Tab**.

Chaque fois que vous introduisez une modification dans l'application BBEdit et que vous souhaitez observer l'effet de cette modification, vous devez :

1. Sauver (raccourci clavier : **Command + S**) le fichier 'essai.html' dans BBEdit.
2. Recharger la page web dans le navigateur.

Les pages suivantes présentent la liste des balises principales en HTML.

Testez les différentes balises en introduisant le code en gris dans la partie `<body>` de votre fichier *essai.html*, puis en observant le rendu du code ainsi modifié dans le navigateur.

Par exemple pour tester la balise `<a>`, vous devez suivre les étapes suivantes :

1. Dans BBEdit, introduisez l'extrait de code à tester dans la balise `<body>` de la manière suivante :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Titre de la page</title>
  </head>
  <body>
    <a href=" https://www.ocinfo.ch/OCinfo/" target="_blank">OCinfo.ch</a>
  </body>
</html>
```

2. Sauvez (raccourci clavier : **Command + S**) le fichier 'essai.html' dans BBEdit.
3. Allez dans le navigateur (raccourci clavier : **Command + Tab**).
4. Rechargez la page web dans le navigateur et observez l'effet de l'introduction de la balise.

Balise	Syntaxe
<b>a</b>	<pre>&lt;a href="url" target="_blank"&gt;Contenu&lt;/a&gt;</pre> <pre>&lt;a href=" https://www.ocinfo.ch/OCinfo/" target="_blank"&gt;OCinfo.ch&lt;/a&gt;</pre>
<b>blockquote</b>	<pre>&lt;blockquote&gt;Contenu&lt;/blockquote&gt;</pre> <pre>&lt;blockquote&gt;Ce texte est maintenant une citation.&lt;/blockquote&gt;</pre>
<b>br</b>	<pre>&lt;br&gt;</pre> <pre>&lt;p&gt;</pre> <p>La balise br permet de faire apparaître un retour à la ligne  <pre>&lt;br&gt;</pre> à la position souhaitée sans l'introduction d'un nouveau paragraphe.</p>



	<p>&lt;/p&gt; &lt;p&gt;</p> <p>La balise br est une balise solitaire.</p> <p>&lt;/p&gt; &lt;p&gt;</p> <p>Cette balise doit être utilisée avec parcimonie. En particulier, la balise p doit toujours être privilégiée pour faire apparaître des paragraphes.</p> <p>&lt;/p&gt;</p>
<b>div</b>	<p><b>&lt;div class="nom du style" style="format"&gt;Contenu&lt;/div&gt;</b></p> <p>&lt;div style="background-color:black;color:white;"&gt;</p> <p>    &lt;h1&gt;la balise div&lt;/h1&gt;</p> <p>    &lt;p&gt;</p> <p>        Une div est une section dans le document qui peut contenir différents éléments (paragraphes, titres, tableaux, etc...)</p> <p>    &lt;/p&gt;</p> <p>    &lt;p&gt;</p> <p>        Une div permet de grouper des éléments au même endroit dans une page web et de leur attribuer un même format.</p> <p>    &lt;/p&gt;</p> <p>    &lt;p&gt;Une div occupe toujours une zone rectangulaire&lt;/p&gt;</p> <p>&lt;/div&gt;</p>
<b>em</b>	<p><b>&lt;em&gt;Contenu&lt;/em&gt;</b></p> <p>&lt;p&gt;</p> <p>    Ceci est un paragraphe dans lequel se trouve un &lt;em&gt;élément&lt;/em&gt; que l'on souhaite mettre en évidence et qui apparaîtra par défaut en italique.</p> <p>&lt;/p&gt;</p>
<b>h1, h2, ..., h6</b> <b>Les titres</b>	<p><b>&lt;h1&gt;Contenu&lt;/h1&gt;</b></p> <p>    &lt;h1&gt;Chapitre 1&lt;/h1&gt;</p> <p>    &lt;p&gt;bla bla&lt;/p&gt;</p> <p>    &lt;h2&gt;Partie 1&lt;/h2&gt;</p> <p>    &lt;p&gt;bla bla&lt;/p&gt;</p> <p>    &lt;h3&gt;Partie 1.1&lt;/h3&gt;</p> <p>    &lt;p&gt;bla bla&lt;/p&gt;</p>

<b>hr</b>	<p><b>&lt;hr&gt;</b></p> <p>&lt;p&gt;La balise hr permet de générer une ligne horizontale.&lt;/p&gt;</p> <p>&lt;hr&gt;</p>
<b>img</b>  <b>Les images</b>	<p><b>&lt;img src="url"width="pixel   pourcent" height="pixel   pourcent"&gt;</b></p> <p>&lt;img src="http://ocinfo.ch/ocinfo.gif" width="400" height="200" &gt;</p> <p>Modifiez la largeur et la hauteur de l'image en px ou en utilisant les pourcentages (valeur de 50%, par exemple) et observez l'effet obtenu...</p>
<b>ol, ul, li</b>  <b>Les listes</b>	<p><b><u>Liste ordonnée (1., 2., 3., etc.) :</u></b></p> <p><b>&lt;ol&gt;</b></p> <p><b>&lt;li&gt;</b> Contenu 1 <b>&lt;/li&gt;</b></p> <p><b>&lt;li&gt;</b> Contenu 2 <b>&lt;/li&gt;</b></p> <p><b>&lt;li&gt;</b> Contenu 3 <b>&lt;/li&gt;</b></p> <p><b>&lt;/ol&gt;</b></p> <p>&lt;ol&gt;</p> <p>&lt;li&gt;Mon premier se dirige quelque part.&lt;/li&gt;</p> <p>&lt;li&gt;Mon deuxième est la moitié d'un cheveu.&lt;/li&gt;</p> <p>&lt;li&gt;Mon tout vit à la ferme.&lt;/li&gt;</p> <p>&lt;/ol&gt;</p> <p><b><u>Liste non ordonnée (puces) :</u></b></p> <p><b>&lt;ul&gt;</b></p> <p><b>&lt;li&gt;</b> Contenu 1 <b>&lt;/li&gt;</b></p> <p><b>&lt;li&gt;</b> Contenu 2 <b>&lt;/li&gt;</b></p> <p><b>&lt;li&gt;</b> Contenu 3 <b>&lt;/li&gt;</b></p> <p><b>&lt;/ul&gt;</b></p> <p>&lt;ul&gt;</p> <p>&lt;li&gt;Mon premier se dirige quelque part.&lt;/li&gt;</p> <p>&lt;li&gt;Mon deuxième est la moitié d'un cheveu.&lt;/li&gt;</p> <p>&lt;li&gt;Mon tout vit à la ferme.&lt;/li&gt;</p> <p>&lt;/ul&gt;</p>

<b>p</b>	<p><b>&lt;p&gt;</b>Contenu<b>&lt;/p&gt;</b></p> <p>&lt;p&gt;Ceci est un paragraphe.&lt;/p&gt;</p>
<b>span</b>	<p><b>&lt;span class="nom du style" style="format"&gt;</b>Contenu<b>&lt;/span&gt;</b></p> <p>&lt;p&gt;</p> <p>Ceci est un paragraphe dans lequel se trouve un        &lt;span style="font-family:verdana;font-size:30px;"&gt;élément&lt;/span&gt;        que l'on souhaite mettre en évidence en lui attribuant un style particulier.</p> <p>&lt;/p&gt;</p>
<b>table, tr, td</b>  <b>Les</b>  <b>tableaux</b>	<p><b>&lt;table border="1"&gt;</b> //début du tableau</p> <p><b>&lt;tr&gt;</b> //début de la première ligne</p> <p><b>&lt;td&gt;</b>Contenu_cellule ligne 1 colonne 1<b>&lt;/td&gt;</b></p> <p><b>&lt;td&gt;</b>Contenu_cellule ligne 1 colonne 2<b>&lt;/td&gt;</b></p> <p><b>&lt;/tr&gt;</b> //fin de la première ligne</p> <p><b>&lt;tr&gt;</b> //début de la deuxième ligne</p> <p><b>&lt;td&gt;</b>Contenu_cellule ligne 2 colonne 1<b>&lt;/td&gt;</b></p> <p><b>&lt;td&gt;</b>Contenu_cellule ligne 2 colonne 2<b>&lt;/td&gt;</b></p> <p><b>&lt;/tr&gt;</b> //fin de la deuxième ligne</p> <p><b>&lt;/table&gt;</b> //fin du tableau</p> <pre> &lt;table border="1" width="600"&gt;   &lt;tr&gt;     &lt;td colspan="3" align="center"&gt;       Valeurs des actions le 22 juillet 2010 à 10 h 54     &lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;UBS&lt;/td&gt;     &lt;td align="center"&gt;Crédit Suisse&lt;/td&gt;     &lt;td align="right"&gt;Nestlé&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;Frs. 15.31&lt;/td&gt;     &lt;td align="center"&gt;Frs. 43.08&lt;/td&gt;     &lt;td align="right"&gt;Frs. 53.20&lt;/td&gt;   &lt;/tr&gt; &lt;/table&gt; </pre>

## Quelques combinaisons, trucs et astuces

### Liens absolus vs liens relatifs

Les attributs href de la balise <a> et src de la balise <img> pointent sur un fichier qui peut se trouver soit sur un autre serveur, soit localement sur la même machine. La valeur de ces attributs est donc l'adresse qui donne accès à la page ou à l'image voulue. Il existe plusieurs possibilités en fonction de la situation :

Types	Exemples
absolu	<pre>&lt;a href="https://www.ocinfo.ch/elevés/index.html"&gt;</pre> <p>liens sur les pages d'accueil des élèves</p> <pre>&lt;/a&gt;</pre> <pre>&lt;img src="https://www.ocinfo.ch/image_pingouin.png"&gt;</pre> <p>Le lien absolu est une adresse <u>complète</u> !</p>
relatif	<pre>&lt;a href="ex_01.html"&gt; exercice 1 &lt;/a&gt;</pre> <pre>&lt;img src="image_pingouin.png"&gt;</pre> <p>Dans ces exemples, le lien relatif pointe sur un document/une image qui se trouve dans le même dossier que le document HTML contenant le lien.</p>
	<pre>&lt;a href="m01/ex_01.html"&gt; exercice 1 &lt;/a&gt;</pre> <pre>&lt;img src="images/image_pingouin.png"&gt;</pre> <p>Dans ces exemples, le lien relatif pointe sur un document/une image qui se situe dans un sous-dossier du dossier dans lequel se trouve le document HTML contenant le lien.</p>
	<pre>&lt;a href=" ../ ex_01.html "&gt; exercice 1 &lt;/a&gt;</pre> <pre>&lt;img src=" ../ image_pingouin.png"&gt;</pre> <p>Dans ces exemples, le lien relatif pointe sur un document/une image qui se situe dans le dossier parent du dossier dans lequel se trouve le document HTML contenant le lien.</p>

## Combinaison de balises HTML

```
<a href="index.html">
```

```
  
```

```
</a>
```

Dans cet exemple, c'est une image qui fait office de lien : si l'on clique sur l'image du pingouin, la page index.html s'ouvre.



## Principes à respecter lors de l'écriture des balises HTML

**Principe 1** : toute balise ouvrante doit obligatoirement être fermée.

```
<p>Ceci est un paragraphe <!-- FAUX -->  
<p>Ceci est un paragraphe</p> <!-- JUSTE -->
```

**Principe 2** : les balises et noms de leurs attributs s'écrivent en minuscules

```
<IMG SRC='images/1.png'> <!-- FAUX -->  
<img src='images/1.png'> <!-- JUSTE -->
```

**Principe 3** : les valeurs des attributs sont toujours encadrées par des guillemets ou des apostrophes

```
<div id=resultat class=divResultat> <!-- FAUX -->  
  <p>Ceci est un paragraphe dans une div</p>  
</div>  
  
<div id="resultat" class='divResultat'> <!-- JUSTE -->  
  <p>Ceci est un paragraphe dans une div</p>  
</div>
```

**Principe 4** : les balises solitaires s'écrivent <nomBalise>

```
<img src = "images/pingouin.png"> <!-- JUSTE -->  
</img src = "images/pingouin.png"> <!-- FAUX -->
```

*Remarque* : En xhtml (version précédente de html), les balises solitaires étaient considérées comme « ouvrantes-fermantes » et la syntaxe était la suivante : <br/>, <img/>, etc. Le « slash » a été abandonné lors du passage à html.

## Principe 5 : chaque élément doit être imbriqué correctement

```
<div id="resultat" class='divResultat'>
<p>Ceci est un paragraphe dans une div</div></p> <!-- FAUX -->

<div id="resultat" class='divResultat'> <!-- JUSTE -->
  <p>Ceci est un paragraphe dans une div</p>
</div>
```

## Principe 6 : le code doit être indenté

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8">
5     <title>liens sur les pages d'accueil</title>
6     <link rel="stylesheet" href="styles.css">
7   </head>
8   <body>
9     <table>
10      <tr>
11        <td colspan="4">
12          <h1 class="titrePrincipal">
13            liens sur les pages d'accueil personnelles
14            <br>
15            des élèves de l'oc informatique
16          </h1>
17        </td>
18      </tr>
19      <tr class='index'>
20        <td>
21          <h2>
22            <a href="e01/index.html">nom de l'élève n°01</a>
23          </h2>
24        </td>
25        <td>
26          <h2>
27            <a href="e11/index.html">nom de l'élève n°11</a>
28          </h2>
29        </td>
30        <td>
31          <h2>
32            <a href="e21/index.html">nom de l'élève n°21</a>
33          </h2>
34        </td>
35        <td>
36          <h2>
37            <a href="e31/index.html">nom de l'élève n°31</a>
38          </h2>
39        </td>
40      </tr>
41      <tr class='index'>
42        <td>
43          <h2>
44            <a href="e02/index.html">nom de l'élève n°02</a>
45          </h2>
46        </td>
47        <td>
48          <h2>
49            <a href="e12/index.html">nom de l'élève n°12</a>
50          </h2>
51        </td>
```

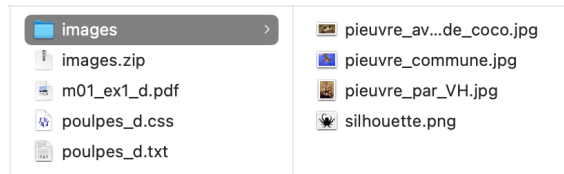
## Exercice 1, étape 1

Après avoir découvert et testé par vous-même un certain nombre de balises HTML et intégré les principes de base d'écriture du langage HTML, il est temps d'appliquer ces connaissances en créant une page web complète. Ceci sera fait en deux étapes :

- **Etape 1** : marquage d'un texte avec les balises HTML
- **Etape 2** : définition des styles à appliquer aux balises. Cette étape sera effectuée après la lecture de la théorie sur les feuilles de styles CSS (pages 18 à 27).

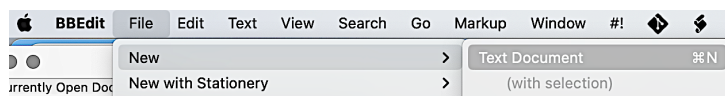
Pour effectuer cet exercice, vous devez télécharger depuis le moodle du cours les fichiers suivants, que vous trouverez sous la rubrique 'exercice 1 : donnée' :

- un fichier nommé 'images.zip', qui contient des fichiers compressés. En double-cliquant sur ce fichier, vous obtiendrez quatre fichiers d'images (trois fichiers .jpg et un fichier .png), groupés dans un dossier nommé 'images'.
- le fichier de texte « pur » nommé 'poulpes\_d.txt'
- le fichier css nommé 'poulpes\_d.css', que vous devez renommer 'poulpes\_c.css'<sup>2</sup>. Ce fichier sera complété dans la deuxième partie de l'exercice
- la donnée de l'exercice (fichier pdf nommé 'm01\_ex1\_d.pdf').



De plus, vous devez créer dans BBEdit un nouveau fichier de type html que vous nommerez 'poulpes\_c.html'. Pour cela, suivez les étapes suivantes :

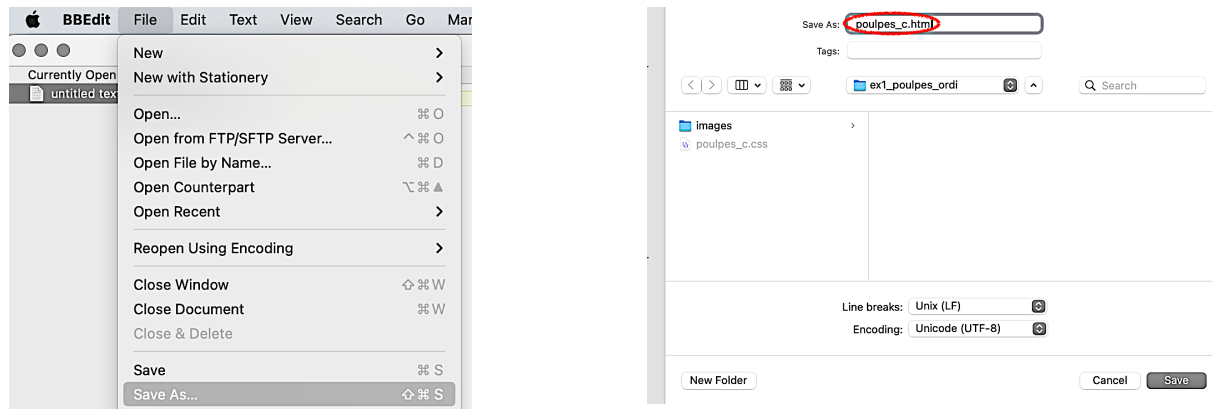
- Ouvrez l'éditeur BBEdit.
- Créez un nouveau fichier de type Text Document.



<sup>2</sup> Dans ce cours, les fichiers dont le nom se termine par '\_d' sont les fichiers contenant les données initiale et les fichiers dont le nom se termine par '\_c' sont les fichiers contenant le corrigé.



- Sauvez ce fichier sous le nom 'poulpes\_c.html' dans le même dossier que celui dans lequel se trouvent le dossier 'images' et le fichier 'poulpes\_c.css'.



#### Remarque :

Contrôlez bien que les trois éléments (dossier 'images', fichier 'poulpes.css' et fichier 'poulpes.html') se trouvent bien dans le même dossier dans vos documents. Si ce n'est pas le cas, les liens entre eux ne fonctionneront pas : les images n'apparaîtront pas et/ou la mise en forme du texte ne pourra pas être lue... L'emplacement des deux autres fichiers ('poulpes\_d.txt' et 'm01\_ex1\_d.pdf') n'a pas d'incidence sur le rendu de l'exercice, mais il est fortement conseillé de placer les fichiers de manière logique dans vos documents afin d'être en mesure de les retrouver facilement lors des révisions.



**Suivez attentivement les explications et consignes contenues dans l'étape 1 de la donnée de l'exercice 1 (pages 1 à 4 du fichier 'm01\_ex1\_d.pdf') en respectant scrupuleusement les principes de base énoncés précédemment et en testant régulièrement le résultat dans votre navigateur.**

Cette première étape réalisée, il vous reste encore à définir pour chaque élément la manière dont il doit s'afficher, c'est-à-dire à indiquer son format. Ceci sera réalisé dans l'étape 2 de l'exercice 1, après lecture des pages 18 à 27.

## ***Les feuilles de styles CSS (Cascading Style Sheets)***

CSS permet de définir l'apparence des textes (police, couleur, taille, bordure, ...) ainsi que l'agencement de la page (marges, espace avant et après un paragraphe, arrière-plan, ...).

### **Syntaxe CSS de base : général et exemple**

```
sélecteur {  
    propriété1 : valeur 1 ;  
    propriété2 : valeur 2 ;  
    propriété3 : valeur 3 ;  
    propriété4 : valeur 4 ;  
}  
  
p {  
    font-family : verdana ;  
    font-size : 2em ;  
    color : white ;  
    background-color : black ;  
}
```

Le code écrit en css ci-dessus à droite pour le sélecteur p permettra de faire apparaître tous les textes contenus à l'intérieur des balises <p> avec le style suivant :

**Ceci est un paragraphe mis en page avec un style**

Un sélecteur est composé d'une ou plusieurs *propriétés* auxquelles on attribue une/des *valeur(s)* (un couple propriété-valeur s'appelle une « *déclaration* »).

## Où se trouve le code CSS ?

Il existe trois possibilités pour lier du code css à un fichier html.

Le code css peut être placé :

*1) Dans un fichier externe, afin de séparer les données de leur mise en page.*

C'est de loin la meilleure solution ; elle présente en particulier les avantages suivants :

- la possibilité d'appliquer les mêmes styles à plusieurs documents : il suffit pour cela d'introduire un lien sur la même feuille de styles dans chacun des documents.
- une gestion facilitée de la gestion et de l'évolution du site, grâce à une séparation très claire des parties HTML et CSS.

*2) A l'intérieur du fichier HTML, entre les balises <head>.*

Cette solution permet de conserver la séparation entre les données et le format mais n'est intéressante que pour formater un fichier "solitaire" ; à partir du moment où plusieurs pages web doivent être formatées de la même manière, la solution n°1 s'impose !

*3) Inline, directement à l'intérieur d'une balise.*

Cette solution, dans laquelle il n'existe plus de séparation entre les données et le format, est à éviter de manière générale, sauf dans des cas très spécifiques, lorsque les fichiers sont très courts (par exemple, lors des premiers essais d'interactions html-javascript dans le module 8...).

Ainsi, les trois possibilités présentées ci-dessous donneront le même résultat sur la page web associée. Ce résultat est le suivant :

**Ce paragraphe est mis en page avec un style.**

### 1. Dans un fichier externe

fichier index.html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>exemple</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>

    <p>
      Ce paragraphe est mis en page avec un style.
    </p>

  </body>
</html>
```

fichier styles.css

```
p{
  font-family : verdana;
  font-size : 2em;
  font-weight : bold;
  color : red;
  letter-spacing : 6px;
}
```

### 2. Entre les balises head

fichier index.html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>exemple</title>
    <style type="text/css">

      p{
        font-family : verdana;
        font-size : 2em;
        font-weight : bold;
        color : red;
        letter-spacing : 6px;
      }

    </style>
  </head>
  <body>

    <p>
      Ce paragraphe est mis en page avec un style.
    </p>

  </body>
</html>
```

### 3. Inline

fichier index.html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>exemple</title>
  </head>
  <body>

    <p style = "font-family:verdana;font-size:2em;
      font-weight:bold;color:red;letter-spacing:6px;">
      Ce paragraphe est mis en page avec un style.
    </p>

  </body>
</html>
```

## Les sélecteurs (non-exhaustif)

« Une des grandes richesses de CSS est la multiplicité des possibilités de sélection des éléments auxquels on veut attribuer un style donné. Cette très grande diversité permet en effet d'appliquer un style aussi facilement à tous les éléments, en une seule ligne de code, qu'à un unique élément isolé dans la page web, sans avoir à écrire la définition localement. De plus, la combinaison de plusieurs sélecteurs dans la même déclaration ouvre la voie à une quasi-infinité de combinaisons, propres à répondre à tous les besoins, même les plus complexes. »<sup>3</sup>

### Les sélecteurs correspondant aux balises HTML

```
p {font-family:verdana; font-size:2em; color:red;}
```

En utilisant ce sélecteur, nous définissons pour chaque paragraphe, dans l'exemple de la page précédente, la même police (verdana), la même taille (2 em, qui correspond à deux fois la taille de la police par défaut) et la même couleur de la police (rouge).

### Plusieurs sélecteurs pour un même format

Nous désirons appliquer la même police de caractère (verdana) aux titres 1, 2 et 3 :

```
h1 {font-family:verdana;}
```

```
h2 {font-family:verdana;}
```

```
h3 {font-family:verdana;}
```

Plutôt que d'écrire ces trois lignes, nous pouvons nous contenter de n'en écrire qu'une :

```
h1,h2,h3 {font-family:verdana;}
```

Rien n'empêche d'ajouter ensuite d'autres propriétés à l'un ou l'autre des sélecteurs :

```
h1,h2,h3 {font-family:verdana;}
```

```
h1 {font-size:35px;}
```

```
h2 {font-size:28px;}
```

La taille des titres 1 et 2 ne sera pas la même, mais la police sera identique !

---

<sup>3</sup> Citation tirée du livre XHTML et CSS de Jean Engels

## Les sélecteurs 'html' et 'body'

On emploie très souvent des sélecteurs pour attribuer des styles aux balises `<html>` et `<body>`. La distinction entre les sélecteurs 'html' et 'body' n'est cependant pas évidente. On peut dire que le sélecteur 'html' est associé à toute la fenêtre du navigateur (hormis, bien sûr le bandeau supérieur), alors que le sélecteur 'body' n'agit que dans la zone rectangulaire contenant tous les éléments (texte et autre) représentés dans la page.

En particulier, cela signifie que l'on ne peut pas attribuer de largeur à la balise `<html>` (puisque l'utilisateur a la possibilité de modifier la taille de la fenêtre), alors que cette valeur peut être imposée à la balise `<body>`. Cette dernière ne sera néanmoins entièrement visible que si la largeur choisie est inférieure à celle de la fenêtre du navigateur.

En pratique et dans le cadre de ce cours, on se basera sur les constatations suivantes :

- Le rôle principal du sélecteur 'html' est de fixer la couleur ou l'image de fond de la fenêtre.
- Le sélecteur 'body' permet de dimensionner la région « utile » de la fenêtre, d'y appliquer des marges interne et/ou externes (pour la centrer, par exemple) et d'imposer certaines valeurs par défaut à toutes les balises situées à l'intérieur de la balise `<body>`, par exemple le nom de la police de caractères ou la taille des caractères.

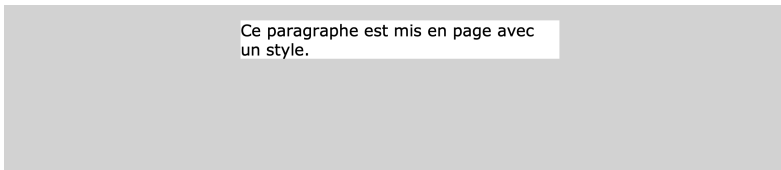
On écrira donc par exemple :

```
html {  
    background-color:lightGrey;  
}  
  
body {  
    font-size:25px;  
    font-family:verdana;  
    background-color:white;  
    width:500px;  
    margin-top:20px;  
    margin-right:auto;  
    margin-left:auto;  
}
```

L'exemple ci-contre montre un usage typique de ces sélecteurs. Il permet de produire une page d'une largeur de 500 pixels, centrée dans la fenêtre du navigateur (grâce aux déclarations 'margin-left:auto' et 'margin-right:auto'), avec une marge supérieure de 20 pixels. Le fond de la fenêtre est gris clair mais le fond de la page elle-même est blanc. Par défaut, le texte est en police 'verdana' avec une taille de 25 pixels.

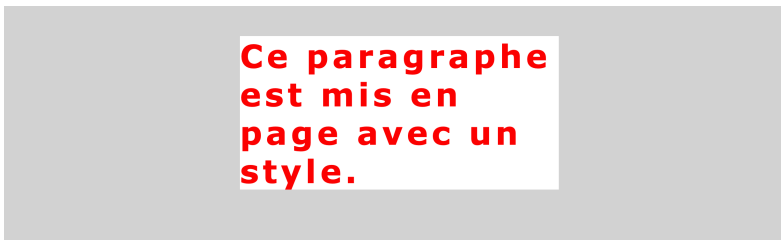
En reprenant le fichier index.html de l'exemple de la page 20 et en ajoutant (ou pas) le sélecteur 'p' dans le fichier de styles, on obtient les pages web suivantes :

page web



fichier styles.css

```
html {  
    background-color:lightGrey;  
}  
  
body {  
    font-size:25px;  
    font-family:verdana;  
    background-color:white;  
    width:500px;  
    margin-top:20px;  
    margin-right:auto;  
    margin-left:auto;  
}
```



```
html {  
    background-color:lightGrey;  
}  
  
body {  
    font-size:25px;  
    font-family:verdana;  
    background-color:white;  
    width:500px;  
    margin-top:20px;  
    margin-right:auto;  
    margin-left:auto;  
}  
  
p{  
    font-size:2em;  
    font-weight:bold;  
    color:red;  
    letter-spacing:6px;  
}
```

Dans le deuxième exemple, on constate que la taille par défaut pour les textes (25px) a été remplacée par la taille associée au sélecteur 'p' (2em). On dit que cette valeur a été « écrasée » (c'est-à-dire « perdue au profit d'une autre »).

## Les sélecteurs « classe »

Il existe deux manières de sélectionner une classe :

	<i>Globalement</i>	<i>Pour une balise particulière</i>	
CSS	<pre>.message{     background-color:red;     color:white; }</pre>	<pre>p.message {     background-color:red;     color:white; }</pre>	<pre>h1.message {     background-color:green;     color:yellow; } p.message {     background-color:red;     color:white; }</pre>
HTML	<pre>&lt;h1 class="message"&gt;     Ceci est un titre. &lt;/h1&gt; &lt;p class="message"&gt;     Ceci est un paragraphe. &lt;/p&gt;</pre>	<pre>&lt;h1 class="message"&gt;     Ceci est un titre. &lt;/h1&gt; &lt;p class="message"&gt;     Ceci est un paragraphe. &lt;/p&gt;</pre>	
Résultat	<div>Ceci est un titre.</div> <div>Ceci est un paragraphe.</div>	<div>Ceci est un titre.</div> <div>Ceci est un paragraphe.</div>	<div>Ceci est un titre.</div> <div>Ceci est un paragraphe.</div>

La première possibilité permet à toute balise HTML d'utiliser la classe « **message** » en utilisant son attribut **class**.

La deuxième écriture restreint l'utilisation de la classe **message** à un seul type de balise.



Remarque : pour appliquer un format sur une partie d'un élément (par exemple, sur un mot), il faut utiliser la balise `<span>` (ou `<em>`, voir remarque en bas de page). En utilisant différentes classes, cela permet d'obtenir divers styles de mise en évidence :

CSS	<pre>.substantif,.adjectif {     font-style : italic; }  .substantif {     color : red;     text-decoration : overline; }  .adjectif {     color : blue; }</pre>
HTML	<pre>&lt;p&gt;   Ici, nous mettons en &lt;span class="substantif"&gt;évidence&lt;/span&gt;   des mots &lt;span class="adjectif"&gt;particuliers&lt;/span&gt; du texte   et ce, de &lt;span class="adjectif"&gt;différentes&lt;/span&gt;   &lt;span class="substantif"&gt;manières&lt;/span&gt;. &lt;/p&gt;</pre>
Résultat	<p>Ici, nous mettons en <u>évidence</u> des mots <i>particuliers</i> du texte et ce, de <i>différentes</i> <i>manières</i>.</p>

Cette manière de faire est de loin préférable à l'usage de balises « toutes faites » comme `<b>` (gras), `<i>` (italique) ou `<u>` (souligné), qui sont des balises obsolètes dont on proscrira l'usage. En effet, il est important de distinguer clairement la partie html, dans laquelle on se concentre sur la signification et la structure de ce que l'on écrit, et la partie css, dans laquelle on gère la mise en forme.

Il existe aussi une balise `<em>` (pour « emphasize ») qui s'utilise de la même manière qu'une balise `<span>` ; l'unique différence entre ces deux balises est que, par défaut, le texte contenu à l'intérieur de la balise `<em>` apparaît en italique sur la page web.

Considérons par exemple, le morceau de code html suivant,

<p>

Ici, nous mettons en <em>évidence</em> des mots <em>particuliers</em> du texte.

</p>

et observons de quelle manière la mise en évidence peut-être réalisée à l'aide de code CSS.

Code CSS	Résultat dans le navigateur
<pre>em {   font-family : arial, sans;   text-decoration : underline;   color : red; }</pre>	Ici, nous mettons en <u>évidence</u> des mots <u>particuliers</u> du texte.
<pre>em {   font-style : normal; /* pas en italique */   font-weight : bold;   color : orange;   background-color : green; }</pre>	Ici, nous mettons en <b>évidence</b> des mots <b>particuliers</b> du texte.

## Les sélecteurs « identificateur »

Si on souhaite appliquer un style particulier à un élément unique (une seule balise), il est préférable de lui attribuer un identificateur (attribut 'id') et d'utiliser un sélecteur d'id (commençant par '#'), comme dans l'exemple qui suit, plutôt que d'utiliser une classe (attribut 'class' de la balise :

Code HTML	<pre>&lt;p id="special"&gt;   Voici un paragraphe unique en son genre. &lt;/p&gt; &lt;p&gt;   Celui-ci est des plus communs.. &lt;/p&gt;</pre>
Code CSS	<pre>#special {   text-align : center;   font-weight : bold;   border : 5px solid red; }</pre>
Résultat dans le navigateur	<div><b>Voici un paragraphe unique en son genre.</b></div> <p>Celui-ci est des plus communs..</p>

## Les propriétés et valeurs possibles

Pour découvrir les propriétés et valeurs possibles pour un style, il est maintenant temps de passer à l'étape 2 de l'exercice 1.

### **Exercice 1, étape 2**

Suivez les instructions du fichier *m01\_ex1\_d.pdf* à partir de la page 5.

### **Exercices à faire ensuite**

exercices 2 (sur papier), 3 (sur ordinateur), 4 (sur ordinateur) et 5 (sur papier) à télécharger sur le moodle de [www.ocinfo.ch](http://www.ocinfo.ch).

## Pour aller plus loin...

### *Quelques liens sur le web*

HTML wikipedia : [http://fr.wikipedia.org/wiki/Hypertext\\_Markup\\_Language](http://fr.wikipedia.org/wiki/Hypertext_Markup_Language)

Référence HTML : <http://www.w3schools.com/tags/default.asp>

Référence CSS : <https://www.w3schools.com/cssref/>

Markup Validation Service : <http://validator.w3.org/>

CSS Validation Service : <http://jigsaw.w3.org/css-validator/>

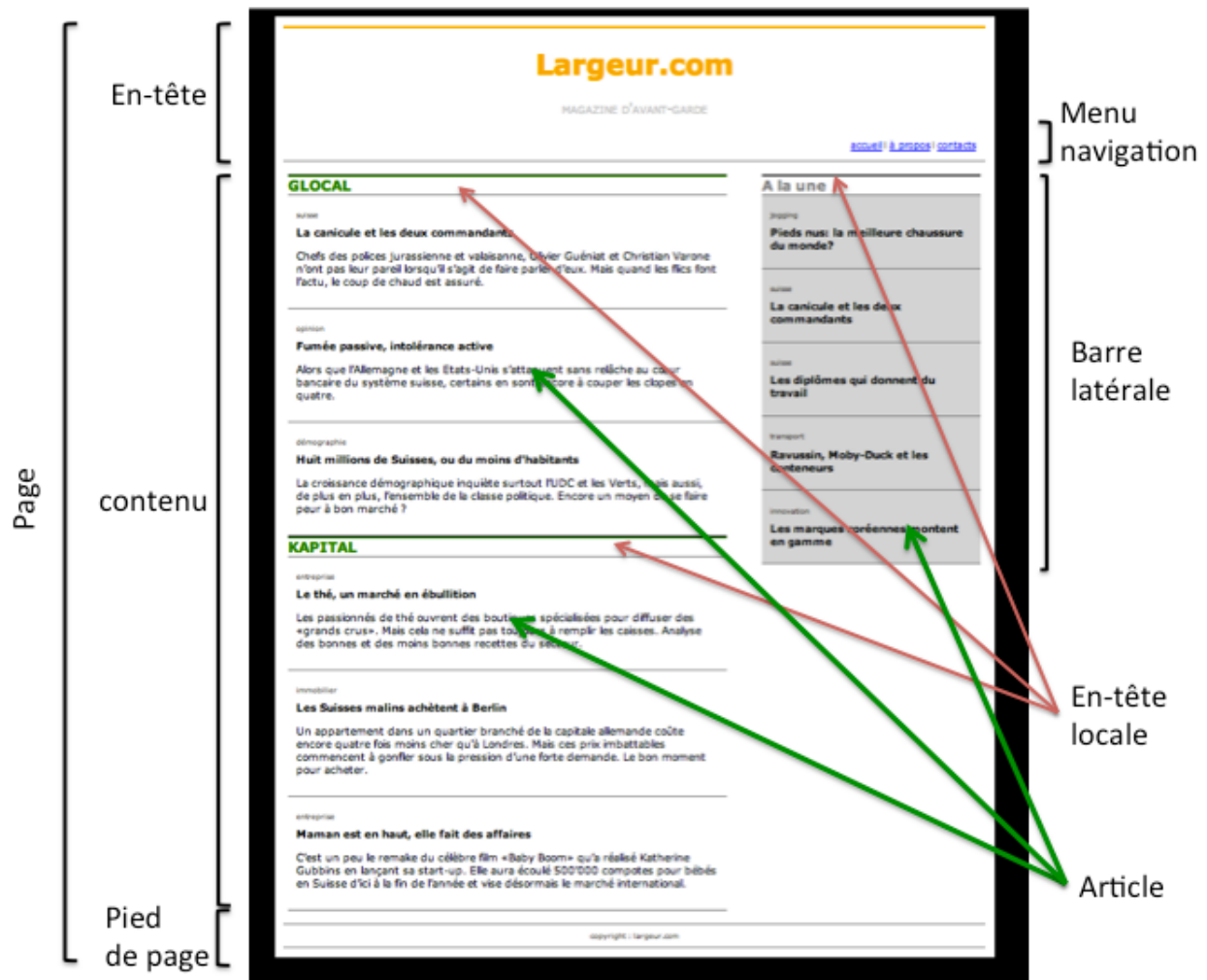
Valdateur HTML expérimental : <http://html5.validator.nu/>

### *Des balises plus explicites pour remplacer les <div> (facultatif)*

Cette section facultative présente des balises dont l'usage n'est pas indispensable mais permet de clarifier le code html pour un lecteur humain.

Nous avons vu jusqu'ici qu'une balise **div** est une section dans un document qui peut contenir différents éléments (paragraphe, titres, tableaux, etc...). Une div permet en fait de grouper des éléments au même endroit dans une page web et leur attribuer un format prédéfini.

Téléchargez les fichiers de donnée de l'exercice 6 : fichier *index\_d.html*, associé au fichier de styles *styles\_d.css*. En ouvrant *index\_d.html*, vous obtenez la page web visible sur la page suivante.



Cette page est entièrement construite avec des div, qui ont été utilisées pour définir les positions de la page, de l'en-tête, du pied de page, du menu de navigation, de la barre latérale, des en-têtes locales, ainsi que des articles.

Voici un extrait du code html de cette page :

```
<!DOCTYPE html>
<html lang="fr">

  <head>
    <meta charset="utf-8">
    <title>Largeur.com</title>
    <link rel="stylesheet" href="styles_d.css">
  </head>

  <body>

    <!-- entête de la page -->
    <div id="entete">
      <h1 class="titrePrincipal">Largeur.com</h1>
      <p class="titreSecondaire">magazine d'a
      <div id="navigation">
        <a href="accueil.html">accueil</a>
        <a href="apropos.html">à propos</a>
        <a href="contacts.html">contacts</a>
      </div>
    </div>

    <!-- barre latérale -->
    <div id="barreLaterale">
      <div id="aLaUne">A la une</div>

      <a href="article101.html" class="lienAr
      <div class="articleBarre">
        <p class="categorie2">jogging</p>
    </div>
  </body>
</html>
```

Le corps de la page est composé de **div** qui permettent d'en séparer le contenu :

1. une div avec l'id « entete » qui permet de lui attribuer un style particulier, dans laquelle se trouve une div avec l'id « navigation »
2. une div avec l'id « barreLaterale »
3. une div avec l'id « contenu »
4. une div avec l'id « pied »

Certaines de ces div de « premier niveau » contiennent elles-mêmes des div pour les structurer.

Le code CSS associé à ces div :

```
#entete {
  border-top: 3px solid orange;
  border-bottom: 1px solid grey;
  margin-bottom: 10px;
}

#barreLaterale{
  width:250px;
  float:right;
  padding:5px;
}
```

```
#contenu{
  width:500px;
  padding:5px;
}

#pied{
  margin-top: 10px;
  border-top: 1px solid grey;
  border-bottom: 1px solid grey;
  text-align:center;
  font-size: 8px;
}
```

Bien que le fichier *index\_d.html* soit magnifiquement indenté, vous constatez qu'il est difficile d'y repérer les différentes parties de la page html. Tout ce qu'on voit, ce sont les nombreuses div auxquelles on a associé des id ou des classes.

La version 5 d'HTML permet de clarifier un peu le contenu d'un fichier html en définissant toute une série de nouvelles balises qui peuvent remplacer les <div> dans des rôles spécialisés. Notons que cela ne simplifie pas vraiment *l'écriture* du fichier et ne change (presque) pas le *rendu* de la page html car, pour le navigateur, chacune de ces nouvelles balises est essentiellement équivalente à une balise <div><sup>4</sup>. Leur utilisation est donc laissée à votre libre appréciation mais il est important que vous soyez conscient de leur existence. C'est le rôle de l'exercice 6.

Modifions pas à pas les fichiers *index\_d.html* et *styles\_d.css* de cet exercice pour introduire les nouvelles balises lorsque cela est pertinent.

## Balise pour générer des en-têtes : <header> </header>

### Code avec des div

```
<!-- entete de la page -->  
<div id="entete"> ☒ </div>
```

### Code avec header

```
<!-- entete de la page -->  
<header> ☒ </header>
```

### CSS avec des div

```
#entete {  
  border-top: 3px solid orange;  
  border-bottom: 1px solid grey;  
  margin-bottom: 10px;  
}
```

### CSS avec header

```
header {  
  border-top: 3px solid orange;  
  border-bottom: 1px solid grey;  
  margin-bottom: 10px;  
}
```

Attention ! Si vous transformez une div avec un id (par exemple "entete") en une balise html 5 (par exemple "header"), vous devez adapter le code CSS en conséquence: dans l'exemple proposé, le sélecteur **#entete** doit être renommé **header**.

---

<sup>4</sup> Plus précisément, il existe de petites différences entre les balises <div> et ces nouvelles balises, mais elles sont minimes. Par exemple, les valeurs des marges par défaut peuvent légèrement différer d'un type de balise à l'autre.

## Balise pour générer une barre latérale : <aside> </aside>

### Code avec des div

```
<!-- barre latérale -->  
<div id="barreLaterale"> ☒ </div>
```

### Code avec aside

```
<!-- barre latérale -->  
<aside> ☒ </aside>
```

### CSS avec des div

```
#barreLaterale{  
  width:250px;  
  float:right;  
  padding:5px;  
}
```

### CSS avec aside

```
aside{  
  width:250px;  
  float:right;  
  padding:5px;  
}
```

## Balise pour générer une section : <section> </section>

### Code avec des div

```
<!-- contenu de la page -->  
<div id="contenu"> ☒ </div>
```

### Code avec section

```
<!-- contenu de la page -->  
<section> ☒ </section>
```

### CSS avec des div

```
#contenu{  
  width:500px;  
  padding:5px;  
}
```

### CSS avec section

```
section{  
  width:500px;  
  padding:5px;  
}
```



## Balise pour générer un pied de page : <footer> </footer>

### Code avec des div

```
<!-- pied de la page -->
<div id="pied">
  <p>copyright : largeur.com</p>
</div>
```

### CSS avec des div

```
#pied{
  margin-top: 10px;
  border-top: 1px solid grey;
  border-bottom: 1px solid grey;
  text-align:center;
  font-size: 8px;
}
```

### Code avec footer

```
<!-- pied de la page -->
<footer>
  <p>copyright : largeur.com</p>
</footer>
```

### CSS avec footer

```
footer{
  margin-top: 10px;
  border-top: 1px solid grey;
  border-bottom: 1px solid grey;
  text-align:center;
  font-size: 8px;
}
```

## Balise pour générer une zone menu/navigation : <nav> </nav>

### Code avec des div

```
<div id="navigation">
  <a href="accueil.html">accueil</a> | <a href="aprc.html">aprc</a>
</div>
```

### CSS avec des div

```
#navigation {
  padding:10px;
  font-family: courier;
  font-size: 10px;
  text-align:right;
}
```

### Code avec nav

```
<nav>
  <a href="accueil.html">accueil</a> | <a href="aprc.html">aprc</a>
</nav>
```

### CSS avec nav

```
nav {
  padding:10px;
  font-family: courier;
  font-size: 10px;
  text-align:right;
}
```

Intéressons-nous à présent au contenu principal de la page :

```
<!-- contenu de la page -->
<div id="contenu">
  <!-- catégorie 1 -->
  <div class="categorie1">GLOCAL</div>
  <!-- article 1 -->
  <a href="article11.html" class="lienArticle">
    <div class="article"> </div>
  </a>
  <!-- article 2 -->
  <a href="article12.html" class="lienArticle">
    <div class="article"> </div>
  </a>
  <!-- article 3 -->
  <a href="article13.html" class="lienArticle">
    <div class="article"> </div>
  </a>
</div>
```

Le contenu de la page comporte des div de classe « categorie1 » et des div de classe « article »...

Nous utiliserons pour la classe « **categorie1** » la balise **header** (déjà vue) et pour la classe « **article** » la balise **article** (également introduite dans HTML5).

## Balise pour générer une zone header et une zone article :

### <header> </header> et <article> </article>

On utilise ici la balise header pour montrer qu'il existe un en-tête qui coiffe les articles rattachés à un domaine. Cependant, comme on a déjà utilisé la balise header pour l'en-tête de la page avec un style précis, il faut conserver la classe « categorie1 », qui devient une classe de la balise header.

La lisibilité du code s'améliore grandement en utilisant ces nouvelles balises...

#### Code avec des div

```
<!-- contenu de la page -->
<div id="contenu">
  <!-- catégorie 1 -->
  <div class="categorie1">GLOCAL</div>
  <!-- article 1 -->
  <a href="article11.html" class="lienArticle">
    <div class="article">
  </a>
  <!-- article 2 -->
  <a href="article12.html" class="lienArticle">
    <div class="article">
  </a>
  <!-- article 3 -->
  <a href="article13.html" class="lienArticle">
    <div class="article">
  </a>
```

#### Code avec les balises de html 5

```
<!-- contenu de la page -->
<section>
  <!-- catégorie 1 -->
  <header class="categorie1">GLOCAL</header>
  <!-- article 1 -->
  <a href="article11.html" class="lienArticle">
    <article>
  </a>
  <!-- article 2 -->
  <a href="article12.html" class="lienArticle">
    <article>
  </a>
  <!-- article 3 -->
  <a href="article13.html" class="lienArticle">
    <article>
  </a>
```

#### CSS avec des div

```
div.categorie1 {
  border-top-color:green;
  border-top-style:groove;
  border-top-width:3px;
  border-bottom: 1px grey solid;
  color:green;
  font-weight: bold;
}

a.lienArticle{
  text-decoration: none;
}

div.article{
  padding:10px;
  border-bottom: 1px grey solid;
  color:black;
  text-decoration: none;
}
```

#### CSS avec les balises de html 5

```
header.categorie1 {
  border-top-color:green;
  border-top-style:groove;
  border-top-width:3px;
  border-bottom: 1px grey solid;
  color:green;
  font-weight: bold;
}

a.lienArticle{
  text-decoration: none;
}

article{
  padding:10px;
  border-bottom: 1px grey solid;
  color:black;
  text-decoration: none;
}
```

Procédez de même avec les articles rattachés au domaine « Kapital ».

Finalement, modifiez les div de la barre latérale (correction ci-dessous).

## Correction pour la barre latérale...

```
<!-- barre latérale -->
<aside>
  <header class="aLaUne">A la une</header>
  <a href="article101.html" class="lienArticle">
    <article class="articleBarre">
      <p class="categorie2">jogging</p>
      <h1 class="titreArticle">Pieds nus: la meill
    </article>
  </a>
  <a href="article102.html" class="lienArticle">
    <article class="articleBarre">
      <p class="categorie2">suisse</p>
      <h1 class="titreArticle">La canicule et les
    </article>
  </a>
  <a href="article103.html" class="lienArticle">
    <article class="articleBarre">
      <p class="categorie2">suisse</p>
      <h1 class="titreArticle">Les diplômes qui dc
    </article>
  </a>
  <a href="article104.html" class="lienArticle">
    <article class="articleBarre">
      <p class="categorie2">transport</p>
      <h1 class="titreArticle">Ravussin, Moby-Duck
    </article>
  </a>
  <a href="article105.html" class="lienArticle">
```

```
header.aLaUne{
  border-top-color:grey;
  border-top-style:groove;
  border-top-width:3px;
  border-bottom: 1px grey solid;
  color:grey;
  font-weight: bold;
}

article.articleBarre{
  padding:10px;
  border-bottom: 1px grey solid;
  color:black;
  text-decoration: none;
  background-color: lightgrey;
}
```

Comme exercice supplémentaire, vous pouvez adapter les fichiers html et css de l'exercice 4 en remplaçant les multiples balises <div> du code par ces nouvelles balises.