

Conception de documents & interfaces



Partie 2

Sébastien VIALLEMONTEIL

Développeur Web

sebastien.viallemonteil@agrosupdiyon.fr

Balises HTML5, Listes, Tableaux



Balises HTML5

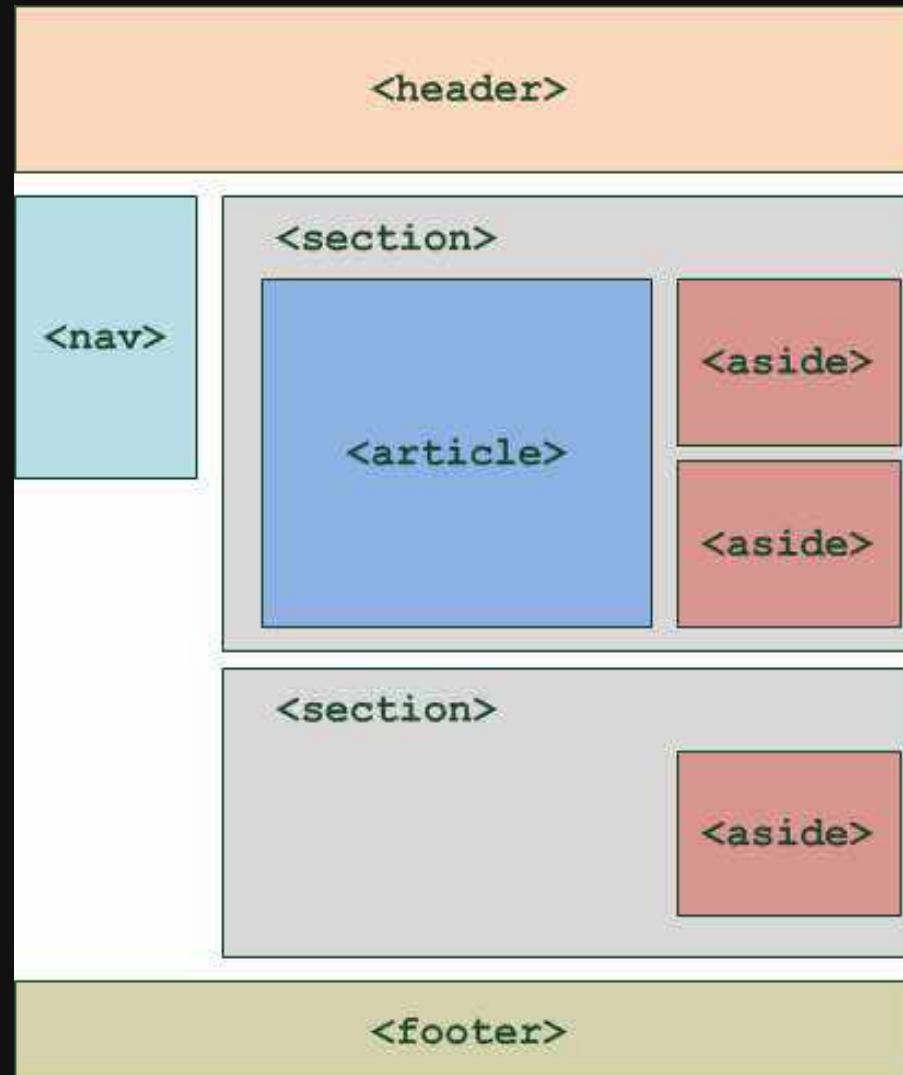
Nouvelles balises plus spécifiques à la structuration

- **HEADER** : l'en-tête de la page
- **FOOTER** : le pied de page
- **SECTION** : section de page (séparation de contenu)
- **ARTICLE** : portion autonome
- **NAV** : contient les liens de navigation (menu)
- **ASIDE** : informations complémentaires

Balises HTML5

```
< HTML />
1  <!DOCTYPE html> <!-- Doctype HTML5 -->
2  <html lang="fr">
3      <head>
4          <meta charset="UTF-8">
5          <title>Document</title>
6      </head>
7      <body>
8          <header> <!-- En-tête de page --></header>
9          <nav> <!-- Menu avec des liens --></nav>
10         <section>
11             <!-- Une section de page contenant un article-->
12             <article>...</article>
13             <aside>Une partie complémentaire</aside>
14         </section>
15         <footer> <!-- Pied de page --></footer>
16     </body>
17 </html>
```

Les balises HTML5



Source: OpenClassrooms

Les listes

Il existe 2 types de liste

1. Les listes à puces ou listes non ordonnées
2. Les listes ordonnées ou numérotées

Il est possible d'imbriquer des listes

Liste à puces

```
< HTML /> 1 <ul>
2     <li>Item 1</li>
3     <li>Item 2</li>
4     <li>Item 3</li>
5 </ul>
```

Liste à puces

- Item 1
- Item 2
- Item 3

Liste ordonnée

```
< HTML /> 1 <ol>
2     <li>Premier</li>
3     <li>Deuxième</li>
4     <li>Troisième</li>
5 </ol>
```

Liste ordonnée

1. Premier
2. Deuxième
3. Troisième

Liste particulière

Il existe une autre liste plus particulière qui permet d'écrire des définitions

```
< HTML />
1 <dl>
2   <dt>Terme 1</dt>
3   <dd>Définition 1</dd>
4   <dt>Terme 2</dt>
5   <dd>Définition 2</dd>
6   <dt>Terme 3</dt>
7   <dd>Définition 3</dd>
8 </dl>
```

Liste particulière

Terme 1

Définition 1

Terme 2

Définition 2

Terme 3

Définition 3

Les tableaux

Le langage HTML offre la possibilité de faire des tableaux simples (en-tête, lignes, bordures)

Mais aussi des tableaux plus complexes (fusion de colonnes, fusion de lignes, titre de tableau...)

Tableau simple

```
< HTML />
1 <table>
2   <caption>Titre du tableau</caption>
3   <tr>
4     <th>Titre 1</th>
5     <th>Titre 2</th>
6   </tr>
7   <tr>
8     <td>Val 1 ligne 1</td>
9     <td>Val 2 ligne 1</td>
10  </tr>
11  <tr>
12    <td>Val 1 ligne 2</td>
13    <td>Val 2 ligne 2</td>
14  </tr>
15 </table>
```

Tableau simple

TITRE DU TABLEAU

TITRE 1	TITRE 2
Val 1 ligne 1	Val 2 ligne 1
Val 1 ligne 2	Val 2 ligne 2

Tableau plus complexe

< HTML />

```
1 <table>
2   <!-- Titre du tableau -->
3   <caption>Titre du tableau</caption>
4   <!-- En-tête du tableau -->
5   <thead>
6     <tr>
7       <th>Titre 1</th>
8       <th>Titre 2</th>
9     </tr>
10  </thead>
11  <!-- Pied du tableau -->
12  <tfoot>
13    <tr>
14      <th>Titre 1</th>
15      <th>Titre 2</th>
16    </tr>
17  </tfoot>
```

Tableau plus complexe

```
< HTML />
1  <!-- Corps du tableau -->
2      <tbody>
3          <tr>
4              <td>Val 1 ligne 1</td>
5              <td>Val 2 ligne 1</td>
6          </tr>
7          <tr>
8              <td>Val 1 ligne 2</td>
9              <td>Val 2 ligne 2</td>
10         </tr>
11     </tbody>
12 </table>
```

Tableau plus complexe

TITRE DU TABLEAU

TITRE 1	TITRE 2
Val 1 ligne 1	Val 2 ligne 1
Val 1 ligne 2	Val 2 ligne 2
Titre 1	Titre 2

La fusion

2 possibilités

- La fusion de colonnes avec **colspan**
- La fusion de lignes avec **rowspan**

Colspan

```
< HTML />
1 <table>
2   <tr>
3     <th>Titre 1</th>
4     <th>Titre 2</th>
5     <th>Titre 3</th>
6   </tr>
7   <tr>
8     <td>Contenu 1</td>
9     <td>Contenu 2</td>
10    <td>Contenu 3</td>
11  </tr>
12  <tr>
13    <td>Contenu 1</td>
14    <td colspan="2">Contenu 2 & Contenu 3</td>
15  </tr>
16 </table>
```

Colspan

Titre 1	Titre 2	Titre 3
Contenu 1	Contenu 2	Contenu 3
Contenu 1	Contenu 2 & Contenu 3	

Rowspan

```
< HTML />
1 <table>
2   <tr>
3     <th>Titre 1</th>
4     <td>Contenu 1</td>
5     <td>Contenu 2</td>
6   </tr>
7   <tr>
8     <th>Titre 2</th>
9     <td>Contenu 1</td>
10    <td rowspan="2">Contenu 2 & Contenu 2</td>
11  </tr>
12  <tr>
13    <th>Titre 3</th>
14    <td>Contenu 1</td>
15  </tr>
16 </table>
```

Rowspan

Titre 1	Contenu 1	Contenu 2
Titre 2	Contenu 1	Contenu 2 & Contenu 2
Titre 3	Contenu 1	

A collection of vintage tools and objects arranged on a dark wooden surface. The items include a claw hammer, a double-bit axe, a hand saw, a pair of work gloves, a metal mug, a small bell, a tape measure, a screwdriver, a pair of pliers, and a small metal container. The scene is lit with warm, natural light from the left, creating soft shadows.

À vous de jouer maintenant

Écrire le code du tableau suivant:

P1	P2	P3	P4
Spock	Bikini	Arc	
Bazinga	Pain aux raisins	Vert	Arène
Knock Knock Knock	Princesse	Capuche	District 12
?			



Correction

```
1 <table>
2   <thead>
3     <tr>
4       <th>P1</th>
5       <th>P2</th>
6       <th>P3</th>
7       <th>P4</th>
8     <tr>
9   </thead>
10  <tbody>
11    <tr>
12      <td>Spock</td>
13      <td>Bikini</td>
14      <td colspan="2">Arc</td>
15    </tr>
```

```
1      <tr>
2          <td>Bazinga</td>
3          <td>Pain aux raisins</td>
4          <td>Vert</td>
5          <td>Arène</td>
6      </tr>
7      <tr>
8          <td>Knock Knock Knock</td>
9          <td>Princesse</td>
10         <td>Capuche</td>
11         <td>District 12</td>
12     </tr>
13     <tr>
14         <td colspan="4">?</td>
15     </tr>
16 </tbody>
17 </table>
```




CSS, la suite

Règles pour les listes

Il est possible d'affecter un style différent aux listes

- **list-style-type** : permet de changer le type de la liste
- **list-style-image** : affecte une image à la place des puces (ou numéros)
- **list-style-position** : définit la position des puces (ou numéros) (valeurs : *inside*, *outside*)

Règles pour les listes

Les types les plus utilisés

- **disc** : un cercle plein (valeur par défaut)
- **circle** : un cercle vide
- **square** : un carré plein
- **decimal** : nombres décimaux (valeur par défaut avec ``)
- **lower-alpha** : lettres minuscules (commence à **a**)
- **upper-alpha** : lettres majuscules (commence à **A**)

Tableaux et bordures

Deux propriétés CSS pour les tableaux à retenir

- **border-collapse** : indique si les bordures des cellules sont séparées ou fusionnées (valeurs : *separate*, *collapse*)
- **border-spacing** : indique l'espacement entre les cellules, seulement applicable avec border-collapse à *separate*.

Tableaux et bordures

Bordures séparées et 10px d'espace

Titre 1	Titre 2	Titre 3
Texte	Texte	Texte
Texte	Texte	Texte
Texte	Texte	Texte

Tableaux et bordures

Bordures fusionnées

Titre 1	Titre 2	Titre 3
Texte	Texte	Texte
Texte	Texte	Texte
Texte	Texte	Texte

Les ids et classes

- Permettent de donner des règles à des éléments via des sélecteurs plus précis
- Un id est unique dans une page (affecté à un seul élément HTML)
- Une classe peut être affectée à plusieurs éléments HTML

Les ids et classes

```
< HTML />
1 <div id="pics_container">
2   ...
3 </div>
4 <span class="error">Une erreur est survenue !</span>
```

```
{ CSS }
1 #pics_container {
2   width: 80%;
3   background: #000000;
4 }
5 .error {
6   color: #FF0000;
7 }
```


Hériorarchie avec id et class

Il est possible d'être encore plus précis en combinant les ids/classes avec la hiérarchie des sélecteurs CSS

```
{css} 1 #container .error {  
2     color: #FF0000;  
3     font-weight: bold;  
4 }
```

S'il existe des éléments avec la classe **error** en dehors de l'élément avec l'id **container**, ces règles CSS ne leurs seront pas appliquées

Sélecteurs plus complexes

```
{css} 1 input[type="button"] { // Les éléments input avec l'attribut type à button
2     color: #FFFFFF;
3 }
4 span[class*="button"] { // Les éléments span avec l'attribut class
5     color: #FFFFFF;      // contenant le terme button
6 }
7 img[title^="Mon"] {      // Les éléments img avec l'attribut title
8     border-color: #FFFFFF; // commençant par Mon
9 }
10 a[href$=".html"] { // Les éléments a avec l'attribut href terminant par .html
11     text-decoration: none;
12 }
```

Pas trop peur ?

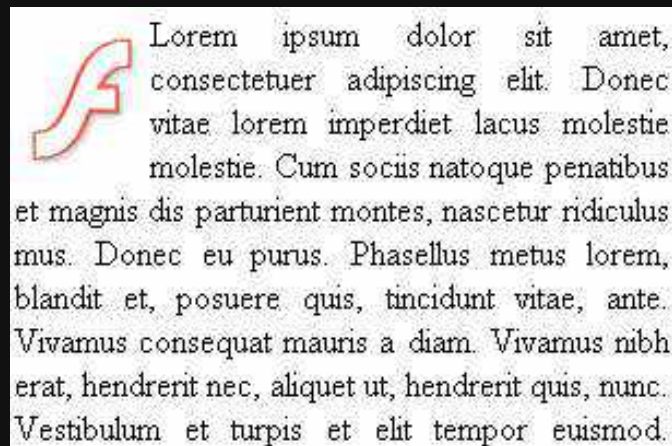
Float, ça flotte...

Comme son nom l'indique **float** permet de faire flotter du contenu, à droite ou à gauche

On placera du contenu autour de l'élément flottant

```
{css} 1 #mon_image {  
2     float: left; /* Ou right */  
3 }
```

Float, ça flotte...



Source: OpenClassrooms

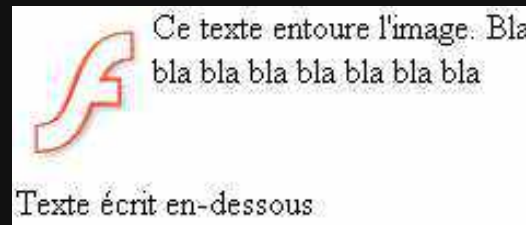
Clear, c'est clair ?

Pour arrêter un flottant on utilise **clear**

Il existe trois valeurs possibles : left | right | both

```
{css}
1 .clear_left {
2   clear: left;
3 }
```

Clear, c'est clair ?



Source: OpenClassrooms

Width et height

Permettent de définir la largeur et hauteur d'un élément (fonctionne seulement sur un élément de type *block* ou *inline-block*)

```
{css} 1 #container {  
2     width: 50%;  
3     height: auto;  
4 }
```

Unités : px, %, em, rem, in, cm,...

Min-width et min-height

Permettent de définir la largeur minimum et hauteur minimum d'un élément (fonctionne seulement sur un élément de type *block* ou *inline-block*)

```
{css} 1 #container {  
2     min-width: 150px;  
3     min-height: 150px;  
4 }
```


Max-width et max-height

Permettent de définir la largeur maximum et hauteur maximum d'un élément (fonctionne seulement sur un élément de type *block* ou *inline-block*)

```
{css} 1 #container {  
2     max-width: 90%;  
3     max-height: 500px;  
4 }
```

Bordures

- Ajoute une bordure à un élément
- Il existe plusieurs types de bordure (solid, dashed, dotted, double, groove, ridge, inset, outset)
- Epaisseur et couleur en option (par défaut medium (3px) de la couleur du texte)

Bordures

- **border-style** : définit le style de la bordure
- **border-width** : définit l'épaisseur de la bordure (px)
- **border-color** : couleur de la bordure (nom, #hexa, rgb, rgba)

Bordures

```
{css} 1 nav {  
2     border: 1px solid #FF0000;  
3 }
```

Bordures

```
{css} 1 img {  
2     border-top-style: dotted;  
3     border-right-style: solid;  
4     border-bottom-style: dotted;  
5     border-left-style: solid;  
6     border-color: #FFFFFF;  
7     border-width: 2px;  
8 }
```




Exercice, la suite

- Créer une division de 80% de large avec une bordure rouge de 2px en pointillés
- Mettre une image flottante à gauche
- Ajouter deux paragraphes, le deuxième doit s'afficher en dessous de l'image flottante
- Vous êtes libre d'utiliser des **ids** et/ou des **classes**

A 4x4 Rubik's cube is shown in the center of the image, slightly out of focus. The cube has various colored squares (yellow, green, blue, red, orange, white, and grey) and a small circular logo on one of its faces. A dark blue horizontal bar with rounded ends is positioned across the middle of the image, containing the word "Correction" in white text. The background is a dark blue gradient, and the cube is reflected on a dark surface below it.

Correction

Pour l'HTML

```
< HTML /> 1 <div id="container">
2   
3   <p>AgroSup Dijon - Enseignement à distance est...</p>
4   <p class="last-paragraph">Plus d'informations sur le site</p>
5 </div>
```

Pour le CSS

```
{css}
1 #container {
2     width: 80%;
3     border: 2px dotted red;
4 }
5 img {
6     float: left;
7 }
8 .last-paragraph {
9     clear: left;
10 }
```



AgroSup Dijon - Enseignement à distance
est...

Plus d'informations sur le site

A black dog, possibly a Labrador, is sitting on a wooden floor and looking directly up at the camera with a questioning expression. The dog's eyes are wide and brown. A semi-transparent dark grey horizontal bar is overlaid across the middle of the image, containing the text "Des questions ?" in white. The background shows a wooden floor, a white vent, and a person's leg in grey pants.

Des questions ?