

Introduction à HTML et CSS

0 But de l'atelier

Le but de cet atelier est de comprendre un peu mieux le HTML pour faire des pages Web basiques, et un peu de CSS pour donner une apparence correcte à nos pages.

Pendant cet atelier, nous allons apprendre les bases pour mettre en forme notre carte interactive, pour arriver à un résultat similaire à celui-ci, après l'exercice:

Le campus de l'UNIL



1 Rappel du dernier atelier

Lors du dernier atelier, nous avons déjà vu quelques éléments HTML. Voici les principales balises que nous avons vu:

<code><html>...</html></code>	L'ensemble du document HTML
<code><head>...</head></code>	Les définitions du document HTML
<code><body>...</body></code>	Le corps du document, donc la partie visible dans le navigateur Web
<code><h1>...</h1></code>	Titre de niveau 1
<code><h2>...</h2></code>	Titre de niveau 2
<code><h7>...</h7></code>	Titre de niveau 7
<code><p>...</p></code>	Un paragraphe de texte
<code><i>...</i></code>	Un texte en italique
<code>...</code>	Un texte gras (bold)
<code></code>	Une image où src indique le lien vers le fichier de l'image (JPEG, GIF ou PNG seulement)
<code>...</code>	Un lien vers l'URL indiquée dans href
<code><div>...</div></code>	Un conteneur vide, permet de regrouper un certain nombre d'autres éléments HTML. Est aussi utilisé pour donner un style commun aux éléments à l'intérieur.

Et voici le document HTML que nous avons créé pour afficher notre carte interactive (sans le code Javascript pour créer la carte):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Ma première carte interactive</title>
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.2.0/dist/leaflet.css"/>
  <script src="https://unpkg.com/leaflet@1.2.0/dist/leaflet.js"></script>
</head>
<body>
  <h1>Ma carte interactive ou quelque chose de plus intelligent</h1>
  <div id="map" style="width: 600px; height: 450px;"></div>
  <script type="text/javascript">
    ...
  </script>
</body>
</html>
```

2 Un peu plus de HTML...

Jusqu'à maintenant, nous avons déjà vu les balises HTML principales. Même s'il y a en a encore beaucoup d'autres balises (dont la plupart relativement peu utilisées), nous allons en voir seulement 5 car ça suffit pour la grande majorité des situations. Il s'agit de **table** (avec quelques balises pour les lignes et cellules), **ul** et **ol** (pour les listes), **input** (pour les champs de textes, cases à cocher etc.), **button** et **span**.

2.1 Tableaux

La balise **table** permet de définir des tableaux de manière suivante:

```
<table border="1" cellspacing="3" cellpadding="5">
  <tr>
    <td>Ligne 1, colonne 1</td>
    <td>Ligne 1, colonne 2</td>
    <td>Ligne 1, colonne 3</td>
  </tr>
  <tr>
    <td>Ligne 2, colonne 1</td>
    <td>Ligne 2, colonne 2</td>
    <td>Ligne 3, colonne 3</td>
  </tr>
</table>
```

avec le résultat suivant:

Ligne 1, colonne 1	Ligne 1, colonne 2	Ligne 1, colonne 3
Ligne 2, colonne 1	Ligne 2, colonne 2	Ligne 3, colonne 3

La balise **tr** crée une ligne du tableau, tandis que **td** représente une cellule. Les colonnes sont donc définies de manière indirecte, à travers le nombre de balises **td** dans chaque colonne.

L'attribut **border** définit la bordure noire (mettre sur 0 pour faire un tableau sans bordure), **cellspacing** définit l'espace entre les cellules (en pixels), et **cellpadding** l'espace à l'intérieur de la cellule entre le bord et le contenu (en l'occurrence du texte, mais ça peut vraiment être n'importe quoi).

Les tableaux sont utiles typiquement pour des boîtes d'info pour donner les propriétés d'un élément sélectionné sur la carte. Ainsi, nous pouvons créer une boîte d'info pour le bâtiment Géopolis sélectionné sur la carte (en dessous du **div** de notre carte par exemple):

```
<div id="infobox">
  <table cellpadding="3">
    <tr>
      <td><b>Bâtiment:</b></td>
      <td>Géopolis</td>
```

```

    </tr>
    <tr>
      <td><b>Facultés:</b></td>
      <td>
        <a href="http://unil.ch/gse">GSE</a>,
        <a href="http://unil.ch/ssp">SSP</a>
      </td>
    </tr>
  </table>
</div>

```

Vous pouvez mettre ce code déjà maintenant dans votre carte interactive. Nous allons introduire les bâtiments plus tard, ainsi que le code Javascript nécessaire pour mettre à jour le contenu de ce tableau. Notez que nous avons placé le tableau dans un **div** qui possède une **id**. Ce n'est pas obligatoire, mais plutôt une bonne pratique qui facilite la mise en page qui est faite plus tard.

Le placement sur la page HTML n'est bien évidemment loin d'être idéal pour le moment. Nous allons corriger cela plus tard.

2.2 Listes

HTML possède trois tags liés aux listes:

- **ul** (unordered list) pour une liste à puces
- **ol** (ordered list) pour une liste numérotée
- **li** (list item) pour les éléments des listes

Voici deux exemples:

```

<ul>
  <li>Élément 1</li>
  <li>Élément 2</li>
</ul>

```

```

<ol>
  <li>Élément 1</li>
  <li>Élément 2</li>
</ol>

```

Nous pouvons donc utiliser une liste p.ex. pour la liste des facultés dans le tableau ci-dessus:

```

...
<td style="vertical-align: top;"><b>Facultés:</b></td>
<td>
  <ul>
    <li><a href="http://unil.ch/gse">GSE</a></li>
    <li><a href="http://unil.ch/ssp">SSP</a></li>
  </ul>
</td>
...

```

2.3 Inputs et boutons

La balise **input** permet de créer des champs de texte, cases à cocher et boutons radio. La balise **select** représente des menus déroulants, tandis que la balise **button** est un bouton simple qui permettra de déclencher une action (en Javascript).

Un champ de texte est défini comme ça:

```
<input type="text" name="mon-champ-de-texte" value="" />
```

La balise **input** n'a pas de balise de fermeture. Le contenu du champ de texte se trouve dans l'attribut **value**.

Les cases à cocher sont définies de manière similaire, à part pour l'attribut **type**:

```
<input type="checkbox" name="ma-case-a-cocher" checked /> Étiquette de la case
```

Cette balise possède une nouvelle particularité qui est l'attribut **checked** qui n'a pas de valeur. Ces attributs sans valeurs sont très rares, mais en voilà un... Il permet simplement de cocher la case.

Bien évidemment, une case à cocher a besoin d'une étiquette séparée pour savoir de quoi il s'agit.

Les cases à cocher sont carrées et elles ont deux statuts: cocher ou non. Les boutons radios sont ronds, et permettent de sélectionner une option unique parmi toute une série d'options. Il y a toujours plus d'un bouton radio (un bouton radio par option), et un seul peut être sélectionné. Voici un exemple d'un groupe de 3 boutons radios:

```
<input type="radio" name="mon-bouton-radio" value="1" checked /> Option 1  
<input type="radio" name="mon-bouton-radio" value="2" /> Option 2  
<input type="radio" name="mon-bouton-radio" value="3" /> Option 3
```

C'est en donnant une valeur identique à l'attribut **name** que nous pouvons grouper les boutons radios.

Finalement, nous avons des boutons simples que nous pouvons créer avec **button**:

```
<button>Chercher</button>
```

ce qui affiche un bouton qui ne fait rien. Pour déclencher une action, il faut définir un peu de code Javascript à l'intérieur de l'attribut **onclick**. En voici un exemple très simple qui nous suffira pour le moment:

```
<button onclick="alert('Trouvé!');">Chercher</button>
```

Ceci nous permet de créer un champ de texte et un bouton pour la recherche d'éléments dans notre carte interactive:

```
<div id="recherche">  
  <input type="text" name="recherche" placeholder="Rechercher dans la carte" />  
  <button onclick="alert('Trouvé!');">Chercher</button>  
</div>
```

2.4 La balise span

La balise span permet de modifier le style d'un morceau de texte, p.ex.

```
<p>Ceci est un texte normal, sauf pour <span style="color: #f00;">une partie</span> qui est rouge.</p>
```

3 CSS

C'est le code CSS qui permet d'appliquer des styles aux balises HTML et ainsi définir en grande partie comment une page Web se présente. C'est une tentative de dissocier le contenu d'une page Web (encodé dans le HTML) du style (dans le CSS).

Nous avons trois possibilités d'inclure du CSS dans notre page Web:

1. Écrire le code CSS dans un fichier à part (fichier avec extension .css), et l'intégrer dans notre fichier HTML principal à l'aide de la balise **link**:
<link rel="stylesheet" href="style.css" />
Contrôlez alors la console du navigateur Web pour voir si le fichier style.css a été effectivement chargé.
2. Nous pouvons écrire le code CSS à l'intérieur de l'attribut **style** d'une balise HTML. Ceci est surtout utile pour rapidement essayer un style. Dans la pratique, il est déconseillé d'utiliser l'attribut **style** pour avoir une meilleure séparation entre contenu et style.
3. Il est théoriquement possible d'introduire une balise HTML **style** dans le **head** du document HTML et d'y ajouter le CSS. Cette méthode présente les inconvénients des deux méthodes précédentes et aucun avantage. C'est donc une méthode à éviter...

Nous allons maintenant voir quelques instructions CSS, avant de mettre en forme notre carte interactive.

3.1 Propriétés du texte

font-family permet de définir la police de caractères. Étant donné qu'il est impossible de savoir quelles polices sont installées sur l'ordinateur de la personne qui regarde la page Web, on va définir plusieurs en ordre décroissant de priorité. Voici une définition typique:

```
font-family: "Helvetica Neue", Helvetica, Arial, Verdana, sans-serif;
```

Le texte sera donc en Helvetica Neue si disponible, sinon en Helvetica si disponible, etc. À la fin il y a l'instruction générique sans-serif qui caractérise le type de police à utiliser. Il y a en effet les polices **sans-serif** et **serif**; les dernières présentent de petits «pieds» (comme par exemple Times ou Garamond) pour faciliter la lecture.

font-size définit la taille de la police de caractères, normalement en points, par exemple **font-size: 12pt;**

font-weight définit le «poids» des polices, et prend normalement les valeurs **normal** ou **bold**.

text-align définit l'alignement du texte et peut prendre les valeurs **left**, **center**, **right** ou **justify**.

color définit la couleur du texte en code hexadécimal (**#008ccc** pour le bleu de l'UNIL p.ex.).

line-height est pour régler l'intervalle des lignes, en points: **line-height: 18pt;**

3.2 Liens

Les liens peuvent être formatés comme n'importe quel texte, sauf qu'ils peuvent prendre des états différents, soit en passant avec la souris dessus, ou après les avoir visités. Pour définir le style d'un lien où la souris est actuellement dessus, nous utilisons dans notre fichier CSS:

```
a:hover {  
    ...  
}
```

Et pour un lien après avoir visité la destination:

```
a:visited {  
    ...  
}
```

3.3 Couleur de fond, bordures et marges

background-color définit la couleur de fond d'un texte, ou de n'importe quel autre élément HTML. La couleur est à nouveau en format hexadécimal.

border permet de créer une bordure autour d'un élément HTML. La bordure est composée de l'épaisseur du trait en pixels, le type de trait (normal ou ligne pointillée), ainsi que la couleur. Dans la pratique cela donne un code comme:

border: 1px solid #000000;

pour obtenir une bordure noire de 1 pixel. À la place de **solid**, on peut aussi mettre **dashed** ou **dotted**.

Il est possible de régler les bordures de chaque côté de manière indépendante, en utilisant les propriétés **border-top**, **border-bottom**, **border-right**, **border-left**, p.ex. **border-top: 1px solid #ff0000;**

margin définit la marge extérieure autour d'une balise HTML. Cette marge peut être donnée en pixels ou en points (**margin: 10px;** ou **margin: 10pt;**). Donc par exemple

si nous avons deux balises **div** l'une sur l'autre, **margin** permet d'espacer les deux éléments.

On peut régler séparément les marges de chaque côté. Il y a deux possibilités pour cela. La première consiste à donner les 4 marges en même temps:

margin: 10px 5px 10px 0;

ce qui définit les marges dans l'ordre haut, droite, bas et gauche (dans le sens des aiguilles d'une montre en partant du haut). La deuxième variante permet de modifier uniquement une des marges à la fois, en utilisant les propriétés CSS **margin-top**, **margin-bottom**, **margin-right** ou **margin-left**, donc p.ex. **margin-top: 10px;**

padding fait la même chose que **margin**, sauf pour la marge intérieure d'un élément. Bien souvent, l'effet est le même que pour **margin**, mais pas toujours. Par exemple, si vous avez une balise **div** avec bordure, vous voudrez pouvoir modifier indépendamment la marge entre la bordure et les éléments autour de la marge entre la bordure et le contenu du **div**. Essayez donc par exemple le code suivant:

```
<div style="border: 1px solid #ff0000;>Bonjour</div>
<div style="margin: 10px; padding: 20px; border: 1px solid #000000;">à tous</div>
```

et jouez un peu avec le **margin** et le **padding**. Vous allez très vite comprendre et apprécier la différence.

3.4 Définir la position à l'écran

Les propriétés **width** et **height** permettent de définir la largeur et hauteur d'un élément. Nous les avons déjà rencontré dans le **div** de la carte interactive:

```
<div id="map" style="width: 500px; height: 300px;"></div>
```

En principe, il est aussi possible de définir ces valeurs en pourcentages de la taille de la fenêtre:

```
<div id="map" style="width: 80%; height: 80%;"></div>
```

Parfois il peut être nécessaire de placer un élément exactement à une position donnée. Ceci peut être faite avec la position absolue:

```
<div style="position: absolute; left: 10px; top: 10px; width: 100px; height: 200px;">hello</div>
```

Les propriétés **left** et **top** indiquent alors la position du coin supérieur gauche dans la fenêtre du navigateur.

Le contraire de **position: absolute;** est **position: relative;**

3.5 Classes CSS

Une classe CSS est un ensemble de propriétés CSS auquel nous donnons un nom. Par la suite, nous pouvons appliquer ces propriétés à n'importe quel balise HTML en insérant l'attribut **class**. Voici un extrait d'un fichier HTML, comme exemple:

```
<p class="intro">Utilisez cette carte pour vous orienter à l'UNIL.</p>
<p>Elle a été créée par <span class="intro">Léo</span>.</p>
```

et puis dans notre fichier CSS nous définissons à la fois le style du texte pour tous les paragraphes, et puis pour la classe intro:

```
p {
  font-family: Helvetica, sans-serif;
  font-size: 10pt;
  color: #000000;
  margin: 0;
}
.intro {
  font-weight: bold;
  color: #990000;
}
```

Notez le point avant intro; c'est lui qui dit qu'il s'agit d'une classe CSS et non d'un nom de balise! Voici le résultat:

Utilisez cette carte pour vous orienter à l'UNIL.
Elle a été créée par **Léo.**

Nous pouvons également donner plus d'une classe à un élément HTML en les séparant avec un espace, pour appliquer les deux styles à la fois:

```
<p class="intro cool">...</p>
```

Il est également possible de faire de définitions de style un peu plus complexes dans notre fichier CSS. Si nous avons par exemple le code HTML suivant:

```
<div class="info">
  <p><b>Bâtiment:</b> Géopolis</p>
  <p><b>Facultés:</b> GSE, SSP</p>
</div>
```

Maintenant nous pouvons définir le style pour toutes les balises **p** contenus dans la balise avec classe **info**, de la manière suivante:

```
.info p {
  font-size: 10pt;
}
```

Ce mécanisme permet de réduire le nombre d'attributs **class="..."** à devoir placer dans les balises HTML.

4 Exercice

Essayez de mettre en forme la carte interactive que nous avons commencé à faire la dernière fois. Pour avoir une idée du résultat qu'il faudrait idéalement obtenir, référez-vous à la première page de cet atelier.

Si vous n'y arrivez pas au premier coup, ce n'est pas très dramatique. Apprendre le HTML et le CSS prend du temps. Par contre, n'oubliez pas que c'est uniquement en faisant qu'on apprend réellement! C'est comme apprendre à nager, il faut essayer; lire la théorie ne suffit pas...

5 Librairies de composantes et de styles

Un dernier mot encore par rapport à ce que nous avons vu dans cet atelier. C'est rare dans la pratique qu'on part de zéro comme nous l'avons fait aujourd'hui. Souvent, nous allons recourir à une librairie déjà existante qui contient déjà des styles pré-faits pour nous, et qui contient des composantes typiques comme des icônes, styles de boutons etc. prêtes à l'emploi.

La librairie la plus répandue est probablement Bootstrap. Vous pouvez aller regarder un peu déjà maintenant pour voir ce qui est possible (<http://getbootstrap.com/>). Regardez p.ex. dans les «Components»; vous allez peut-être reconnaître certains des éléments que vous aurez déjà rencontré sur des sites Web...