

Harvard CS30W HTML and CSS

Franz Girardin

1 Mai 2023

2 | CHAPITRE 1 HTML

- 1.1 Listes HTML 3
- 1.2 Images HTML 3
- 1.3 Liens HTML 4
- 1.4 Tableau HTML 4
- 1.5 Formulaire HTML 5

6 | CHAPITRE 2 CSS

8 | CHAPITRE 3 Conception Réactive

- 3.1 Concept de réactivité 8
- 3.2 Media Queries 8
- 3.3 Flexbox 9
- 3.4 Grid 10
- 3.5 Bootstrap 11

14 | CHAPITRE 4 SASS

- 4.1 Création de SASS 14
- 4.2 Compilation de SCSS 15
- 4.3 Simplification de la sélection d'éléments 15
- 4.4 Héritage 16

Définition 1 HTML *Hypertext Markup Language*

Langage de balisage qu'on utilise pour décrire la structure d'un document destiné à afficher une page web.

Définition 2 Déclaration doctype

Formule qu'on utilise pour préciser la version de HTML utilisé dans la page qui suit : `<!DOCTYPE html>`

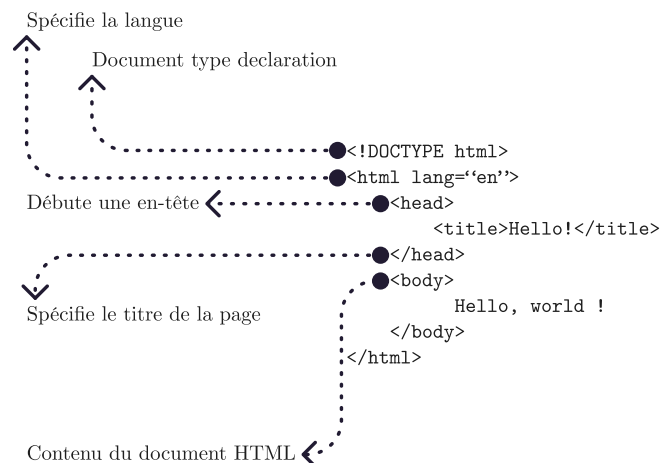
Remarque.

Lorsqu'on précise `html` après `!doctype` ce signale au navigateur qu'on utilise le standard HTML5 pour interpréter le contenu du document HTML.

Syntaxe. En-tête `<head>`

Cette section contient des informations d'ordre générale concernant la page. Notamment, on y trouve le titre de la page : `<title> </title>`

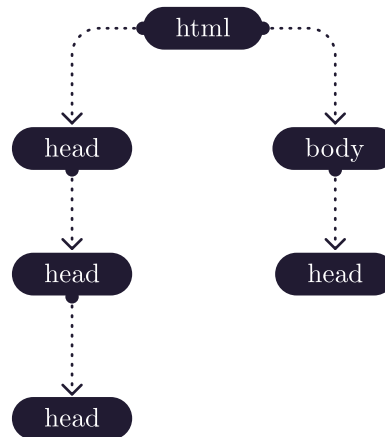
FIGURE 1.1 – Structure générale d'un document HTML



Définition 3 DOM *Document Object Model*

Structure qui permet de représenter les différents éléments d'une page web et la relation hiérarchique entre chacun

FIGURE 1.2 – DOM d'un document HTML



1.1 Listes HTML

Il est possible de créer des listes en faisant appel aux balises ` ` et ` ` qui créent des listes ordonnées et non ordonnées, respectivement. Chaque élément d'une liste est introduit par la balise ``.

Exemple.

```
1  <!DOCTYPE html>           <!-- Debut de la balise html contenant la page web entiere -->
2  <html lang="en">
3    <head>
4      <title>Hello!</title>
5    </head>
6    <body>
7      An ordered List:
8      <ol>
9        <li>First Item</li>
10       <li>Second Item</li>
11       <li>Third Item</li>
12     </ol>
13   </body>
14 </html>
15
```

1.2 Images HTML

Il est possible d'inclure une image grâce à la balise ``. L'attribut `src` prend comme argument le lien de l'image qu'on désire inclure. L'attribut `alt` prend comme argument un texte qui sera affiché

si le navigateur *n'arrive pas à afficher l'image*.

Exemple.

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>image</title>
5    </head>
6    <body>
7      <!-- Fournit le lien de l'image cat.jpg -->
8      
9    </body>
10 </html>
11
```

1.3 Liens HTML

Il est possible de partager des liens grâce à la balise ``. La balise contient un text qui s'affichera comme un lien cliquable.

Exemple.

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>lien</title>
5    </head>
6    <body>
7      <!-- Fournit un lien vers la page google.com -->
8      <a href="https://google.com">Click Here</a>
9    </body>
10 </html>
11
```

1.4 Tableau HTML

Il est possible de générer un tableau grâce à la balise `<table></table>`. Cette balise est un contenant pour la balise `<thead></thead>` qui permet de générer une *en-tête* pour le tableau. Chaque élément de l'entête est introduit par `<th></th>`. On peut ensuite introduire les éléments du tableau grâce au contenant `<tbody></tbody>`. Dans le corps du tableau, chaque rangée est introduite par `<tr></tr>`. On ajoute les éléments grâce à `<td></td>`.

Exemple.

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>Tableau</title>
5    </head>
6    <body>
7      <table>
8        <tr>
```

```

9         <thead>
10             <th>Ocean</th>
11             <th>Average Depth</th>
12             <th>Maximum Depth</th>
13         </thead>
14     </tr>
15
16     <tbody>
17         <tr>
18             <td>Pacific Ocean</td>
19             <td>4,280 m</td>
20             <td>10,911 m</td>
21         </tr>
22
23         <tr>
24             <td>Atlantic Ocean</td>
25             <td>3,4646 m</td>
26             <td>8,486 m</td>
27         </tr>
28
29     </tbody>
30
31 </table>
32 </body>
33 </html>
34

```

1.5 Formulaire HTML

Il est possible de créer un formulaire qui donne l'opportunité à l'utilisateur d'entrée des données. La balise `<form></form>` introduit ce formulaire. L'entrée de données par l'utilisateur est rendu possible par l'usage de `<input type="text">`. Le `placeholder` est un indice visuel qui permet l'utilisateur de comprendre la donnée qui lui est demandée.

Exemple.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Forms</title>
5  </head>
6  <body>
7      <form>
8          <input type="text" placeholder="First Name" name="first">
9          <input type="password" placeholder="Password" name="password">
10         <div>
11             Favorite Color:
12             <input name="color" type="radio" value="blue"> Blue
13             <input name="color" type="radio" value="green"> Green
14             <input name="color" type="radio" value="yellow"> Yellow
15             <input name="color" type="radio" value="red"> Red
16
17         </div>

```

```

18         <input type="submit">
19     </form>
20 </body>
21 </html>
22

```

CHAPITRE 2

CSS

Il est possible de modifier l'apparence ou *le style* des éléments d'un document HTML grâce au langage CSS. L'une des façons de spécifier un style est d'utiliser l'attribut `style` d'une balise. Chaque style est un couple de

clé: valeur;.

Exemple.

```

1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <title>Hello!</title>
5      </head>
6      <body>
7          <h1 style="color: blue; text-align: center;"> Welcome to my home page</h1>
8          Hello, world!
9      </body>
10 </html>
11

```

Il est également possible de modifier le style en utilisant des balises `<style></style>` dans une autre section du document

Exemple.

```

1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <title>Hello!</title>
5          <style>
6              h1 {color: blue;
7                  text-align: center;
8              }
9          </style>

```

```

10     </head>
11     <body>
12         <h1> Welcome to my home page</h1>
13         Hello, world!
14     </body>
15 </html>
16

```

La troisième alternative est d'utiliser une feuille de style CSS et d'en fournir le lien dans le document HTML

Exemple.

```

1  <!-- document1.html-->
2  <!DOCTYPE html>
3  <html lang="en">
4      <head>
5          <title>Hello!</title>
6          <link rel="stylesheet"
7              ↪ href="style.css">
8      </head>
9      <body>
10         <h1>Welcome to my home page</h1>
11         <h1>This is another heading</h1>
12         Hello, world!
13     </body>
14 </html>
15
16
17  <!-- document1.css-->
18  h1 {color: blue;
19      text-align: center;
20  }
21

```


3.1 Concept de réactivité

Définition Responsive Design

Un design réactif permet l’affichage optimale d’un page web, indépendamment de l’outil utilisé pour la consulter. Le design est réactif lorsqu’il s’adapte au *viewport*—la portion visible de la page.

Note :

Par défaut, les téléphones intelligents réduisent l’espace que prennent chaque éléments, pour présenter l’entière de la page telle qu’elle serait visible sur un écran d’ordinateur.

Syntaxe. Modification du *viewport*

La commande `<meta name="viewport" content="width=device-width, initial-scale=1.0">` permet d’ajuster les dimensions du *viewport* à la largeur de l’écran.

3.2 Media Queries

Ce sont des commandes qui permettent de contrôler l’apparence de la page en manipulant des paramètres de taille et de présentation. Dans l’exemple suivant, on modifie l’apparence de la page en fonction de la taille de l’écran de l’utilisateur.

Exemple. Changement de la couleur en fonction de la taille de l’écran

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Changement de couleur</title>
5
6
7      <style>
8          <!--Ce query change la couleur de fond si la largeur de l'ecran est >=
           ↳ 600px-->
9
10         @media (min-width: 600px) {
11             body {
12                 background-color: red;
```

```

13         }
14     }
15     <!-- Ce query change la couleur de fond si la largeur de l'ecran est <= 599
    ↪ px-->
16     @media (max-widthL 599px) {
17     body {
18         background-color: blue;
19     }
20     }
21     </style>
22 </head>
23 <body>
24     <h1>Welcome to my web page !</h1>
25 </body>
26 </html>

```

Remarque. Créer un *Media Query*

Chaque media query est introduit par `@media` suivit d'une expression entre parenthèses. Dans les parenthèses, on spécifie pour quel genre de media on veut appliquer le *query*. Ce qui suit l'expression entre parenthèse est une expression entre crochets qui spécifie le *query*.

3.3 Flexbox

Syntaxe. Flexbox

L'utilisation de *flexbox* peut être pertinente lorsqu'on désire présenter plusieurs éléments sur une page et que ces éléments pourraient sortir du champ de la page si on ne conçoit pas un design réactif.

Exemple.

```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>Hello!</title>
5      <style>
6        #container {
7          <!--indique qu'on veut affiche les elements a l'interieur
8            de container en mode flex-->
9            display: flex;
10
11
12            <!-- indique qu'on veut que les elements "wrap"; s'il n'y a
13              pas suffisamment d'espace
14              sur une ligne, placer les prochains elements dans le container
15              sur la ligne suivante.
16              flex-wrap: wrap;
17          }
18
19          <!-- selectionne tous les elements div qui sont enfant du container-->
20          #container > div {
21              background-color: springgreen;
22              font-size: 20px;

```

```

23         margin: 20px;
24         padding: 20px;
25         width: 200px;
26     }
27 </head>
28 <body>
29     <div id="container">
30         <div class="grid-item"></div>
31         <div>1.This is some sample text inside of div to demo Flexbox.</div>
32         <div>2.This is some sample text inside of div to demo Flexbox.</div>
33         <div>3.This is some sample text inside of div to demo Flexbox.</div>
34         <div>4.This is some sample text inside of div to demo Flexbox.</div>
35     </div>
36 </body>
37 </html>
38

```

Remarque. 1 Créer un *flexbox*

Les éléments qu'on désire contrôler par *flexbox* doivent être placés à l'extérieur d'un `<div> </div>`. On donne à ce div un id qu'on pourra sélectionner lorsqu'on modifiera son style.

3.4 Grid

Syntaxe. Grid

L'utilisation de grid est pertinente lorsqu'on veut organiser les éléments en grilles.

Exemple.

```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>Hello!</title>
5      <style>
6        #grid {
7          <!--indique qu'on veut affiche les elements a l'interieur
8            de container en mode grid-->
9          display: grid;
10
11
12          <!-- indique l'espace qu'on veut entre chaque element de la grille
13            grid-column-gap: 20px;
14            grid-row-gap: 10px;
15
16            grid-template-columns: 200px 200px auto;
17          }
18
19          <!-- selectionne tous les elements div qui faisant parti de la classe
20            ↪ "grid-item"-->
21          .grid-item {
22            background-color: white;
23            font-size: 20px

```

```

23         padding: 20px
24         text-align: center;
25
26     }
27
28 </head>
29 <body>
30     <div id="container">
31         <div class="grid-item">1. Un element</div>
32         <div class="grid-item">2. Un element</div>
33         <div class="grid-item">3. Un element</div>
34
35         <div class="grid-item">1. Un element</div>
36         <div class="grid-item">2. Un element</div>
37         <div class="grid-item">3. Un element</div>
38
39     </div>
40 </body>
41 </html>
42

```

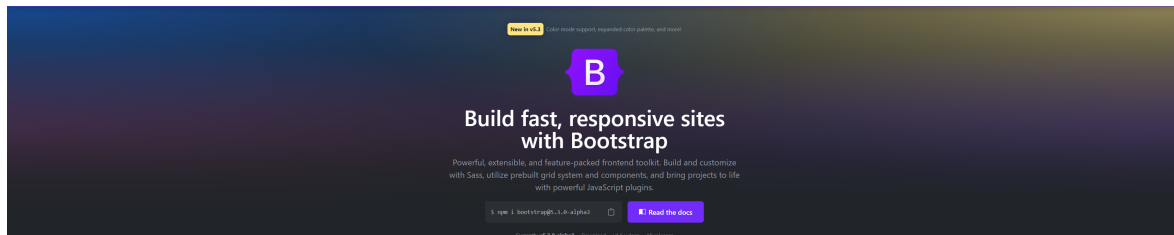
Remarque. 2 Créer un *grid*

Les éléments qu'on désire contrôler par *grid* sont placés dans un `<div> </div>`. On donne à ce `div` un `id` qu'on pourra sélectionner. La commande `grid-template-columns` permet de préciser la quantité de colonnes et la largeur de chacune de celles-ci. Lorsqu'on spécifie "auto", la colonne prend toute l'espace restant disponible.

3.5 Bootstrap

Note :

Bootstrap est une librairie qui contient des styles pour modifier l'apparence de certains éléments, simplement en référençant le CSS et sans avoir à définir le style soit-même.



Syntaxe. Référencer bootstrap

Pour utiliser bootstrap il faut, dans le document html concerné, fournir un lien référant à la librairie bootstrap : `<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4">`

Sur le site de bootstrap, on peut trouver une panoplie de composantes. Pour obtenir le style d'une composante de notre choix, il suffit d'attribuer à l'élément désiré la classe correspondante.

Définition Bootstrap Column Model

Système d'organisation des éléments utilisé par bootstrap pour s'assurer que la page web soit réactive. Bootstrap divise une page en sections où chaque rangée est divisée en colonnes de 12 unités.

Remarque. Fonctionnement de Bootstrap

One popular bootstrap feature is their grid system. Bootstrap automatically splits a page into 12 columns, and we can decide how many columns an element takes up by adding the class col-x where x is a number between 1 and 12. For example, in the following page, we have a row of columns of equal width, and then a row where the center column is larger :

Exemple.

```
1  <html lang="en">
2    <head>
3      <title>My Web Page!</title>
4      <link rel="stylesheet"
    ↪ href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
    ↪ integrity="sha384-Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    ↪ crossorigin="anonymous">
5      <style>
6        .row > div {
7          padding: 20px;
8          background-color: teal;
9          border: 2px solid black;
10       }
11     </style>
12   </head>
13   <body>
14     <div class="container">
15       <div class="row">
16         <div class="col-4">
17           This is a section.
18         </div>
19         <div class="col-4">
20           This is another section.
21         </div>
22         <div class="col-4">
23           This is a third section.
24         </div>
25       </div>
26     </div>
27     <br/>
28     <div class="container">
29       <div class="row">
30         <div class="col-3">
31           This is a section.
32         </div>
33         <div class="col-6">
34           This is another section.
35         </div>
36         <div class="col-3">
37           This is a third section.
38         </div>
39       </div>
40     </div>
41   </body>
42 </html>
```

Note :

Le système bootstrap gère l'apparence des éléments lorsque la fenêtre est modifiée. En modifiant la taille de la fenêtre, les colonnes se comportent de façon réactive ; la taille de chaque colonne sera ajustée pour qu'elles apparaissent toutes sur la même ligne.

Remarque. Taille de colonne en fonction de la taille de l'écran

To improve mobile-responsiveness, bootstrap also allows us to specify column sizes that differ depending on the screen size. In the following example, we use col-lg-3 to show that an element should take up 3 columns on a large screen, and col-sm-6 to show an element should take up 6 columns when the screen is small :

Exemple.

```
1      <!DOCTYPE html>
2      <html lang="en">
3      <head>
4          <title>My Web Page!</title>
5          <link rel="stylesheet"
6              ↪ href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
7              ↪ integrity="sha384-Vkoo8x4CGs03+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
8              ↪ crossorigin="anonymous">
9          <style>
10             .row > div {
11                 padding: 20px;
12                 background-color: teal;
13                 border: 2px solid black;
14             }
15         </style>
16     </head>
17     <body>
18         <div class="container">
19             <div class="row">
20                 <div class="col-lg-3 col-sm-6">
21                     This is a section.
22                 </div>
23                 <div class="col-lg-3 col-sm-6">
24                     This is another section.
25                 </div>
26                 <div class="col-lg-3 col-sm-6">
27                     This is a third section.
28                 </div>
29                 <div class="col-lg-3 col-sm-6">
30                     This is a fourth section.
31                 </div>
32             </div>
33         </div>
34     </body>
35 </html>
```

4.1 Création de SASS

Définition 4 SCSS

Il s'agit d'un langage qui est une extension de CSS. SCSS possède toutes les fonctionnalités de CSS et plus encore—notamment, la possibilité de créer des variables.

Syntaxe. Création d'un fichier SASS

Les fichiers SASS ont comme extension `.scss` au lieu de `css`. Chaque variable débute avec le symbole `$`

Exemple.

```
1  <!--Fichier HTML-->
2  <!DOCTYPE html>
3  <html lang="en">
4    <head>
5      <title>Formulaire</title>
6      <link rel="stylesheet" href="variable.scss">
7      <style>
8        .row > div {
9          padding: 20px;
10         background-color: teal;
11         border: 2px solid black;
12       }
13     </style>
14   </head>
15   <body>
16     <div class="container">
17       <div class="row">
18         <div class="col-3">This is a section</div>
19         <div class="col-3">This is another section</div>
20         <div class="col-3">This is a third section</div>
21       </div>
22     </div>
23   </body>
24 </html>
```

```

1  /* Fichier SASS */
2  $variable:red;
3  /* on assigne la couleur rouge a la virable "variable" */
4  ul {
5      font-size:14px;
6      color:variable; /* On utilise la variable "variable" pour donner une
    couleur rouge au texte contenu dans <ul> */
7  }
8  ul {
9      font-size:18px;
10     color:variable;
11 }
12

```

4.2 Compilation de SCSS

Le navigateur ne comprends pas le langage SCSS. Lorsqu'on écrit le code CSS, il faut le compiler en CSS. Dans le terminal on peut utiliser la commande `sass nomFichier.scss:nomFichierFin.css` où `nomFichier` est le nom du fichier scss et `nomFichierVer` est le nom du fichier final vers lequel on veut faire la conversion

Remarque. Recompilation du fichier SCSS

Après chaque modification du fichier SCSS, il faut le compiler à nouveau pour que les changements soient appliqués. Lorsqu'on travail à partir du terminal, on peut utiliser la commande `-watch nomFichier.scss:nomFichierVer.css` pour que la compilation se fasse automatiquement.

4.3 Simplification de la sélection d'éléments

Syntaxe. Sélecteur simplifié

Il est possible de simplifier la sélection d'élément structurant les éléments dans le fichier SCSS pour mettre en évidence la relation entre les éléments. Dans l'exemple suivant, seuls les éléments `p` et `ul` qui sont dans un `div` sont modifié.

Exemple.

```

1  /* Fichier SASS : organisation heriarchique facilitant la selection */
2  div {
3      font-size: 18px;
4
5      p {
6          color: blue;
7      }
8
9      ul {
10         color: green;
11     }
12 }
13
1

```



```

2  /* Fichier CSS correspondant : addition automatique des selecteur en s'inspirant
   de la hirerachie dictée par SASS */
3  div {
4      font-size: 18px;
5  }
6
7  div p {
8      color: blue;
9  }
10
11 div ul {
12     color: green;
13 }
14

```

4.4 Héritage

Il est possible de donner des propriétés à certains éléments et de faire en sorte que d'autres éléments héritent de ces propriétés. L'exemple suivant montre que tous les éléments `success` héritent des propriétés de la classe

`message`.

Exemple.

```

1  %message {
2      font-family:sans-serif;
3      font-size:18px;
4      font-weight:bold;
5      border:1px solid black;
6      padding:20px;
7      margin:20px;
8  }
9
10 success {
11     @extend %message; /* success herite des proprietes de message */
12     background-color: green;
13 }

```

Exemple. Résumé des principales balises

```

1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <title>HTML Elements</title>
5      </head>
6      <body>
7          <!-- We can create headings using h1 through h6 as tags. -->
8          <h1>A Large Heading</h1>
9          <h2>A Smaller Heading</h2>
10         <h6>The Smallest Heading</h6>
11
12         <!-- The strong and i tags give us bold and italics respectively. -->

```

```

13     A <strong>bold</strong> word and an <i>italicized</i> word!
14
15     <!-- We can link to another page (such as cs50's page) using a. -->
16     View the <a href="https://cs50.harvard.edu/">CS50 Website</a>!
17
18     <!-- We used ul for an unordered list and ol for an ordered one. both
    ↪ ordered and unordered lists contain li, or list items. -->
19     An unordered list:
20     <ul>
21         <li>foo</li>
22         <li>bar</li>
23         <li>baz</li>
24     </ul>
25     An ordered list:
26     <ol>
27         <li>foo</li>
28         <li>bar</li>
29         <li>baz</li>
30     </ol>
31
32     <!-- Images require a src attribute, which can be either the path to a file
    ↪ on your computer or the link to an image online. It also includes an alt
    ↪ attribute, which gives a description in case the image can't be loaded. -->
33     An image:
34     
35     <!-- We can also see above that for some elements that don't contain other
    ↪ ones, closing tags are not necessary. -->
36
37     <!-- Here, we use a br tag to add white space to the page. -->
38     <br/> <br/>
39
40     <!-- A few different tags are necessary to create a table. -->
41     <table>
42         <thead>
43             <th>Ocean</th>
44             <th>Average Depth</th>
45             <th>Maximum Depth</th>
46         </thead>
47         <tbody>
48             <tr>
49                 <td>Pacific</td>
50                 <td>4280 m</td>
51                 <td>10911 m</td>
52             </tr>
53             <tr>
54                 <td>Atlantic</td>
55                 <td>3646 m</td>
56                 <td>8486 m</td>
57             </tr>
58         </tbody>
59     </table>
60 </body>
61 <html>

```