

Mission 2- HTML / CSS avancé

dans le cadre du Projet 3

structuration et stylisation, block & inline, combinaison de sélecteurs

La spécification HTML5 propose différentes manières de classer les balises/éléments selon leurs caractéristiques. Nous allons ici nous intéresser à deux types spécifiques :

- les balises de structure : elles permettent de délimiter l'articulation logique de la page web en la découpant en différentes sections. Nous y consacrerons la prochaine section.
- les balises au niveau du texte : elles apportent une précision sur la sémantique d'une partie du texte (mise en avant d'une partie de texte importante, ajout d'un type exposant, time,...).

Voici quelques exemples de balises au niveau du texte :

- `<sup>` : un exposant
- `<time>` : mise en valeur d'une date
- `` : emphase sur du texte
- `` : on rajoute de l'importance à un texte
- `
` : saut de ligne

Ces balises au niveau du texte sont souvent naturellement liées à un style associé (les `` seront stylisées par une mise en italique, les `<sup>` en exposants,...). Les navigateurs se chargent d'ajouter pour nous certains styles par défaut très courant.

Il existe encore beaucoup d'autres balises HTML. Cela dit, il arrive qu'aucune ne corresponde à ce que l'on veut exprimer (lors d'une construction de layout par exemple). Deux balises neutres ont été ajoutées pour ces constructions :

- Au niveau du texte `` : cette balise est neutre, sans signification particulière. Son utilisation permet entre autres de créer des règles de formatage spécifiques du contenu textuel (par exemple lorsque nous avons ajouter la class `skill`).
- Balise de structure `<div>` : cette balise est neutre, son utilisation permet de distinguer une section qui ne revêt aucune signification particulière. Contrairement au `span` elle provoque un saut de ligne.

Structuration de la page

Vous allez d'abord structurer logiquement le contenu du site. Voici un *template* HTML (modèle HTML en français) d'une structuration classique de page Web.

```
<!DOCTYPE html>
```

```

<html>
  <head>...</head>
  <body>
    <header>
      ...
      <nav>...</nav>
    </header>
    <main>
      <article>...</article>
      <article>...</article>
      <aside>...</aside>
    </main>
    <footer>
      ...
    </footer>
  </body>
</html>

```

Pour fixer les idées, voici un aperçu d'une mise en page correspondante à l'exemple précédent :



Par défaut, ces balises découpent la page web en sections horizontales qu'elles occupent en entier (du bord gauche au bord droit de la section). Par défaut Les balises de structure s'empilent verticalement car elles ne peuvent pas partager une même section horizontale (donc nous devrons appliquer un style particulier pour `<aside>`).

Expliquons le rôle de quelques balises de structure:

- `<header>` : section contenant l'en-tête affichée de la page (à ne pas confondre avec `<head>`, qui sont les méta-informations de la page Web)
- `<nav>` : section contenant une série de liens hypertextes pour la navigation sur le site
- `<main>` : section principale de la page, celle qui contient le contenu spécifique à cette page. Cette balise ne peut être présente dans une autre balise présentée ici à l'exception de `<div>`.
- `<article>` : section contenant un document « auto-suffisant », *i.e.* qui peut être séparé du reste de la page et gardera cependant tout son sens. Une page peut contenir plusieurs articles, dans le cas d'un blog par exemple, de commentaires, d'une page listant les publications récentes ...
- `<aside>` : section contenant du matériel périphérique au contenu principal. Cela peut-être par ex. une série de liens spécifiques au document principal, un bandeau de publicités ...
- `<footer>` : section contenant le pied de page.
- `<section>` : une section non spécifique, ou une sous-section d'un article, d'un menu... Doit typiquement commencer par un titre `<hX>`.
- `<figure>` : une illustration (au sens large) « auto-suffisante » illustrant le document principal ou un article, et qui doit pouvoir être placée librement dans la page sans en altérer le déroulement (par ex. dans le texte, dans un appendice...)
- `<h1>`- `<h6>` : titres de section
- `<blockquote>` : une citation avec en particulier `<cite>` pour la référence de la citation.
- `<p>` : paragraphes de texte
- `<address>` : coordonnées de contact de l'auteur. Il ne peut y avoir qu'un bloc `<address>` par article et un pour le restant de la page

Exercice 1

1. Ajoutez une balise `<header>`. Son contenu sera à votre choix et une barre de navigation `<nav>` vide pour l'instant,
2. Ajoutez une balise `<main>`, une balise `<article>` et une balise `<aside>` comme dans le *template* précédent. Mettez l'ancien contenu de la page dans `<article>` sauf les deux dernières sections ("Les sites amis" et "Le Top 10 des derniers facts proposés") qui vont dans `<aside>`,
3. Ajoutez une balise `<footer>` qui contient le lien vers le retour au début du site,
4. Ajoutez dans la balise `<nav>` deux liens dans une structure de liste contenant : Un lien nommé "Accueil" qui pointe sur la page courante `index.html` et un nommé "Contact" qui pointe vers une future page `contact.html`,

5. Validez votre pages HTML sur le validateur <https://html5.validator.nu/>. (Faites-le systématiquement sans qu'on vous le demande :-).

Exercice2

1. Construire une page **contact.html** au même niveau que **index.html**. Elle doit contenir le même template HTML (patron HTML) que **index.html**. En particulier:
 - i. elle reprend les mêmes **<header>** et **<footer>** que **index.html**,
 - ii. elle appelle la même feuille de style CSS,
 - iii. elle définit son propre **<title>**.
2. Dans la partie **<main>** de la page, ajoutez
 - i. un titre **Adresse**
 - ii. l'image [contact.jpg](#) pour illustrer que nous sommes bien à l'écoute.
 - iii. Ajoutez l'adresse avec la balise **<address>** contenant :
 - iv. Lycée Auguste Blanqui
 - v. Email : nom.prenom@gmail.com
3. Validez votre page HTML.

À ce point, le travail de division du site n'a pas encore de résultat visuel marquant. C'est avant tout un travail de structuration logique qui permet au navigateur, ou à un moteur de recherche, de mieux comprendre votre page web. Nous verrons comment changer la mise en page globale dans les TDs suivants. Pour la suite du TD, nous allons ajouter du style aux éléments de la page courante.

Règles de compositions des CSS

Rappelons qu'une règle CSS est composé d'un sélecteur CSS et d'un bloc de déclaration composé de plusieurs paires propriété CSS / valeur. Un sélecteur CSS indique à quels éléments HTML s'applique le style.

À partir des sélecteurs de bases (de balise, de classe et d'identifiant) présentés, il est possible de créer des sélecteurs complexes. Par exemple, nous allons voir comment sélectionner les **<div>** ayant la classe **toto** et qui sont fils d'un élément d'identifiant **titi**.

Nous exposons dans cette section les principaux moyens de composer un sélecteur CSS complexe.

Regroupement

La première façon de composer des sélecteurs est le regroupement. Les trois règles suivantes :

```
h1 {color: red}
h2 {color: red}
h3 {color: red}
```

peuvent s'écrire :

```
h1,h2,h3 {color: red}
```

Combinaison

Pour préciser un élément, il suffit de concaténer (sans espaces entre eux) plusieurs sélecteurs de base (balise, classe ou identifiant). Par exemple :

```
div.toto
```

correspond au sélecteur des `<div>` qui ont la classe `toto`. Ou encore

```
.titi.toto
```

correspond aux éléments qui ont la classe `toto` **ET** `titi`.

Descendance

On veut pouvoir limiter une règle CSS à une sous partie de l'arborescence HTML, pour cela on utilise la relation de descendance.

Descendance directe (enfant)

La relation de descendance directe est signifiée par le caractère `>`. Par exemple,

```
#titi > .toto
```

sélectionne les éléments de la classe `toto` qui sont **enfant** (direct) de l'élément d'identifiant `titi`.

Descendance indirecte (descendant)

La relation de descendance indirecte est signifiée par le caractère d'espacement. Par exemple,

```
#titi .toto
```

signifie les éléments qui ont la classe `toto` qui sont **descendants** (direct ou indirect) de l'élément d'identifiant `titi`. Donc la différence avec `>` est qu'on n'est plus limité aux enfants puisqu'on intègre aussi les petits-enfants, arrière-petits-enfants ...

Notes :

- Vous pouvez passer directement à l'exercice que vous voulez en rentrant le numéro de l'exercice à la place du sélecteur.

Pseudo Classes

Une pseudo-classe est un moyen d'indiquer un état particulier de l'élément qui doit être sélectionné. Voici quelques exemples :

```
/* Style des liens <a> n'ayant pas été visités */  
a:link {color: yellow;}  
/* Style des liens <a> ayant été visités */  
a:visited {color: purple;}  
/* Style des liens <a> si la souris les survole */  
a:hover {text-decoration: underline;}  
  
/* un paragraphe qui est le premier fils de son père */  
p:first-child {color:red;}  
/* le troisième paragraphe */  
p:nth-child(3) {color: green;}  
/* les textes des éléments li impairs seront verts et les pairs rouges */  
li:nth-child(odd) {color: green;}  
li:nth-child(even) {color: red;}
```

Exercice 3

1. Enlevez le fait que (tous) les liens soient soulignés. Pour cela, allez vous renseigner sur [la propriété text-decoration](#).
2. Faites en sorte que les liens `<a>` visités apparaissent en bleu plus léger `#0088FF`.
3. Lorsque la souris passe sur un lien, lui donner la couleur orange.

Tableaux

L'élément `<table>` correspond à une structuration récurrente, qui sert à représenter un ensemble de données sous forme de colonnes et de lignes.

Les éléments `<table>`, `<tr>`, et `<td>`

L'élément `<table>` contient la table. La table est composée de ligne (l'élément `<tr>`) contenant des cellules (élément `<td>`).

L'élément `<th>`

Dans l'arborescence du document, un élément `<th>` doit être le fils d'un élément `<tr>`. Il représente une cellule en-tête (le titre d'une colonne ou le titre d'une ligne du tableau). Il peut être utilisé à la place d'un élément `<td>`.

Voici un squelette de table :

```
<table>
```

```

<tr>
  <th>Caract1</th>
  <th>Caract2</th>
  <th>Caract3</th>
  <th>Caract4</th>
</tr>
<tr>
  <td>Val1_1</td>
  <td>Val1_2</td>
  <td>Val1_3</td>
  <td>Val1_4</td>
</tr>
<tr>
  <td>Val2_1</td>
  <td>Val2_2</td>
  <td>Val2_3</td>
  <td>Val2_4</td>
</tr>
...
</table>

```

Exercice 4

1. Créez une table avec les sept noms de colonnes suivants : **Acteurs**, **Karaté**, **Taekwondo**, **Judo**, **Chun Kuk Do**, **Tangsudo**, **Ju-jitsu**. Cette table doit se trouver en bas de la page **index.html**, dans la partie complémentaire **<aside>**. Les noms doivent être contenus dans des balises ****.
2. Ajoutez la classe **skill** aux **** correspondant à des noms d'arts martiaux.
3. Ajoutez les six lignes suivantes (les nombres correspondent à la valeur de l'acteur dans l'art martial correspondant) :
 - Chuck Norris, 5, 5, 5, 5, 5, 5
 - Steven Seagal, 3, 5, 3, 2, 3, 5
 - Bruce Lee, 5, 3, 3, 3, 4, 3
 - Jean-Claude Van Damne, 5, 3, 3, 3, 4, 3
 - Bolo Yeung, 2, 4, 4, 2, 5, 3
 - Dolph Lundgren, 2, 4, 4, 2, 5, 3
4. Testez la conformité de votre site.

Les éléments **<thead>** et **<tbody>**

Les éléments **<thead>** et **<tbody>** servent à définir plus explicitement la structure de notre table:

- **<thead>** : la définition des colonnes (**Acteurs**, **Karaté**,...)

- **<tbody>** : le corps du tableau, c'est-à-dire les lignes (nos héros et leurs niveaux de compétence).

Exercice 5

1. Ajoutez ces balises pour englober ces deux parties (en oubliant pas leurs balises fermantes **</thead>** et **</tbody>**)
2. Testez la conformité de votre site.

À ce stade la structure de votre table reflète le sens que vous vouliez y mettre. Voyons maintenant comment la styliser.

Exercice 6

1. Définissez une couleur de fond **#00AAFF** pour la partie en-tête **thead** du tableau.
2. Donnez la couleur violette **#640051** au texte des skills dans le tableau sans modifier le style des éléments ayant la classe **skill** dans les paragraphes,
3. ajoutez une règle pour que le fond d'une ligne (*row*) sur deux du corps de la table apparaisse en blanc et l'autre avec la couleur **#CCC SANS** modifier de quelque façon le HTML

Attention : La ligne du **<thead>** doit rester bleue.

Les attributs **rowspan** et **colspan**

Les balises **<th>** et **<td>** peuvent prendre des attribut **rowspan** et/ou **colspan**, qui permettent d'étirer la cellule courante pour prendre la place de plusieurs cellules :

- **rowspan** permet d'étirer la cellule sur plusieurs lignes (i.e rows),
- **colspan** permet d'étirer la cellule sur plusieurs colonnes.

Exercice 7

Il apparaît que Chuck Norris est toujours au top (niveau 5) dans tous les martiaux.

1. Faites une cellule qui prend toute la largeur de manière à mettre cela encore plus en exergue.
2. Mettez le 5 de Chuck en avant avec une balise **** pour bien montrer qui est le patron.
3. (Optionnel) Si vous souhaitez centrer le 5.

Le modèle de boîte

Comme vous l'avez vu précédemment, les balises de type structure définissent des boîtes. Ces boîtes disposent toutes des propriétés CSS suivantes :

- **margin** : marge à l'extérieur de la bordure, entre cette boîte et la suivante, et/ou entre cette boîte et son parent. La zone couverte par la marge est de la même couleur que son parent,
- **border** : bordure qui entoure le contenu. Cette propriété attend trois valeurs :
 1. **width**, par ex. **1px**,
 2. **style**, par ex. **solid**, **dotted**, **dashed**, ...
 3. **color**, par ex. **black**.

Attention : Un border n'a pas de style par défaut, donc lui donner une width ne suffit pas pour le voir.

- **padding** : marge intérieure à la bordure, c'est-à-dire espacement entre le contenu et la bordure de la boîte. Le padding partage le même arrière plan (**background-color**) que la boîte,
- **width** : la largeur du contenu, *i.e.* de la boîte *content*
- **height** : la hauteur du contenu, *i.e.* de la boîte *content*



Par exemple le code suivant

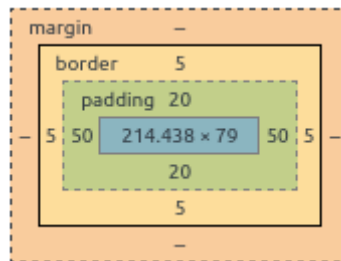
```
.maboite {
  padding:20px 50px 20px 50px;
  border:5px solid green;
  background-color:gold;
}
```

s'affiche comme ceci.

La zone de contenu

Exercice 8

Inspectez la boîte ci-dessus pour voir le style qui y est appliqué. Trouvez dans la partie Style des outils de développement le modèle de boîte (comme ci-dessous) et inspectez les quatre boîtes *content*, *padding*, *border* et *margin* pour les voir se dessiner à l'écran.



Il y a trois syntaxes différentes pour donner des valeurs au **margin**, au **padding** et au **border** :

- **margin : t r b l ;** : Si on donne 4 tailles **t**, **r**, **b** et **l**, alors **t** est associé à la valeur du haut (top), **r** est la valeur droite (right), **b** au bas (bottom) et **l** à la gauche (left);
- **padding : v h ;** : Si on ne donne que 2 tailles **v** et **h** alors **v** est associé aux valeurs verticales et **h** horizontales. C'est donc équivalent à **padding: v h v h ;**.
- **padding : a ;** Si on donne une seule valeur, elle sera associée aux quatre coté de la boîte, comme si on avait écrit **padding: a a a a ;**.

Exemples :

```
#titi { margin : 5px 0 4px 7px; }
/* Marges verticales (haute et basse) de 10px et horizontales de 5px */
div { margin: 10px 5px; }
/* Le padding dans toutes les directions est de 5px */
.toto { padding : 5px }
```

Note : On peut aussi préciser (péniblement) les valeurs unitaires des propriétés **margin-top**, **margin-left**, **margin-bottom**,...

Exercice 9

1. Ajoutez du padding vertical de **10px** aux titres de sections,
2. Ajoutez du margin vertical de **30px** aux paragraphes,
3. Ajoutez du padding horizontal de **5px** aux éléments ayant la class **skill** dans la table (mais pas aux éléments ayant la class **skill** dans les paragraphes).
4. ajoutez une bordure aux titres **<h3>** de **1px**, de style **solid** et de couleur **#CCCCCC**.

Centrer horizontalement :

Pour centrer le contenu d'une balise :

- si l'on veut centrer du texte (ou une balise au niveau du texte) dans une balise : **text-align: center**
- si le contenu est lui-même dans une balise de structure moins large que la balise parent : **margin : auto** sur la balise de structure.

Exercice 10

1. Centrer le **body** horizontalement,
2. Dans la table, centrer le texte des cellules, (le 5 de Chuck notamment est encore trop discret)

Les contenus flottant

La propriété **float** associée à un élément permet de faire flotter ce dernier complètement à gauche ou à droite de la ligne où il se trouve. Les valeurs de la propriété float sont **left**, **right**, **none** et **inherit**.

Exercice 11


1. Placez l'image de Chuck jeune à gauche du texte (comment faire un sélecteur uniquement pour cette balise ?),
2. Placez l'image beware **beware_img** à droite du texte.

Exercice 12

1. Rajoutez un nouveau paragraphe qui commence à la phrase "Spécialiste en arts martiaux, ..." de la section "L'enfance".

Vous devez alors avoir le rendu suivant :

L'enfance



Carlos Ray Norris, plus connu en tant que Chuck Norris, est un acteur américain né le 10 mars 1940 à Ryan (Oklahoma).

Spécialiste en arts martiaux, il s'est essentiellement consacré au cinéma d'action. Natif de Ryan dans l'Oklahoma, Carlos Ray Norris a deux frères cadets dont l'un est le producteur hollywoodien Aaron Norris. Ses parents sont de souche amérindienne (cherokee) et irlandaise. Ses parents divorcent alors qu'il a 16 ans et il déménage en Californie avec sa mère et ses frères. C'est là qu'il finit ses études au lycée et se marie rapidement avec sa petite amie, Diane Holechek.

2. Nous souhaitons plutôt ce rendu :

L'enfance



Carlos Ray Norris, plus connu en tant que Chuck Norris, est un acteur américain né le 10 mars 1940 à Ryan (Oklahoma).

Spécialiste en arts martiaux, il s'est essentiellement consacré au cinéma d'action. Natif de Ryan dans l'Oklahoma, Carlos Ray Norris a deux frères cadets dont l'un est le producteur hollywoodien Aaron Norris. Ses parents sont de souche amérindienne (cherokee) et irlandaise. Ses parents divorcent alors qu'il a 16 ans et il déménage en Californie avec sa mère et ses frères. C'est là qu'il finit ses études au lycée et se marie rapidement avec sa petite amie, Diane Holechek.

Pour interdire à notre paragraphe d'avoir un élément flottant sur son côté gauche, rajoutez-lui la règle `clear:left`.

Note : On peut aussi interdire le côté droit avec `clear:right` et les deux en même temps avec `clear:both`.

Position

La propriété CSS `position` offre de nouvelles possibilités pour le positionnement des éléments. Ses valeurs sont :

- `static` : comportement normal (par défaut), l'élément est inséré normalement.
- `relative` : le reste de la page fait comme si l'élément était positionné "normalement". De son côté, l'élément est positionné *relativement* à la position où il aurait dû être. On voit donc un espace où l'élément aurait dû être en `position:static`.
- `absolute` : le reste de la page fait comme si l'élément n'existait pas. L'élément se positionne relativement à son plus proche ancêtre **positionné** (voir ci-dessous) ou sinon à `<body>` (si aucun ancêtre n'est positionné).
- `fixed` : le reste de la page fait comme si l'élément n'existait pas. L'élément se positionne relativement à la fenêtre d'affichage ; il paraît donc *fixé* lors d'un défilement de la page.



Un élément est dit **positionné** s'il a une position autre que `static` (qui est la valeur par défaut). Pour indiquer le décalage de position, on utilise les propriétés `top`, `left`, `right` et `bottom`. Par exemple, les propriétés

```
position:relative;
top:20px;
left:20px;
```

vont positionner un élément **20px** plus à droite et en bas qu'il aurait dû l'être.

Référence : [Mozilla Developer Network \(MDN\)](https://developer.mozilla.org/fr/)

Exercice 13

1. Testez votre compréhension des propriétés **position: relative; top: 20px; left: 20px;** précédente en les appliquant temporairement sur l'image **chuck-jeune.jpg**.
2. Ajoutez les icônes de réseaux sociaux  et  toujours positionnées en bas à droite de la fenêtre d'affichage l'une au-dessus de l'autre. Essayez aussi temporairement de les afficher tout en bas à droite du document.