Formation HTML 5, CSS 3

M2I - Formation - 2024

Donjon Audrey

m2iformation.fr

Module HTML – CSS Fondamentaux

Introduction

Points principaux qui seront abordés :

- HTML (HyperText Markup Language) : Structure ma page Web.
- CSS (Cascading StyleSheets) : Met en forme ma page Web.

Outils :

- VSC (Visual Studio Code) : Éditeur de texte
- Navigateurs : Firefox, Chrome, Edge…
- Recherche internet / Support de cours

Préparer l’environnement

1. Installer VSCode :
   https://code.visualstudio.com/Download

2. Installer Google chrome ou Mozilla firefox :
   https://www.google.com/intl/fr_fr/chrome/
   https://www.mozilla.org/fr/firefox/new/

3. Télécharger L’extension chrome ou Firefox Web Developer :
   web-developer Chrome
   web-developer Firefox
4. Installer les extensions vs code suivantes :
   a) French Language
   b) Trailing spaces
   c) Auto Rename tag

Le HTML

Introduction HTML

- 1969 : L'ancêtre d'Internet, Arpanet, est créé par l'armée américaine pendant la guerre froide. C'est le premier réseau décentralisé en forme de « toile », pensé pour éviter qu'une attaque nucléaire ciblée puisse le détruire.
- 1983 : Arpanet devient Internet.
- 1990 : Création du Web, du premier navigateur Web et du langage HTML (langage de balisage d'hypertexte : est le langage de balisage conçu pour représenter les pages web.) par Tim Berners-Lee, aussi appelé le « père du web ».
- 1995-1996 : HTML 2.0 => Amélioration balises, Tableaux, concept de formulaire etc.
- 1997 : HTML3.2 => Balises HTML améliorés, CSS dans le fichier HTML , formulaire amélioré etc.
- 1999 : HTML4.01 => fichier CSS externe etc.
- 2014 : HTML5 *(Version actuelle)* => Nouvelles balises HTML, nouveaux éléments de formulaire.

Les bases du HTML

- HTML est un langage de balise.
- Une balise permet d'indiquer la nature du texte.
- La syntaxe pour utiliser une balise est la suivante :

```
<balise_double></balise_double> ou <balise_orpheline/>
```

- La syntaxe pour déclarer une page html est divisée en 4 balises :

```
<!DOCTYPE html> : Doit se situer tout en haut de mon code avant les balises <html>
<html></html>   : Doit englober les balises suivantes
<head></head>   : Doit se situer entre les balises <html> et tout en haut
<body></body>   : Doit se situer entre les balises <html> et après les balises <head>
```

- L'ajout de commentaire en HTML se fait de la manière suivante :

```
<!-- Commentaire HTML -->
```

Les balises

- On structure son code à l'aide des balises, en voici quelques s'unes :

Paragraphe	`<p>Je suis un paragraphe</p>`

Titres (6 niveaux)	`<h1>Je suis un Titre niveau 1</h1>`
	`<h2>Je suis un Titre niveau 2</h2>`
	`<h3>Je suis un Titre niveau 3</h3>`
	`........`
	`<h6>Je suis un Titre niveau 6</h6>`

Saut de ligne	` `

Listes (ordonnées et non ordonnées)	`<ul>`
	`  <li>je suis la première ligne de la liste à puce</li>`
	`  <li>je suis la 2ème ligne de la liste à puce</li>`
	`  <li>je suis la 3ème ligne de la liste à puce</li>`
	`  <li>je suis la 4ème ligne de la liste à puce</li>`
	`</ul>`
	`<ol>`
	`  <li>je suis la première ligne de la liste numérotée</li>`
	`  <li>je suis la 2ème ligne de la liste numérotée</li>`
	`  <li>je suis la 3ème ligne de la liste numérotée</li>`
	`  <li>je suis la 4ème ligne de la liste numérotée</li>`
	`</ol>`

Les balises

- On structure son code à l'aide des balises, en voici quelques s'unes :

Image	`<img src="chemin_vers_mon_image.png" alt="texte_de_image">`
Lien (relatif ou absolu)	`<a href="url(exemple : https://www.google.com/)">lien absolu : construit avec un protocole et nom de domaine</a>` `<a href="chemin(exemple : ./page_contact.html)">lien relatif : construit par rapport à la page actuelle</a>`

Balises sémantiques de texte	`<em>Accentuation du texte</em>`
	`<strong>Texte important</strong>`
	`<mark>Texte pertinent dans son contexte</mark>`

Documentation en + : https://developer.mozilla.org/fr/docs/Web/HTML/Element

Les attributs

- Un attribut est une information additionnelle/spécification liée à une balise.

- Les attributs se déclarent dans les balises ouvrantes et sont cumulables.

- La syntaxe d'un attribut est la suivante :

```
> <balise_html attribut1="valeur_attribut" attribut2="valeur_attribut"> </balise_html>
```

- Quelques exemples :

```
> <img src="chemin_vers_mon_image.png" alt="texte_de_image">
> <a href="https://developer.mozilla.org/fr/docs/Web">Lien vers developer.mozilla.org</a>
```

Exercice 01

Créer une page (index.html) qui permettra d'afficher à l'écran les informations suivantes :

- Un titre d'onglet "Base clients"
- Un titre de niveau 1: "Ma base client"
- Un titre de niveau 2: "Description"
- Un texte : "Cette application permet de faire de la gestion de commande et de client."
- Les mots "commande" et "client" doivent ressortir en important
- Une petite image
- Deux titres de niveau 3: "Clients" et "Commandes"
- Une liste de client: "Jean Dupont", "Sarah Abdram" et "Rachel Taeck"
- Une liste de commande: "CMD01", "CMD02 ", "CMD03" et "CMD04"
- Un lien de redirection vers le site de Freepik

Bonus : Une icône dans l'onglet

Les balises sémantiques de structure

`<header></header>`	Représente l'en-tête de la page(Logo, navigation,etc.)
`<nav></nav>`	Menu de navigation
`<main></main>`	Contenu principal de la page
`<section></section>`	Sert à regrouper des contenus en fonction de leur thématique
`<article></article>`	La balise <article> sert à englober une portion généralement autonome de la page pouvant être repris dans d'autres sites(blog, magazine)
`<aside></aside>`	Informations complémentaires au document
`<footer></footer>`	Pied de page situé en bas du document(contact,mentions légales,etc.)

Exercice 02

Re structurez votre page index.html en utilisant des balises sémantique de structure au besoin.

Le CSS

Introduction au CSS

- Naissance du CSS pour accompagner le HTML fin 1996.
- Le CSS (Cascading Style Sheets : feuille de style en cascade) est un langage de mise en forme.
- HTML affiche et structure les données, CSS les met en forme (positionnement, couleurs, polices, etc...).
- Il est possible d'utiliser le CSS de trois manières différentes :
  - Dans un fichier externe avec l'extension .css avec un lien dans le `<head>` du fichier .html (méthode la plus courante):

```
> <link type="text/css" rel="stylesheet" href="styles.css">
```

  - Directement dans le code HTML (partie `<head>`) avec la balise `<style> </style>` (méthode à éviter autant que possible).
  - Directement dans le code HTML sur l'élément à styliser (méthode à éviter autant que possible). :

```
> <h1 color="red">Titre de niveau 1</h1>
```

La Syntaxe

```
selecteur{
    propriété : valeur ;
}
```

Ici, le sélecteur, c'est l'élément HTML ciblé, exemple :

```css
p{/*Sélecteur*/
    color:red;
    /*Propriété : valeur;*/
}
```

Quelques sélecteurs CSS

(sert à appliquer des propriétés CSS sur les éléments HTML que l'on sélectionne)

Sélecteur d'élément ou balise HTML
Ici : Touchera toutes les balises `<p></p>` de la page web
	CSS
	`p{`
	`  color: red;`
	`}`

Sélecteur de class
Ici : Touchera la/les balises ayant la `class` ="text-red"
	CSS
	`.text_red{`
	`  color: red;`
	`}`

Sélecteur d'ID
Ici : Touchera la balise ayant l'`id` = "color_main_title"
	CSS
	`#color_main_title{`
	`  color: brown;`
	`}`

Sélecteurs universels
Ici : Touchera tous les éléments HTML de la page
	CSS
	`*{`
	`  color: red;`
	`}`

Sélecteur d'attribut
Ici : Touchera toutes les balises `<a></a>` ayant comme attribut : `href`="https://google.com"
	CSS
	`a[href="https://google.com"]{`
	`  color: green;`
	`}`

Quelques Combinateurs

(sert à combiner plusieurs sélecteurs)

Sélection en liste Ici : Touchera toutes les balises `<p></p>` et la/les balises ayant la `class` ="text-red".	CSS `p, .text_red { color: red; }`
Sélection des enfants directs Ici : Touchera toutes les balises `<p></p>` qui sont enfants directs d'une balise `<div></div>`.	CSS `div > p{ color: green; }`
Sélection des enfants à n'importe quel niveau Ici : Touchera toutes les balises `<p></p>` qui sont descendant d'une balise `<div></div>`..	CSS `div p{ color: blue; }`
Selection du voisin direct Ici : Touchera toutes les balises `<p></p>` qui sont voisines d'une balise `<div></div>`..	CSS `div + p{ color: brown; }`
Autres...	https://www.w3.org/TR/selectors-3/#combinators

Principales propriété CSS

Propriété	Syntaxe	Exemples

Couleur
couleur de fond
	color
	background-color
		`color: red;` | `color: rgb(255, 99, 71);` | `color: #F10606;`
		`background-color: gray;` | `background-color: #808080;`
		`background-color: rgba(255, 99, 71, 0.80);`

Police	font-family	`font-family: Times;` | `font-family: Impact, Verdana, "Arial Black";`
Taille texte	font-size	`font-size: 15px;` | `font-size: small;` | `font-size: 1,5em;`
Styles de police	font-style font-weight ...	`font-style: italic;` | `font-weight: bold;` `text-decoration: underline;`
Alignement	text-align	`text-align: center;` | `text-align: right;` `text-align: justify;`
Marges	margin padding	`margin: 5px 5px 5px 5px;` | `margin-top: 10px;` | `margin-bottom: 30px;` | `padding: 10px;` | `padding-left: 15px;`

Principales propriété CSS

Propriété	Syntaxe	Exemples
Bordure	border	`border: 1px solid black;` `/*épaisseur + type de trait + couleur du trait*/`
Bordures arrondies	border-radius	`Border-radius: 10px;`
Soulignements et autres décorations	text-decoration	`text-decoration:underline;` ou `line-through;` ou `overline;` ou `none;`
Ombres block	box-shadow	`box-shadow: 6px 6px 0px black;` `/*Générateur d'ombre sur le block*/`
Hauteur, largeur	height, width	`height:400px;` | `width:300px;`

Image de fond	background-image	`/* Élément qui aura une image de fond */`
		`.element{`
		`  /* Hauteur pour être sûr de voir l'image de fond */`
		`  height: 200px;`
		`  background-image: url('../images/chat.jpg');`
		`}`

Principales pseudos classes

Syntaxe	Exemples

:hover	`/* La pseudo classe "hover" permet d'appliquer des styles uniquement si l'élément est survolé par la souris */`
	`p:hover{`
	`  color: red;`
	`}`

:active	`/* La pseudo classe "active" permet d'appliquer des styles uniquement si l'élément est en train d'être cliqué */`
	`p:active{`
	`  color: blue;`
	`}`

:visited	`/* La pseudo classe ":visited" permet de sélectionner les liens ayant déjà été visités */`
	`a:visited{`
	`  color: grey;`
	`}`

:link	`/* La pseudo classe ":link" permet de sélectionner les liens pas encore visités */`
	`a:link{`
	`  color: gold;`
	`}`

:first-child / :last-child / :nth-child(x)	`/* Sélectionne uniquement le premier enfant de l'élément parent (dans cet exemple le premier li de chaque ul) */`
	`ul > li:first-child{`
	`  color: pink;`
	`}`

:not	`/* Sélectionne tous les éléments ne correspondant pas au test (dans cet exemple, sélectionne tous les h2 n'ayant pas la classe "red") */`
	`h2:not(.red){`
	`  color: green;`
	`}`

Structure des éléments

Modèle de boîte : tout élément est inclus dans une boîte, aide à comprendre la mise en page CSS et le positionnement des éléments d'une page HTML.

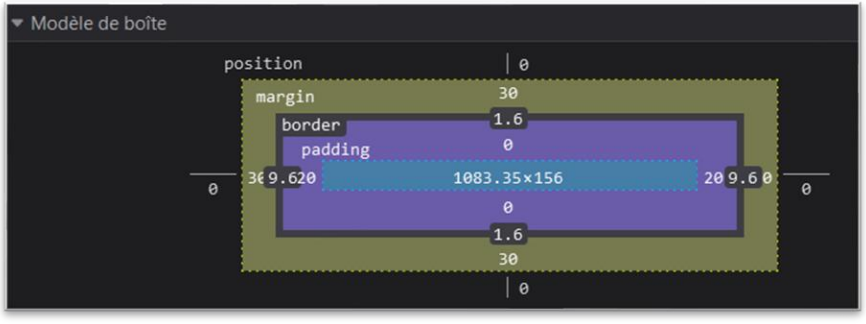
- Élément HTML de type Block
  - **Occupe toute la largeur** disponible de son parent
  - **L'élément suivant ou précedent** l'élément block se positionnera **au dessus ou en dessous de celui-ci** (retour à la ligne)
  - Width & Height modifiables

Exemple de Type block générique: `<h1>...<h6>` / `<p>` / `<div>`

- Élément HTML de type Inline
  - **Occupe la place de leurs contenu**
  - L'élément inline suivant se **positionne à la suite**
  - **Width & Height non modifiables**

  et pas de **padding et margin (top et bottom)**

Exemple de Type inline générique: `<a>` / `<span>`



- Élément HTML de type Inline-block
  - Se comporte **en partie comme un inline** :

    Occupe la place de leurs contenu / L'élément inline suivant se **positionne à la suite**
  - Se comporte **en partie comme un block** :

    Width & Height modifiables et padding et margin (top et bottom) possible

Exemple de Type inline-block générique: `<button>`

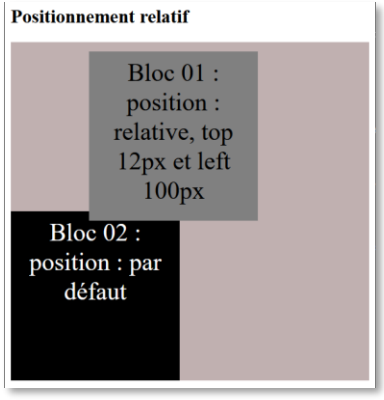
Flux et Positionnement

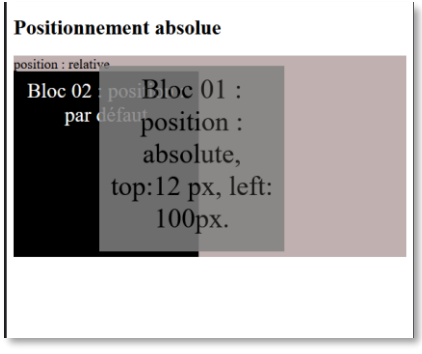
- Le flux HTML correspond à l'écoulement des informations (ou données) que le navigateur va interpréter et afficher. Un navigateur commence par le haut de la page, place les éléments HTML qu'il rencontre les unes à la suite des autres, de gauche à droite puis de haut en bas, à ceci près que cela dépend si ce sont des balises bloc ou en-ligne.

- La propriété CSS position permet de gérer le positionnement des éléments.
- On distingue 4 types principaux de positionnement :
  - (static): Valeur par défaut - Positionnement statique des éléments dans le flux
  - relative : Positionnement ajustée par rapport à la position initiale de l'élément dans le flux
  - absolute : Positionnement ajustée par rapport au parent relative le plus proche hors flux (chevauchement d'éléments alors possible)
  - fixed : semblable a absolute, mais en s'appuyant sur la fenêtre visible du navigateur et hors flux.

- Positionnement ajustée avec les propriétés top, left, bottom et right. Les valeurs négatives sont possibles.

- La propriété z-index permet de prioriser l'élément à afficher (en cas de chevauchement)





Les FlexBox

Flexbox est un **modèle de boîte flexible** permettant de gérer le positionnement des éléments dans un conteneur d'une manière plus fluide qu'avec les modèles de boîte classique.
Pour qu'un conteneur devienne un conteneur flex, il faut changer son type.

```css
div{
    display: flex;
}
```

Il y a 4 directions possibles pour distribuer les enfants d'un élément flex :

- row : horizontale, c'est la valeur par défaut
- column : verticale
- row-reverse : horizontale inversé
- column-reverse : verticale inversé

```Css
div{
    display: flex;
    /* Tous les éléments enfants seront disposés en ligne côte à côte */
    flex-direction: row;
}
```

Il est possible de **gérer l'alignement sur** l'axe principal avec la propriété "justify-content" et sur l'axe secondaire avec la propriété "align-items" :

L'axe principal et secondaire varies en fonction du **paramétrage** de la direction de votre flexbox :

Si "flex-direction" vaux "row" ou "row-reverse", alors l'axe principal sera l'axe horizontal et l'axe secondaire sera l'axe vertical.

Si "flex-direction" vaux "column" ou "column-reverse", alors l'axe principal sera l'axe vertical et l'axe secondaire sera l'axe horizontal.

```Css
div{
    display: flex;
    /* Tous les éléments enfants seront disposés en ligne côte à côte */
    flex-direction: row;
    /* Si il y a un manque de place, les éléments passeront à la ligne du dessous */
    flex-wrap: wrap;

    /* Les éléments seront centrés sur l'axe principal (sur l'axe horizontal dans cet exemple) */
    justify-content: center;

    /* Les éléments seront centrés sur l'axe secondaire (sur l'axe vertical dans cet exemple) */
    align-items: center;
}
```

Les **valeurs possibles** pour les alignements sont les suivantes :

flex-start : éléments positionnés au début de l'axe, c'est la valeur par défaut.

flex-end : éléments positionnés à la fin de l'axe.

center : éléments centrés au milieu de l'axe.

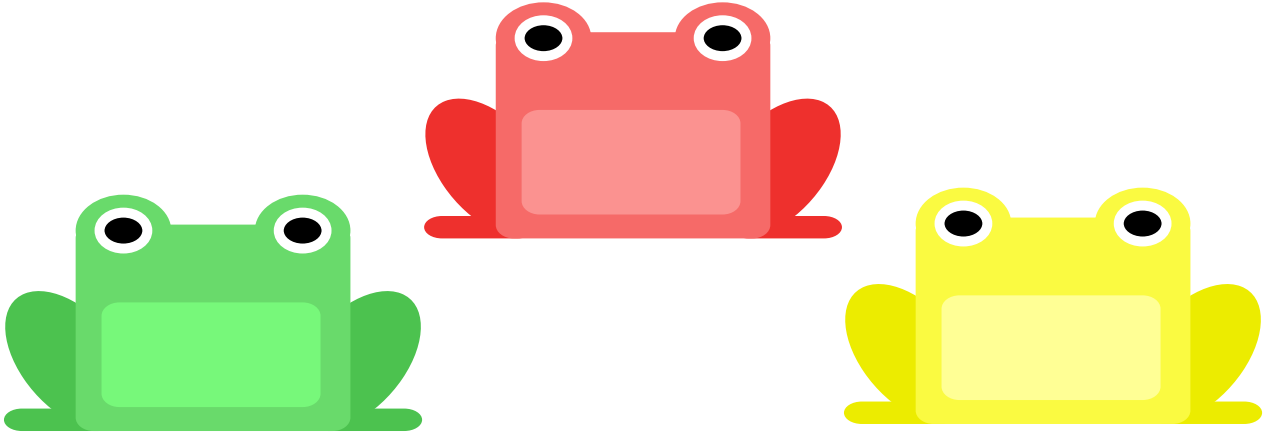
space-between : les éléments sont espacés à égale distance sur tout l'axe (et sont collés aux "bords" de l'axe).

space-around : les éléments sont espacés à égale distance sur tout l'axe (et ne sont pas collés aux "bords" de l'axe).

Entraînement Flexbox : Jeux Flexbox Froggy

Outil pour tester les propriétés css qui existent :
https://codepen.io/enxaneta/full/adLPwv/

Lien vers le jeu : https://flexboxfroggy.com/#fr



Fin du module

m2iformation.fr