

Les langages Web : HTML et CSS

I Introduction

Le *Web* (abréviation de l'anglais *World Wide Web*) est un ensemble de documents reliés entre eux par des *liens hypertexte*. On peut alors se déplacer d'un document à l'autre de manière interactive. On englobe dans le terme Web toute l'infrastructure hébergeant cette collection mondiale de document qui comprend :

- les langages informatiques permettant d'écrire les documents hypertextes, à savoir HTML et CSS.
- une architecture client-serveur utilisant le protocole HTTP (*Hypertext Transfert Protocol* ou sa version sécurisée HTTPS).
- des langages de programmation côté serveur (tels que PHP, Python, et bien d'autres) permettant de *générer* du HTML.
- des langages de programmation côté client (principalement JavaScript) dont le but est de permettre l'écriture d'application Web hautement interactives.

Cette infrastructure s'appuie sur le réseau internet, HTTP étant l'un des nombreux protocole permettant son fonctionnement. Nous nous intéresserons particulièrement ici aux langages HTML et CSS, prérequis pour la future conception des interactions client-serveur.

II HTML

II.1 Premier exemple et définitions

Le format HTML (*Hypertext Markup Language*) est un format textuel de **balisage** permettant de décrire le contenu et la structure d'un document Web. Prenons l'exemple ci-dessous. On y observe les différentes balises qui sont délimités par les caractères "<" et ">" comme par exemple <head> ou </h1>.

```
1 <!DOCTYPE html>
2 <html lang=" fr ">
3   <head>
4     <title>Le titre de la première page</title>
5     <meta charset=" utf-8" />
6   </head>
7   <body>
8     <h1>Première page !</h1>
9     <p>Notre toute première page !</p>
10    <!-- le reste du document peut venir ici -->
11  </body>
12 </html>
```

Première page !

Notre toute première page !

Pour afficher sa première page, il suffit d'écrire le code correspondant dans un fichier texte, et de le sauvegarder sous l'extension *.html*. Le résultat affiché dans le navigateur sera comme ci-dessus.

Une balise telle que <head> ne contenant pas d'antislash comme second caractère est appelée une *balise ouvrante*. A l'inverse, une balise telle que </title> est une *balise fermante*. Finalement, certains sont appelées *balise vide*, c'est le cas de la balise <meta charset=utf-8" />.

Il est possible de renseigner des attributs à l'intérieur d'une balise, c'est le cas des termes **lang** et **charset** des balises <html lang="fr"> et <meta charset="utf-8" />. On associe alors une valeur à ces attributs par le biais du symbole =.

Une paire de balise ouvrante et fermante, ainsi que son contenu situé entre les deux est appelé *élément*. Finalement, tous les informations situées entre les balises <!-- - - > sont des commentaires qui permettent de documenter le code.

II.2 Norme HTML et validation

Le format HTML est standardisé par le **W3C**¹, validateur en ligne qui permet de vérifier un fichier HTML.

Les règles décrivant les documents HTML sont complexes et très détaillées, mais nous pouvons pour le moment le réduire à une structure simple. On retrouvera toujours dans un premier temps une déclaration de type (tel que `<html>` et `</html>`) pour définir la portée de notre document HTML. A l'intérieur de cette balise, on retrouvera toujours les deux sous-éléments *head* et *body*.

- L'élément **head** représente les entêtes du document. ce dernier contient au moins un élément **title** contenant du texte.
- L'élément **body** représente le corps du document, et regroupe tous les éléments qui structurent la page.

Toutes les balises à l'intérieur seront écrites en minuscule, et positionnées tel qu'elles correspondent à leur niveau d'imbrication. Par exemple, la séquence `<i> </i>` n'est pas correcte, il est nécessaire de fermer la balise `<i>` par une balise `</i>` avant de fermer la balise ``.

II.3 Structuration des documents

Passons en revue un certain nombre de balise qui nous permettra de structurer une page Web. La liste n'est pas exhaustive, mais va nous permettre d'aborder quelques uns des éléments et attributs importants de l'HTML.

Titre de section

Les balises **h1**, **h2**, **h3**, **h4**, **h5** et **h6** permettent de définir des titres de sections jusqu'à six niveaux.

Structuration du texte

La balise `<p>` représente un paragraphe. Les espaces et les retours à la ligne du document sont ignorés. Chaque nouvel élément **p** créé un nouveau paragraphe, insérant un retour à la ligne. Si l'on souhaite ajouter un retour à la ligne dans un paragraphe, on peut utiliser la balise `
` (qui est une balise vide).

On peut organiser le texte en listes énumérées ou non. On utilisera respectivement les balises `` (pour *Ordered list*) et `` (pour *unordered list*). Chaque élément de la liste sera ensuite contenu dans un élément ``.

```
1 <ul>
2   <li>Un objet</li>
3   <li>Un autre objet</li>
4 </ul>
5 <ol>
6   <li>Point numéro 1</li>
7   <li>Point numéro 2</li>
8 </ol>
```

On peut organiser du texte dans un tableau à deux dimensions au moyen de l'élément **table**, qui contient l'ensemble des lignes, chacune étant un élément **tr** (qui signifie *table row*). Chacune des cases d'une ligne est décrit par la balise `<td>`.

```
1 <table>
2   <tr>
3     <td>Cellule 1</td> <td>Cellule 2</td> <td>Cellule 3</td>
4   </tr>
5   <tr <td rowspan="2">Deuxième ligne</td> </tr>
6   <tr <td>Val 1</td> <td>Val 2</td> </tr>
7 </table>
```

1. accessible à l'adresse <https://validator.w3.org>

Ici, la deuxième ligne est fusionnée à l'aide de l'attribut *rowspan* et la valeur 2 (fusion sur 3 cases).

Balises de texte

Les balises de texte n'interrompent pas le *flot* du texte (à l'inverse des balises de structure). Ces balises permettent d'indiquer la nature du texte. En voici quelques unes.

- `` met en gras le texte.
- `<i>` met en italique le texte.
- `<u>` met en souligné le texte.
- `<mark>` surligne le texte.
- `<small>` affiche le texte en petits caractères.
- `<code>` est utilisé pour afficher un code informatique.
- `<sub>` met le texte en indice.
- `<sup>` met le texte en exposant.

Liens et ressources

La fonctionnalité principale des documents HTML est la possibilité d'utiliser des *liens hypertextes*. Ces derniers sont insérés dans du texte grâce à la balise `<a>` qui se découpe en deux parties. On ajoute en premier l'attribut **href** qui renseigne l'adresse de la ressource, puis on ajoute le contenu de la balise qui sert à l'affichage du lien

```
1 <a href="mon_document.html"> Le texte servant à pointer vers l'adresse du document </a>
```

Il est aussi possible d'insérer directement dans la page des ressources externes. C'est le cas des images que l'on peut insérer avec la balise `` qui doit contenir deux attributs, **src** qui indique l'emplacement de l'image et **alt** qui renseigne la légende.

Balises neutres, classes et identifiants

Il existe en HTML deux balises neutres, qui n'ont donc pas de rôle particulier.

- `<div>` permet de structurer le document et qui a un comportement par défaut identique à la balise `<p>`.
- `` est une balise de texte, comme `` ou `<i>`, mais qui laisse par défaut le texte inchangé.

Pour chacune de ces balises, il est possible d'ajouter deux attributs **id** et **class**. L'attribut **id** représente l'identifiant et doit être unique dans tout le document. De son côté, l'attribut **class** contient une suite de *noms de classes* séparés par des espaces. Ces deux étiquettes sont utilisées dans la définition du style graphique des éléments

```
1 <h1 id="titre">A faire</h1>
2 <ul>
3   <li class="important">Acheter un vélo</li>
4   <li class="rendez_vous">Aller chez le decteur</li>
5 </ul>
```

III CSS

Le langage *CSS* (pour *Cascading Style Sheets* est le langage de prédilection pour la mise en forme du texte sur un document HTML.

Un exemple

```
1 <p style="text-align:right;"> On écrit un <span style="border:1pt
  solid black;"> paragraphe </span>de texte , mais cette fois , <
  span style="color:gray;"> modifie </span> le style graphique <
  span style="border:1pt solid black;"> des éléments </span>
2 </p>
```

On écrit un paragraphe de texte,
mais cette fois, modifie le style
graphique des éléments

On a ajouté ici à plusieurs endroits l'attribut **style** qui permet de définir des propriétés *CSS*. On a précisé dans la balise `<p>` que le texte sera aligné à droite. L'attribut **style** est ensuite appliqué dans trois `` différents, pour y préciser une mise en forme du texte particulière. Néanmoins, l'écriture dans le code HTML le rend illisible, on lui préférera une rédaction dans un document spécifique qui sera appelé par la page Web. Ajoutons alors ce document *CSS*.

```
1 <p> On écrit un <span class="encadre"> paragraphe </span>
  >de texte , mais cette fois , <span id="gris"> modifie
  </span>le style graphique <span class="encadre">
  des éléments </span>
2 </p>
```

```
1 p { text-align: right; }
2 .encadre { border:1pt solid
  black; }
3 #gris { color: grey; }
```

Le résultat attendu lors de l'affichage est le même, mais toutes les fonctionnalités de mise en forme du texte sont regroupé dans un document unique appelé ici *mon-style.css*. Décrivons les 3 lignes de la feuille CSS :

- **p text-align : right ;** précise que *toutes les balises <p>* ont leur texte aligné à droite.
- **.encadre border :1pt solid black ;** décrit que, dès lors que le mot *encadre* est présent dans l'attribut **class**, celui-ci sera encadré par un trait plein noir.
- **#gris color : grey ;** stipule que lorsque l'attribut **id** vaut *gris*, le texte associé sera écrit en cette couleur.

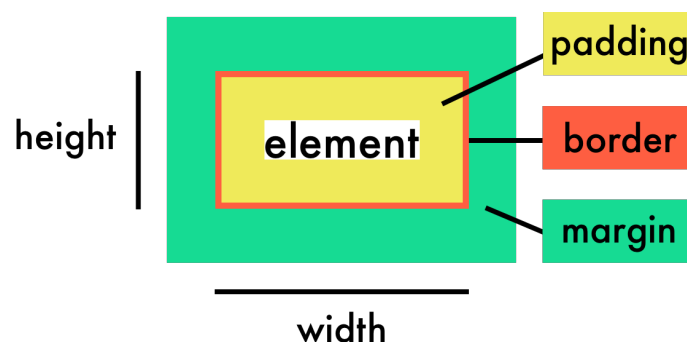
La feuille de style est finalement ajouté au document Web dans l'en-tête (balise `<head>`) par le biais de la ligne ci-dessous :

```
1 <link href="mon-style.css" rel="stylesheet" type="text/css" />
```

III.1 Quelques propriétés CSS

Modèle de boîte

Tout comme HTML, CSS est standardisé par le W3C. La spécification est découpée en différents *module* traitant d'aspects spécifiques : couleurs, sélecteurs, dispositions des éléments, etc. Prenons par exemple l'affichage d'un contenu "élément".



Les trois attributs *margin*, *border* et *padding* sont autant d'éléments que l'on pourra modifier dans la mise en forme d'un document.

longueurs et couleurs

De nombreuses propriétés CSS indiquent des longueurs. Les valeurs sont toujours un nombre suivi de son unité, tel que le pixel (*px*) ou le point (*pt*) comme on peut le retrouver dans les éditeurs de texte. Il est aussi possible de spécifier une proportion à l'aide du symbole %.

Pour ce qui concerne les couleurs, on peut utiliser les noms en anglais (red, blue, grey, black, green, etc.) ou le code exprimé en hexadécimal. Pour renseigner une couleur précise, on précédera la valeur du symbole #. Par exemple, pour modifier la couleur du fond de la totalité d'un document, on pourra écrire dans notre fiche de style :

```
1 body { background-color: #660066; }
```

III.2 Sélecteurs

Dans une feuille de style, une règle CSS est donnée sous la forme suivante :

```
1 p1 p2 ... pn { propriété_1: valeur_1;
2   propriété_2: valeur_2;
3   ...
4   propriété_n: valeur_n; }
```

la partie p1 ... pn est le sélecteur CSS qui représente les différentes étapes (appelés "pas"). L'étape **p1** sélectionne tous les éléments qui vérifient ces conditions. L'étape **p2** sélectionne tous les éléments qui vérifient p2 et qui sont *contenu* dans les balises des éléments qui vérifiaient précédemment p1. Le processus se répète jusqu'à la fin.

On peut ainsi combiner nom de classe et nom de balise en les concaténant sans espaces. Prenons l'exemple ci-dessous :

```
1 div#menu li a { color: pink; background-color: yellow; }
```

On indique ici que l'on souhaite trouver tous les liens (balise <a>) qui sont à l'intérieur d'éléments , qui sont eux-mêmes à l'intérieur de l'unique balise **div** dont l'identifiant est **menu**.



Attention

"div#menu" et "div #menu" ne retourne pas le même résultat. Dans le premier cas, il est nécessaire de se trouver dans une balise **div** dont le nom est **menu**. Dans le second cas, on sélectionne une sous-balise **menu** contenue dans une balise **div**.

A retenir

Savoir-faire

Le langage **HTML** est un langage de description des pages Web. Ces dernières sont des documents structurés par des **balises** suivant un ensemble de règles de **bonne formation**. Après avoir défini la structure d'une page, on peut définir sa charte graphique au moyen du langage **CSS** qui permet d'ajuster finement toutes les caractéristiques d'une page Web. Pour toutes informations complémentaires, W3C propose un site très complet pour les balises et propriétés des langages **HTML** et **CSS**

<https://jigsaw.w3.org>



Exercice 1

Pour chacun des fragments de documents ci-dessous, indiquer s'ils sont valides. S'ils ne le sont pas, dire pourquoi.

```
1 <i><u>du texte</u></i>           1 <p><u>du texte</i></p>
```

```
1 <b class="a"
2   class="b">
3   en gras </b>           1 <head>
2   <b>Bonjour</b>
3 </head>
```

```
1 <p />           1 <h1 id="t1">Titre</h1>
2 <h1 id="t1">Titre</h1>
```



Exercice 2

Ecrire un programme Python qui crée un fichier HTML **table.html** contenant les tables de multiplication de 1 à 10. Chaque table est précédée d'un titre de niveau 1, suivi de la table dans une liste non énumérée. Par exemple, la portion du document correspondant à la table de 1 sera :

```
1 <h1>Table de 1</h1>
2 <ul>
3   <li>1 * 1 = 1</li>
4   <li>1 * 2 = 2</li>
5   ...
6 </ul>
```

Attention, on doit simplifier le code pour générer le plus automatiquement possible nos tables.



Exercice 3

Créer un fichier **damier.css** contenant les règles suivantes :

```
1 body {background: grey;}
2 table {border-collapse: collapse;}
3 td {
4   width: 40pt;
5   height: 40pt;
6 }
7 td.blanc {background: white;}
8 td.noir {background: black;}
```