Formation HTML / CSS

Objectif : réalisation d'un site statique en mobile first

Intro

Définition

Le HTML est un langage créé en 1991. Les sigles « HTML » sont l'abréviation de « HyperText Markup Language » ou « langage de balisage hypertexte » en français.

www : World Wide Web

Le World Wide Web Consortium (W3C) est une communauté internationale travaillant à l'élaboration de normes Web.

W3C régit les grands standards techniques du web.

Son rôle est majeur afin de définir les mêmes règles pour les développeurs web du monde entier.

Dirigé par Tim Berners-Lee, inventeur et directeur du Web, et Jeffrey Jaffe, PDG,

Le W3C a pour mission d'amener le Web à son plein potentiel et définit les standards web.

On **doit** valider notre code HTML sur **W3C validator**.

On va utiliser des balises ou tag en anglais
Ex : <title>Ceci est un titre</title>

Les éléments en HTML sont insensibles à la casse
Mon chat est <STRONG>très</STRONG> grincheux.
Mon chat est **très** grincheux.

Un site internet est un ensemble de fichiers de code et de medias (images, videos, etc.) liés entre eux et disponibles à tout moment via le web.

On les héberge généralement sur un serveur d'un hébergeur professionnel (OVH, Digital Ocean, Ikoula, Gandi)

Le serveur à pour rôle de servir aux clients (ordinateurs, smartphone, télévision).

Nous pouvons surfer sur Internet via des navigateurs (Firefox, Opera, Safari, Edge, Chrome, Tor, Brave…).

Ce sont les navigateurs qui comprennent et affichent les pages HTML.

Chaque navigateur possède sa propre feuille de styles (CSS)

Les quatre étapes de la création d'une page web sont :

- **Style** : le navigateur comprend la structure HTML du code qu'il reçoit et prépare le style qui sera appliqué,
- **Layout** (mise en page) : le navigateur détermine la mise en page et la taille des éléments en fonction du style qu'il a reçu,
- **Paint** : le navigateur transforme les éléments en pixels,
- **Composition :** le navigateur combine tous les éléments pour composer la page qui s'affiche

La structure minimale d'une page HTML

Pour qu'une page HTML soit déclarée valide, elle doit obligatoirement comporter certains éléments et suivre un schéma précis.
https://validator.w3.org/

Une page non valide ne sera pas comprise par le navigateur.
Une page non valide sera également mal analysée par les moteurs de recherche.

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Formation HTML et CSS</title>
</head>
<body>

</body>
</html>
```

Le doctype sert à indiquer le type du document, ici le HTML.

Le tag meta est une meta donnée, ici l'encodage de la page : UTF-8 (tous les alphabets latins).

L'élément head est un élément d'en-tête. Il contient des informations générales sur une page HTML qui ne sont pas affichées sur la page elle-même. Ces informations sont appelées métadonnées et comprennent des éléments tels que le titre du document HTML et des liens vers des feuilles de style.

L'élément `body` va lui contenir tous les éléments définissant les contenus « visibles ». C'est le corps de la page.

L'élément `title` va nous permettre d'indiquer le titre de la page dans le navigateur.

Il y a quelques règles à respecter lorsqu'on enregistre un fichier de code : le nom du fichier ne doit pas contenir d'espace ni de caractères spéciaux (pas de caractères accentués ou de ponctuation ni de sigles) et doit commencer par une lettre.

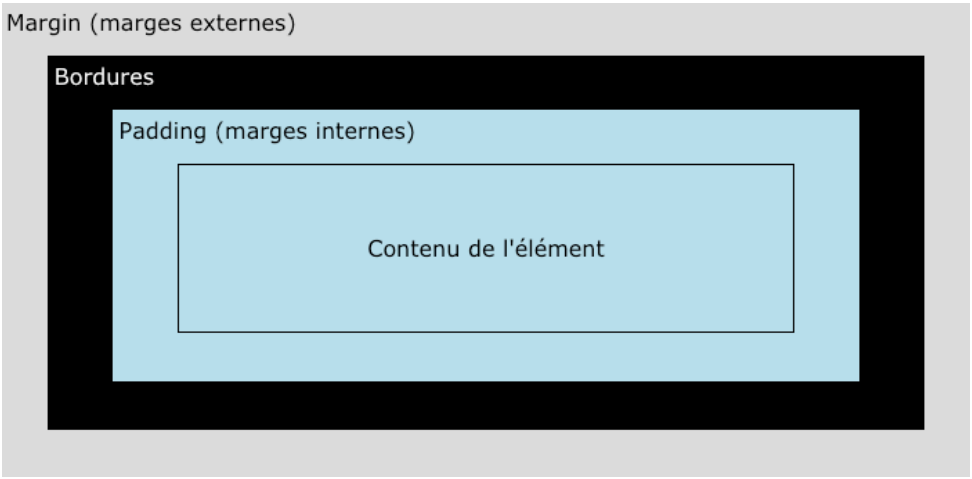
Les boites

L'élément HTML (ex : div, p, h1) contient d'autres boites rectangulaires

- Le contenu
- Les marges internes (padding)
- Les bordures (border)
- Les marges externes (margin)

Ces différentes boites vont se comporter de manière différente en fonction des CSS
Les CSS utilisent ces boîtes pour les agencer dans votre mise en page.

Tout élément HTML peut être représenté sous forme d'une boite rectangulaire



Éléments de type **block**

- Les éléments de type **block** forment **un bloc visible sur une page**
- Un élément de type `block` va toujours « aller à la ligne » (créer un saut de ligne avant et après le bloc)
- Un élément de type `block` va toujours prendre toute la largeur disponible au sein de son élément parent
- Les éléments de type block sont souvent des éléments structurels de la page (section, paragraphes, listes, navigation, footer, etc.

Un élément de niveau bloc ne peut pas être imbriqué dans un élément en ligne, mais il peut être imbriqué dans un autre élément de niveau bloc.

Tag **block**

<body>
** **
<p>
<h1> <h2> <h3> <h4> <h5> <h6>
<header>
<main>
<section>
<article>
<aside>
<nav>
<footer>
** <dl> <dt> <dd>**
<table>
<form> <fieldset>
<figure> <figcaption>
<canvas> <video> <audio>

Éléments de type **inline**

- Les éléments en **ligne** sont contenus **dans des éléments de type block**.
- Un élément de type `inline` ne va occuper que la largeur nécessaire à l'affichage de son contenu par défaut.
- Un élément en ligne ne fait pas apparaître une nouvelle ligne dans le document.
- Il apparaît généralement dans un paragraphe de texte, (hyperlien <a> ou des éléments de mise en évidence tels que ou).

Tag **inline**

** <a> <img> <input> <label> <textarea> <select> <button>**

En CSS

- display : block : affichage sous forme d'un bloc ;
- display : inline : affichage en ligne ;
- display : none : l'élément n'est pas affiché.

```
<div>
      div est de type block, un lien de type inline <a>type inline</a>
</div>
```

Imbrication des balises

<balise ouvrante élément A>

<balise ouvrante élément B>

</ balise fermante élément B>

<balise ouvrante élément C>

</ balise fermante élément C>

<balise orpheline élément D>

</ balise fermante élément A>

Le code HTML **doit** être indenté à chaque nouvelle imbrication.

Les commentaires !

```
<!-- Ceci est un commentaire HTML -->
```

Ils sont utiles dans 3 cas :

- Dans le cas d'un long projet, afin de bien se rappeler soi-même pourquoi nous avons écrit tel ou tel code, ou encore pour se repérer dans le code ;
- Si l'on souhaite distribuer son code, ou si l'on travaille à plusieurs, cela fait beaucoup plus professionnel et permet aux autres développeurs de comprendre beaucoup plus rapidement et facilement le code distribué ;
- Pour « neutraliser » certaines parties d'un code sans toutefois le supprimer. Il suffit en effet de placer toute la partie du code en question en commentaire afin que celui-ci soit ignoré par le navigateur.

Ne placez pas d'informations sensibles en commentaire !

Liste complète des balises HTML5
https://jaetheme.com/balises-html5/

Vérification de votre code HTML
https://validator.w3.org/

Vérification de votre code CSS
https://jigsaw.w3.org/css-validator/

Vérification de la compatibilité d'une propriété CSS ou HTML
https://caniuse.com/

Cours HTML

https://web-god.github.io/greta-project/example.html#anchor voir code source

HTML est composé d'éléments, de balises (tags)

Voici un élément de paragraphe :

- Une balise d'ouverture (** **)
- Le contenu ("Hello World!" Texte)
- Une balise de fermeture (** **)

Les balises orphelines

Les balises orphelines n'ont pas de balises de fermeture

**
** break line; *retour à la ligne*

<hr> horizontal rule

<img> insertion d'image

Les entités HTML

Une entité HTML est une suite de caractère qui est utilisée pour afficher un caractère réservé ou un caractère invisible (comme un espace) en HTML.

-   (« non breaking space ») un espace simple dit espace « insécable » ;
-   (« en space ») une espace double ;
-   (« em space ») une espace quadruple ;
-   (« thin space ») espace très fin (demi-espace).
- ★ une étoile pleine.
- ☆ une étoile vide.
- < plus petit que
- > plus grand que
- & & (esperluette)

La balise

Utilisez `strong` pour marquer un texte très important (SEO).

La balise ne doit pas être utilisée pour mettre le texte en gras

La balise

 emphase

Les moteurs de recherche vont accorder une importance moins grande aux textes dans les éléments `<em>` qu'à ceux dans `<strong>` mais une importance plus grande à ces textes qu'à des simples paragraphes.

L'élément mark

<mark></mark> pour marquer sémantiquement un terme pour l'utilisateur.

L'élément `mark` va être utilisé pour mettre en relief certains textes qui vont être pertinents dans un certain contexte.

L'élément pre

Le tag **`pre`** sert à préformater un texte.
Cela signifie que tout le contenu qui se trouve à l'intérieur de cet élément va conserver la mise en forme que nous allons lui donner lors du rendu fait par le navigateur.
(Voir dans codepen les balises)

Les balises sémantiques

Balises	fonctions
<section>	Définit une section à l'intérieur d'un document HTML5.
<nav>	Définit une section possédant des liens de navigation. En général, on peut trouver ici des renvois vers d'autres pages.
<article>	Caractérise une section en tant que contenu à part entière au sein d'un document HTML5.
<aside>	Définit une section en tant que complément par rapport à ce qui l'entoure.
<header>	Définit une section en tant qu'en-tête. En règle générale, vous y trouverez des logos, le titre du site Web ainsi que la navigation.
<footer>	Définit une section comme un pied de page. Il s'agit souvent de l'adresse, des mentions légales ou du Copyright.
<main>	Caractérise une section en tant que contenu principal d'une page Web. La balise <main> peut être utilisée qu'une fois par document HTML5.

La **balise sémantique** d'une section de page Web présente un grand avantage : elle facilite le travail des moteurs de recherche sur un document HTML5 et répond aux standard du w3c.

Ceux-ci privilégient les contenus dont la structure est claire et concise.

Par conséquent, les éditeurs de sites Internet qui facilitent l'interprétation d'un document HTML en utilisant des balises adaptées ont de grandes chances d'être **bien référencés** par les moteurs de recherche.

Utiliser les bons éléments HTML pour définir précisément les différents contenus de vos pages va permettre aux navigateurs (et aux moteurs de recherche) de comprendre de quoi et comment est composée votre page et donc de l'afficher et de la référencer au mieux.

On appelle cela le **SEO** (Search Engine Optimisation) naturel

L’intégration d’éléments multimédia

Balises	fonctions
<audio>	Caractérise un fichier audio. *3 formats : mp3, ogg et wav*
<video>	Caractérise des contenus vidéo accompagnés de leurs pistes audio. *2 formats : mp4, ogg*

Voir codepen header

Les attributs HTML

Certains attributs sont facultatifs et d’autres obligatoires.

Les attributs apportent des informations supplémentaires sur le comportement d’un élément.

Un attribut contient toujours une valeur

```
<a href="https://www.monsite.fr" title="Tout sur le shifting"></a>
```

href est un attribut obligatoire; **title** est un attribut facultatif

```
<img src="../img/smile.png" alt="sourire">
```

src est un attribut obligatoire; **alt** est un attribut facultatif

Les listes non ordonnées

Les listes vont nous permettre de lister plusieurs éléments en les groupant sous un dénominateur commun qu'est la liste en soi.

Les navigateurs et les moteurs de recherche vont donc comprendre qu'il y a une relation entre les différents éléments de liste.

Liste non ordonnée : **<ul>** Unordered List

- Premier élément de ma liste, **<li>** List Item
- Deuxième élément de ma liste, **<li>** List Item
- Troisième élément de ma liste. **<li>** List Item

Par défaut les éléments d'une liste non ordonnée sont précédés d'une puce.

Ce type de liste va nous servir à créer des menus de navigation par exemple ou encore regrouper des éléments ayant une relation commune.

Les listes ordonnées

Nous allons utiliser les listes ordonnées lorsqu'il y aura une notion d'ordre ou de progression logique ou encore de hiérarchie entre les éléments de notre liste.

Liste ordonnée : **<ol>** Ordered List

1. Verser la farine, **<li>** List Item
2. Ajouter les oeufs, **<li>** List Item
3. Verser lentement le lait, **<li>** List Item

Les listes ordonnées peuvent être des nombres ou des lettres en utilisant l'attribut **type**

A. Sommaire
B. Chapitre 1
C. Chapitre 2
D. Chapitre 3
E. Epilogue

```
<ol type="A">
  <li>Sommaire</li>
  <li>Chapitre 1</li>
  <li>Chapitre 2</li>
  <li>Chapitre 3</li>
  <li>Epilogue</li>
</ol>
```

Les listes de définitions

Les listes de définitions, encore appelées « listes de descriptions » vont nous permettre de lister des termes et d'ajouter des définitions ou descriptions pour chacun de ces termes.

Liste de description : **<dl>** Description List

HTML **<dt>** Description Term
HTML signifie HyperText Markup Language. **<dd>** Description

```
<dl>
  <dt>HTML</dt>
  <dd>HTML signifie HyperText Markup Language.</dd>
</dl>
```

Les listes imbriquées (voir codepen Listes imbriquées)

Avec les listes imbriquées, on comprend l'importance d'indenter son code

Les liens externes

Pour créer un lien on utilise la balise <a> pour anchor

Pour créer un lien vers une page d'un autre site en HTML (ou lien externe), il va falloir indiquer l'adresse complète de la page (c'est-à-dire son URL) en valeur de l'attribut `href` de notre lien.

```
<a href="https://www.wikipedia.org/" target="_blank">la home de Wikipedia</a>
```

`https://www.wikipedia.org/` est un chemin absolu

Lorsque l'attribut `href` prend une URL entière en valeur, on parle de valeur absolue (car celle-ci est fixe et ne dépend de rien).
Les liens externes utilisent toujours des valeurs absolues.

Pour ouvrir le lien dans un onglet différent, on utilise l'attribut **target**
Valeurs possibles de l'attribut target : _blank, _self, _parent, _top, *framename*

On parle de valeur absolue en opposition aux valeurs dites relatives.
Les valeurs relatives vont indiquer l'emplacement de la page cible par rapport à la page du lien.

Le navigateur place automatiquement des styles sur les liens

- le texte (notre ancre) est de couleur différente (bleu avant de cliquer puis violet une fois le lien visité) ;
- le texte est souligné ;
- le curseur de notre souris change de forme lorsqu'on passe sur le lien.

Les liens internes

Pour les liens internes on préconise fortement l'utilisation de valeurs relatives pour le chemin.

Un site web est un ensemble de fichiers et de ressources (pages de code de différents types et fichiers médias comme des images, des pdf, des vidéos…) liés entre eux.

Le site est hébergé sur un serveur dans un dossier, c'est le dossier racine.

Ce dossier aura des sous-dossiers ainsi que des fichiers.
Pour accéder à ces ressources, on utilise des chemins relatifs.

<img src="image.jpg">
Mon super script

L'attribut title permet la création de tooltips donnant des indications sur le lien

Les ancres

Un ancre est un lien permettant d'accéder à une autre partie d'une même page.

L'attribut `id` qui va nous servir à identifier l'endroit de la page où l'utilisateur doit être renvoyé. (Voir codepen les balises)

```
<p id=« sommaire »>Sommaire</p>
```

Le lien d'une ancre est précédé du symbole **#**

#id-arrivee

```
<a href=« #sommaire » title=« Aller au sommaire »>Sommaire</a>
```

Les ancres sont aujourd'hui très utilisées pour créer des liens d'évitement en accessibilité.

Exemple de navigation au clavier
https://www.accede-web.com/notices/

Envoyer un mail

On peut utiliser l'élément <a> avec **mailto** pour ouvrir automatiquement un mailer afin que l'utilisateur envoie un mail avec l'adresse préremplie.

```
<a href=« mailto:john@doe.com »>Contacter nous</a>
```

Voir codepen footer les balises

Télécharger un fichier

Pour télécharger un fichier un simple lien suffit :

```
<a href=« document.pdf » title=« Téléchargement du document »>
Téléchargement du document PDF</a>
```

On peut « forcer » le téléchargement d'un fichier en ajoutant un attribut **download** dans l'élément a tout en indiquant l'adresse du fichier en question en valeur de l'attribut `href`.

```
<a href=« /images/document.pdf" download>
  <img src=« /images/icon.png" alt=« Télécharger le document"
width="25" height="25">
</a>
```

Les navigateurs récents vont bloquer le téléchargement de certains fichiers dont les extensions auront été jugées comme « potentiellement dangereuses »

Les tableaux

Les tableaux en HTML vont nous permettre de présenter des données de manière organisée et sous une certaine forme pour les structurer et les rendre compréhensibles pour les navigateurs, moteurs de recherche et utilisateurs.

Attention ne pas utiliser les tableaux pour la mise en forme d'une page

Déclaration d'un tableau avec l'élément : <table>

Le tableau est constitué de lignes et de colonnes :
<tr> table row est une ligne
<td> table data est une cellule (colonne)

```
<table>
  <tr>
    <td>Nom</td>
    <td>Prénom</td>
    <td>Age</td>
    <td>Mail</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>28</td>
    <td>john.doe@mail.com</td>
  </tr>
  <tr>
    <td>Jane</td>
    <td>Monroe</td>
    <td>27</td>
    <td>jm@gmail.com</td>
  </tr>
</table>
```

Chaque rangée doit contenir le même nombre de cellule

Si une cellule ne contient pas de donnée, elle reste vide : <td></td>

Pour chaque partie d'un tableau, nous disposons d'un élément HTML spécifique :

- `thead` pour l'entête du tableau ;
- `tbody` pour le corps du tableau ;
- `tfoot` pour le pied du tableau.

Les attributs `colspan` et `rowspan` vont nous permettre de fusionner plusieurs cellules adjacentes d'une même ligne ou d'une même colonne.
Générateur de tableaux
https://divtable.com/table-styler/

Formulaire

Codepen formulaire

Du point de vue de l'expérience utilisateur, il est important de garder à l'esprit que plus vous demandez d'informations, plus vous risquez que votre utilisateur s'en aille.

Restez simple et ne perdez pas votre objectif de vue : ne demandez que ce dont vous avez absolument besoin.

La conception de formulaires est une phase importante de la construction d'un site internet ou d'une application.

https://www.smashingmagazine.com/2011/11/extensive-guide-web-form-usability/

Pour construire un formulaire, nous aurons besoin, des éléments HTML suivants :

`<form>` `<label>` `<input>` `<textarea>` `<button>`

La balise form

Elle sert à déclarer un formulaire.

Le formulaire va permettre d’envoyer des données au serveur qui les traitera.

Un formulaire HTML est composé d'un ou plusieurs champs (input).
Ceux-ci peuvent être :
- des zones de texte (sur une seule ligne ou plusieurs lignes),
- des boîtes à sélection,
- des boutons,
- des cases à cocher
- des boutons radio.

La plupart du temps, ces champs sont associés à un label qui décrit leur rôle — des étiquettes correctement implémentées sont susceptibles d'informer clairement l'utilisateur normal ou mal-voyant sur ce qu'il convient d'entrer dans le formulaire.

Attributs de la balise

```
<form action=« foo.php" method="post" enctype="multipart/form-data">
```

method => GET : les données sont envoyées au serveur via l’URL.

method => POST : les données sont ajoutées au corps de la requête HTTP comme variables

Pour voir les données envoyées dans la console
1. Pressez F12
2. Selectionnez « Réseau »
3. Selectionnez « Tout »
4. Selectionnez « foo.php » dans l'onglet « Nom »
5. Selectionnez « En-tête »

enctype : cet en-tête est très important, car il indique au serveur le type de données envoyées, ici un fichier.

Les champs

Les champs comportent des attributs, en voici quelques uns :

- ***type*** - établit le type de champ de texte.
  Par exemple: text, submit, number, email ou password.

- ***name*** - donne un nom au champ, il sera soumis en même temps que les données du formulaire.

- ***disabled -*** cet attribut est un booléen(true, false). Il indique que l'utilisateur ne peut pas avoir d'action sur cet élément.

- ***minlenght*** et ***maxlenght*** - la valeur minimale et maximale des caractères qui peuvent être tapés.

Validation HTML

- Les validations côté client se produisent dans le navigateur avant que les informations ne soient envoyées à un serveur.
- L'ajout de l'attribut **required** à un élément lié à la saisie permet de valider que le champ de saisie contient des informations.
- L'attribution d'une valeur à l'attribut **min** d'un élément de saisie numérique permet de valider une valeur minimale acceptable.
- L'attribution d'une valeur à l'attribut **max** d'un élément de saisie numérique permet de valider une valeur maximale acceptable.
- L'attribution d'une valeur à l'attribut **minlength** d'un élément de saisie de texte permet de valider un nombre minimum acceptable de caractères.
- L'attribution d'une valeur à l'attribut **maxlength** d'un élément de saisie de texte permet de valider un nombre maximal acceptable de caractères.

- L'affectation d'une regex au **pattern** fait correspondre l'entrée à la regex fournie.

- Si les validations sur un <form> ne passent pas, l'utilisateur reçoit un message expliquant pourquoi et le <form> ne peut pas être soumis.

https://my.deejo.com/en/15/black/none/2/none/tree?currency=EUR

https://obys.agency/

http://everylastdrop.co.uk/