

Initiation HTML5 - CSS3

Maxence Cuingnet

DevOps 2021



Structure globale doc HTML5

Ce n'est pas un langage de programmation. S'il y a une erreur, l'interpréteur le lancera quand même.

Un document HTML5 contient au minimum ces champs :

`<html>`, `<head>`, `<title>`, `<meta>`, `<link>`, `<script>` et `<body>`.

La très grande majorité de ces champs sont constitués de deux balises : ouvrante `<nomBalise>` et fermante `</nomBalise>`

Voici l'exemple de l'ossature d'une page HTML :

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Titre de la page</title>
  <link rel="stylesheet" href="style.css">
  <script src="script.js"></script>
</head>
<body>
  ...
  <!-- Le reste du contenu -->
  ...
</body>
</html>
```

`<html>` `</html>`

élément racine, il va englober les deux éléments `<head>` et `<body>`

`lang`

sert à l'aide de la synthèse vocale

`<head>` `</head>`

contient toutes les méta-informations, càd les données qui ne seront pas représentées directement à l'écran, mais qui sont liées au document

```
<meta charset="utf-8">
```

Il permet l'encodage des caractères, utf-8 est universel pour tous les types de caractères

```
<title> </title>
```

Très important, c'est le titre de l'onglet de la page

```
<link ...> <script ...>
```

Ce sont des éléments qui permettent de rattacher du style pour la page html (couleurs, mouvements...)

```
<body> </body>
```

Contiendra la page en elle-même, dont l'apparence sera influencée par les fichiers que nous avons appelé avec les balises

```
<link ...>
```



```
<script ...>
```

Quelques éléments de bases

Balises Titres <h>

```
<html>
  <head>
    <title>Les balises de titre</title>
  </head>
  <body>
    <h1>titre de
    <h2>titre de niveau 2</h2>
    <h3>titre de niveau 3</h3>
    <h4>titre de niveau 4</h4>
    <h5>titre de niveau 5</h5>
    <h6>titre de niveau 6</h6>
    texte normal
  </body>
</html>
```

Paragraphes et passage à la ligne :

Le texte est affiché tel qu'il est saisi dans le navigateur. Si nécessaire, vous pouvez effectuer un passage à la ligne avec la balise
 :

```
Texte sur la ligne 1 <br>
Texte sur la ligne 2
```

Vous pouvez aussi définir des paragraphes avec la balise <p> </p> :

```
<p>Ceci est un paragraphe</p> <p>Ceci est un autre paragraphe</p>
```

Commentaires :

Ctrl+ K+ C

pour les enlever : Ctrl + K + U

Pourquoi le CSS ?

Après avoir essayé rapidement une petite page html, nous voyons vite que le style est très limité.

Nous allons voir comment y insérer du style de trois manières différentes :

CSS signifie Cascading Style Sheets.

Inline CSS

```
<h1 style="color:blue; font-size:50px; font-style: italic;">Un  
titre en bleu</h1>
```

```
<p style="color:red;">Un paragraphe rouge</p>
```

```

```

mais cela encombre énormément la page html, alors on peut passer au Internal CSS

Internal CSS

On insère tout notre style dans les balises `<head>` `</head>`

```
<style>  
h1 {  
    color : blue;  
}  
</style>
```

Cependant, si les propriétés de style augmente, cela pose un second problème, c'est de devoir les ajouter à chaque page html du site et cela rallonge énormément le script de la page et l'alourdit. On utilise alors un dernier moyen qui regroupe tout le style.

External CSS

Créer un fichier style.css et copier le contenu de la balise style à l'intérieur

Maintenant, nous pouvons appeler tout le style dans un fichier CSS indépendant avec la balise `<link...>` entre les balises head

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

Présentation des tableaux et des listes

Il existe plusieurs moyens de mettre en forme la structure d'une page. Nous allons dans ce chapitre présenter deux fondamentaux qui sont les tables et les listes ordonnées et non ordonnées.

Après avoir vu leurs syntaxes nous ajouterons du CSS.

Listes :

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Titre de ma page</title>
  <link rel="stylesheet" href="./style.css">
</head>

<body>
  Ce texte apparaît dans le navigateur <br>

  <p>Voici un autre texte</p>

  <p>Voici un second texte</p>

  <!-- Balises pour les titres -->

  <h1>Ceci est un titre de niveau 1 avec la balise h1</h1>
  <h2>Ceci est un titre de niveau 2 avec la balise h2</h2>

  <!-- Liste ordonnée -->
  <ol class="tableau">
    <li>
      <h1>Premier élément</h1>
    </li>
    <li>Deuxième élément</li>
    <li>Troisième élément</li>
  </ol>
```



```
<!-- Liste non ordonnée -->
<ul>
  <li>Element 1 d'une liste non ordonnée</li>
  <li>Element 2 d'une liste non ordonnée</li>
  <li>Voici une liste ordonnée dans une liste non-ordonnée
    <ol>
      <li>
        Café
      </li>
      <li>Pain</li>
      <li>Beurre</li>
    </ol>
  </li>
</ul>

</body>

</html>
```

Tables:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML TABLES</title>
  <style>
    table {
      width: 70%;
      margin-left: auto;
      margin-right:auto;
      text-align: center;
      border-collapse: separate;
      border-spacing: 10px;
    }

    th,td {
      height: 70px;
      border: 10px solid rgb(135, 135, 218);
      padding: 10px;
    }

    th {
      text-decoration:underline;
    }

    tr:nth-child(even) {
      background-color: rgb(124, 182, 76);
      color: white;
      font-weight: 700;
    }
  </style>
</head>
<body>
  <h1>Premier tableau de base</h1>
  <table>
    <caption><h1>Voici un premier exemple de tableau</h1></caption>
    <tr>
      <th>Prénom</th>
      <th>Nom de Famille</th>
      <th>Points</th>
    </tr>
    <tr>
```

```

        <td>Jojo</td>
        <td>Robert</td>
        <td>50</td>
    </tr>
</table>
<hr>

<h1>Second Tableau</h1>
<table>
    <caption><h1>Voici un second exemple de tableau</h1></caption>
    <tr>
        <th>En-tête 1</th>
        <th>En-tête 2</th>
        <th colspan="2">En-tête 3</th>
        <th>En-tête 5</th>
    </tr>
    <tr>
        <td>
            <table>
                <caption>Ceci est un tableau dans une cellule</caption>
                <tr>
                    <td>1</td>
                    <td>2</td>
                    <td>3</td>
                </tr>
                <tr>
                    <td>1</td>
                    <td>2</td>
                    <td>3</td>
                </tr>
                <tr>
                    <td>1</td>
                    <td>2</td>
                    <td>3</td>
                </tr>
            </table>
        </td>
        <td rowspan="2">Cellule 2 </td>
        <td colspan="3">Cellule 3 </td>
    </tr>
    <tr>
        <td>Cellule 1 </td>
        <td>Cellule 3 </td>
        <td>Cellule 4 </td>
        <td>Cellule 5 </td>
    </tr>
</table>

```

```
</table>  
  
</body>  
</html>
```

2 Exercices :

- 1 - Mettre en place un menu de restaurant centré sur la page avec des listes ordonnées et non ordonnées
- 2 - Créer le tableau suivant

Recettes budgétaires par région			
	En millions d'Euros		2015 en €/hab.
	2014	2015	
Ile-de-france	399258	415920	178,8
Midi-pyrénées	73961	63826	219,4
Centre	43638	55802	185,9

```

<!DOCTYPE html>
<html>

<head> <title>Tableaux</title>
</head>
<body>
  <center>
    Recettes budgétaires par région
    <table border>

<th></th>
<th colspan = "2">En millions d'Euros</th>
<th rowspan = "2">2015 en €/hab.</th>
<tr>
<th></th>
<th>2014</th>
<th>2015</th>
</tr>
<tr align = "center">

<td>Ile-de-france</td>
<td>399258</td>
<td>415920</td>
<td>178,8</td>
</tr>

<tr align = "center"> <td>Midi-pyrénées</td>
<td>73961</td>
<td>63826</td>
<td>219,4</td>
</tr>

<tr align = "center">

```

```
<td>Centre</td>
<td>43638</td>
<td>55802</td>
<td>185,9</td>
</tr>

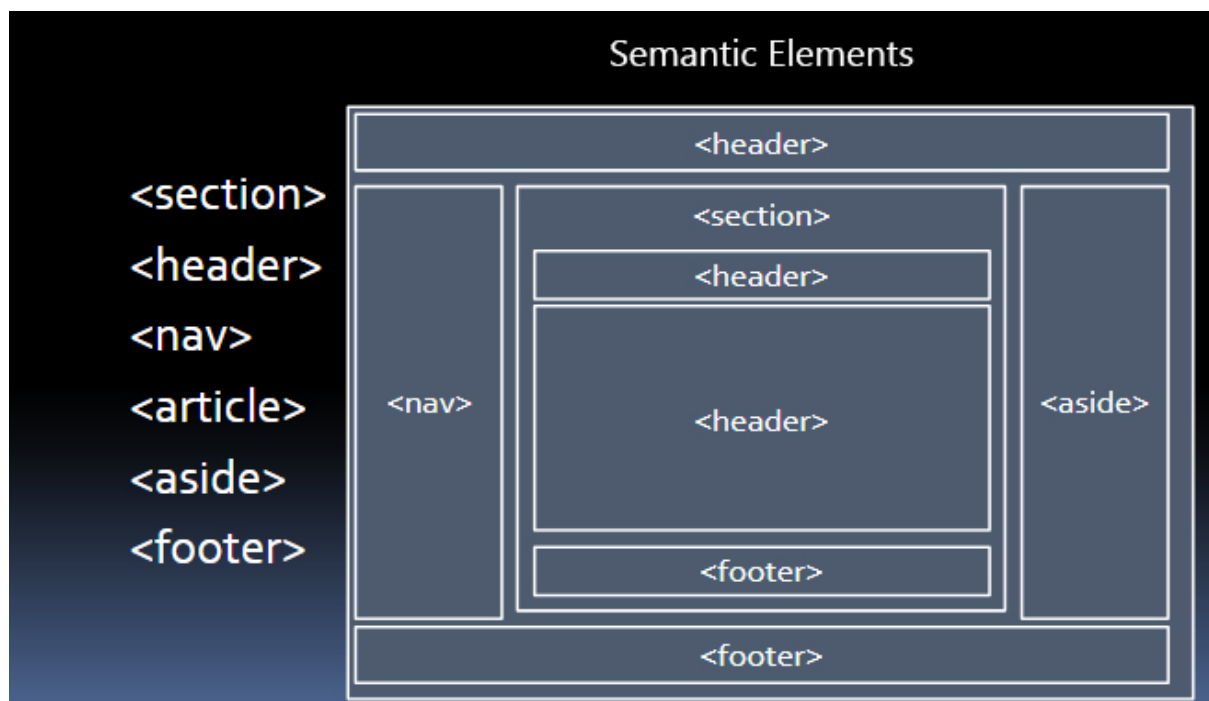
</table>
</body>
</html>
```

Éléments HTML sémantiques

Un élément sémantique décrit clairement sa signification tant pour le navigateur que pour le développeur.

Exemples d'éléments non sémantiques: `<div>` et `` - Ne dit rien sur son contenu.

Exemples d'éléments sémantiques: `<form>`, `<table>` et `<article>` - Définit clairement son contenu.



`<header>` `</header>`

Il existe une différence cruciale entre l'élément d'en-tête et l'utilisation généralement acceptée de l'en-tête (ou du masthead). Il n'y a généralement qu'un seul en-tête ou "bannière Masthead" dans une page. En HTML5, vous pouvez en avoir autant que vous le souhaitez. La spécification le définit comme «un groupe d'aides d'introduction ou de navigation». Vous pouvez utiliser un en-tête dans n'importe quelle section de votre site. En fait, vous devriez probablement utiliser un en-tête dans la plupart de vos sections. La spécification décrit l'élément de section comme "un regroupement thématique de contenu, généralement avec un en-tête".

```
<header>
  <h1>Gardening Times</h1>
  <h2>Getting started with flowers</h2>
</header>
```

```
<nav> </nav>
```

Destiné aux principales informations de navigation. Un groupe de liens regroupés n'est pas une raison suffisante pour utiliser l'élément nav. La navigation à l'échelle du site, en revanche, appartient à un élément de navigation.

```
<h1>The Blog of Examples</h1>
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#/events">Events</a></li>
    <li><a href="#/Blog">Blog</a></li>
    <li><a href="#/Products">Products</a></li>
  </ul>
</nav>
```

```
<section> </section>
```

Utilisé pour regrouper le contenu thématique. Sonne comme un élément div, mais ce n'est pas le cas. Le div n'a aucune signification sémantique. Avant de remplacer tous vos div par des éléments de section, posez-vous toujours la question "Tout le contenu est-il lié?"


```
<aside> </aside>
```

Utilisé pour le contenu lié de manière tangentielle. Le simple fait que certains contenus apparaissent à gauche ou à droite du contenu principal n'est pas une raison suffisante pour utiliser l'élément de côté. Demandez-vous si le contenu à côté peut être supprimé sans réduire la signification du contenu principal.

```
<footer> </footer>
```

On dirait que c'est une description de la position, mais ce n'est pas le cas. Les éléments de pied de page contiennent des informations sur l'élément contenant: qui l'a écrit, les droits d'auteur, les liens vers le contenu associé, etc. Alors que nous avons généralement un pied de page pour un document entier, HTML5 nous permet également d'avoir un pied de page dans les sections.

```
<footer>
<nav>
<p><a href="/blog.html">Blog</a> -
<a href="/archive.html">Archive</a> -
<a href="/index.html">Home</a></P>
</nav>
<p>Mes Coordonnées</p>
</footer>
```

Insertion Images et Vidéo

Insertion Image

```
<h2>Image HTML</h2>


```

L'attribut alt requis fournit un autre texte pour une image, si l'utilisateur ne peut pas l'afficher pour une raison quelconque (en raison d'une connexion lente ou d'une erreur dans l'attribut src).

On peut également mettre des formats gif dans le src et/ou un lien internet.

Et pour utiliser une image comme un lien :

```
<a href="lien de destination">
  
</a>
```

Insertion Vidéo

```
<video width="320" height="240" controls>  
  <source sr  
c="http://techslides.com/demos/sample-videos/small.ogv"  
type=video/ogg>  
  <source src="/build/videos/arcnet.io(7-sec).mp4"  
type=video/mp4>  
</video>
```

ou avec youtube :

Choisir une vidéo
Cliquer sur Share
Prendre le mode intégré
Insérer le code dans la page html

Créer des liens

Pour transformer un texte en lien :

```
<a href="url">link text</a>
```

Il est aussi possible de rajouter un attribut `target` dans la balise `<a>`:

il permet de spécifier où ouvrir le document :

- sur le même onglet : `target="_self"` (par défaut)
- ouvre un nouvel onglet : `target="_blank"`

Ou une image :

```
<a href="url">  
  
</a>
```

Exercice

This is a demo image

FIFA World Cup 2014 News

The biggest scoreline in the history of the **FIFA World Cup** qualifiers - and indeed in the history of international football - was recorded on 11 April 2001, when Australia beat American Samoa 31-0.

This legendary match also brought global renown for *Archie Thompson*, whose 13-goal haul set a new world record, which stands to this day, for an individual player in a single international match.

And though the defeat earned American Samoa ignominy, so inspiring has their subsequent recovery been that is now the subject of an acclaimed documentary, *'Next Goal Wins'*, showing across the world.

Destinations

- [BELO HORIZONTE, STADIUM : Estadio Mineirao](#)
- [BRASILIA, STADIUM : Estadio Nacional](#)
- [CUIABA, STADIUM : Arena Pantanal](#)
- [CURITIBA, STADIUM : Arena da Baixada](#)
- [FORTALEZA, STADIUM : Estadio Castelao](#)
- [MANAUS, STADIUM : Arena Amazonia](#)
- [NATAL, STADIUM : Estadio das Dunas](#)
- [PORTO ALEGRE, STADIUM : Estadio Beira-Rio](#)
- [RECIFE, STADIUM : Arena Pernambuco](#)
- [RIO DE JANEIRO, STADIUM : Maracanã - Estádio Jornalista Mário Filho](#)
- [SALVADOR, STADIUM : Arena Fonte Nova](#)
- [SAO PAULO, STADIUM : Arena de Sao Paulo](#)



Des liens dans la même page :

Pour chaque lien, vous procéderez en deux étapes :

Définition d'un signet pour marquer la partie à accéder.

Définition d'un lien hypertexte vers ce signet.

Pour placer un signet dans le document courant, utiliser une balise `<a>` :

```
<a name="Signet"> Texte</a>
```

Pour créer un lien hypertexte vers le signet que vous venez de définir, vous utiliserez une autre variante de la balise `<a>` :

```
<a href = "#Signet"> Texte </a>
```

où Signet est le nom du marqueur à pointer et Texte le texte qui servira de lien.

Exercice

Définissez un document qui contient plusieurs paragraphes de texte. Ce texte pourra être pris sur le site <http://fr.lipsum.com/>.

Placez ce texte dans des paragraphes. Définissez des signets pour chacun d'entre eux et des liens hypertextes pour accéder aux différents paragraphes.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=UTF-8" /> <title>Liens et signets</title>
</head>
<body>
<a href="#par1">Paragraphe 1</a> <a href="#par2">Paragraphe 2</a> <a
href="#par3">Paragraphe 3</a> <a href="#par4">Paragraphe 4</a> <a
href="#par5">Paragraphe 5</a>
<a name="par1"><p>Paragraphe 1 : Lorem ipsum dolor sit amet, consectetur
adipiscing elit...</p>
<a name="par2"><p>Paragraphe 2 : Lorem ipsum dolor sit amet, consectetur
adipiscing elit...</p>
<a name="par3"><p>Paragraphe 3 : Lorem ipsum dolor sit amet, consectetur
adipiscing elit...</p>
<a name="par4"><p>Paragraphe 4 : Lorem ipsum dolor sit amet, consectetur
adipiscing elit...</p>
<a name="par5"><p>Paragraphe 5 : Lorem ipsum dolor sit amet, consectetur
adipiscing elit...</p>

</body>
</html>
```

Lien vers un document distant :

En utilisant une syntaxe légèrement différente, il est possible d'accéder à un document d'un autre site :

```
<a href = "http://adresse_url"> texte </a>
```

C'est le même principe pour un document local en fournissant une adresse relative ou absolue

Formulaires HTML

Un formulaire HTML est utilisé pour collecter les entrées de l'utilisateur. L'entrée utilisateur est le plus souvent envoyée à un serveur pour traitement.

L'élément HTML `<form>` est utilisé pour créer un formulaire HTML pour la saisie utilisateur:

L'élément `<form>` est un conteneur pour différents types d'éléments d'entrée, tels que: champs de texte, cases à cocher, boutons radio, boutons d'envoi, etc.

L'élément `<input>`

L'élément HTML `<input>` est l'élément de formulaire le plus utilisé.

Un élément `<input>` peut être affiché de plusieurs façons, selon le type attribut.

Voici quelques exemples:

Type	Description
<code><input type = "text"></code>	Affiche un champ de saisie de texte sur une seule ligne
<code><input type = "radio"></code>	Affiche un bouton radio (pour sélectionner l'un des nombreux choix)
<code><input type = "checkbox"></code>	Affiche une case à cocher (pour sélectionner zéro ou plusieurs choix parmi tant d'autres)
<code><input type = "submit"></code>	Affiche un bouton d'envoi (pour soumettre le formulaire)
<code><input type = "button"></code>	Affiche un bouton cliquable

Type text

Le `<input type="text">` définit un champ de saisie sur une seule ligne pour la saisie de texte

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Attention : la largeur du champ par défaut est de 20 caractères

L'élément `<label>`

Notez l'utilisation du `<label>` élément dans l'exemple ci-dessus.

La balise `<label>` définit une étiquette pour de nombreux éléments de formulaire.

Le `<label>` est utile pour les utilisateurs de lecteurs d'écran, car le lecteur d'écran lira à haute voix l'étiquette lorsque l'utilisateur se concentrera sur l'élément d'entrée.

L'élément `<label>` aide également les utilisateurs qui ont des difficultés à cliquer sur de très petites régions (telles que des boutons radio ou des cases à cocher) - car lorsque l'utilisateur clique sur le texte dans l'élément `<label>`, il active le bouton radio / la case à cocher.

L'attribut `for` de la balise `<label>` doit être égal à l'attribut `id` du `<input>` élément pour les lier.

Boutons Radios

Le `<input type="radio">` définit un bouton radio.

Les boutons radio permettent à un utilisateur de sélectionner UN parmi un inputnombre limité de choix.

```
<!DOCTYPE html>
<html>
<body>

<h2>Radio Buttons</h2>

<form>
  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Male</label><br>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Female</label><br>
  <input type="radio" id="other" name="gender" value="other">
  <label for="other">Other</label>
</form>

</body>
</html>
```

Cases à cocher

Le `<input type="checkbox">` définit une case à cocher .

Les cases à cocher permettent à un utilisateur de sélectionner ZERO ou PLUS d'options sur un nombre limité de choix.

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

Bouton Submit

Le `<input type="submit">` définit un bouton pour soumettre les données du formulaire à un gestionnaire de formulaire.

Le gestionnaire de formulaire est généralement un fichier sur le serveur avec un script pour le traitement des données d'entrée.

La destination des informations est spécifiée dans le formulaire action attribut.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

Remarque :

Chaque champ de saisie doit avoir un attribut name à soumettre.

Si l'attribut name n'est pas attribué dans l'input, la valeur du champ saisie ne sera pas du tout envoyée.

L'attribut action

L'attribut action définit l'action à effectuer lorsque le formulaire est soumis.

Habituellement, les données du formulaire sont envoyées à un fichier sur le serveur lorsque l'utilisateur clique sur le bouton d'envoi.

Dans l'exemple ci-dessous, les données du formulaire sont envoyées dans un fichier appelé "action_page.php". Ce fichier contient un script côté serveur qui gère les données du formulaire:

```
<!DOCTYPE html>
<html>
```

```
<body>

<h2>HTML Forms</h2>

<form action="/action_page.html">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>

<p>En cliquant sur le bouton submit les informations du formulaire
seront transmises à la page appelée action_page</p>

</body>
</html>
```

Si l'attribut action est omis, l'action est définie sur la page actuelle.

L'attribut target

L'attribut target spécifie où afficher la réponse reçue après l'envoi du formulaire. Il peut avoir les valeurs suivantes :

Value	Description
_blank	La réponse est affichée dans une autre page
_self	La réponse est affichée dans la page courante
_parent	La réponse est affichée dans le cadre parent
_top	La réponse est affichée dans le corps de la page

La valeur par défaut est `_self` ce qui signifie que la réponse s'ouvrira dans la fenêtre courante

```
<form action="/action_page.php" target="_blank">
```

L'attribut de méthode

L'attribut `method` spécifie la méthode HTTP à utiliser lors de la soumission des données du formulaire.

Les données de formulaire peuvent être envoyées sous forme de variables URL (avec `method="get"`) ou en tant que transaction de publication HTTP (avec `method="post"`).

La méthode HTTP par défaut lors de la soumission de données de formulaire est GET.

```
<form action="/action_page.php" method="get">  
<form action="/action_page.php" method="post">
```

Remarquez la différence des adresses URL après avoir cliqué sur le bouton submit.

Remarques sur GET:

- Ajoute les données du formulaire à l'URL, par paires nom / valeur
- N'utilisez JAMAIS GET pour envoyer des données sensibles! (les données du formulaire soumis sont visibles dans l'URL!)
- La longueur d'une URL est limitée (2048 caractères)
- Utile pour les soumissions de formulaire où un utilisateur souhaite ajouter le résultat à ses favoris
- GET convient aux données non sécurisées, comme les chaînes de requête dans Google

Remarques sur POST:

- Ajoute les données du formulaire dans le corps de la requête HTTP (les données du formulaire soumises ne sont pas affichées dans l'URL)
- POST n'a pas de limites de taille et peut être utilisé pour envoyer de grandes quantités de données.
- Les soumissions de formulaire avec POST ne peuvent pas être ajoutées aux favoris

Astuce: utilisez toujours POST si les données du formulaire contiennent des informations sensibles ou personnelles!

L'attribut de saisie semi-automatique

L'attribut `autocomplete` spécifie si un formulaire doit avoir la saisie semi-automatique activée ou désactivée.

Lorsque la saisie semi-automatique est activée, le navigateur complète automatiquement les valeurs en fonction des valeurs que l'utilisateur a saisies auparavant.

```
<form action="/action_page.php" autocomplete="on">
```

L'attribut Novalidate

L'attribut `novalidate` est un attribut booléen.

Lorsqu'il est présent, il spécifie que les données de formulaire (entrée) ne doivent pas être vérifiées lors de la soumission.

Éléments de formulaire HTML

Ce chapitre décrit tous les différents éléments du formulaire HTML

La balise HTML `<form>` peut contenir un ou plusieurs des éléments de formulaire suivants:

- `<input>`
- `<label>`
- `<select>`
- `<textarea>`
- `<button>`
- `<fieldset>`
- `<legend>`
- `<datalist>`
- `<output>`
- `<option>`
- `<optgroup>`

L'élément `<input>`

L'un des éléments de formulaire les plus utilisés est l'élément `<input>`.

L'élément `<input>` peut être affiché de plusieurs manières, selon le type d'attribut. Nous les verrons après.

Nous avons déjà vu l'élément `label`.

L'élément `<select>`

L'élément `<select>` définit une liste déroulante:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```


L'élément `<option>` définit une option qui peut être sélectionnée.

Par défaut, le premier élément de la liste déroulante est sélectionné.

Pour définir une option présélectionnée, ajoutez le `selected` attribut à l'option:

```
<option value="fiat" selected>Fiat</option>
```

Valeurs visibles:

Utilisez le `size` attribut pour spécifier le nombre de valeurs visibles:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Autoriser les sélections multiples:

Utilisez le `multiple` attribut pour permettre à l'utilisateur de sélectionner plus d'une valeur:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Utiliser la touche Ctrl

```
<form action="/action_page.php">
  <label for="cars">Choose a car:</label>
  <select name="cars" id="cars">
    <optgroup label="Swedish Cars">
      <option value="volvo">Volvo</option>      <option
```

```
value="saab">Saab</option>
</optgroup>
<optgroup label="German Cars">
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</optgroup>
</select>
<br><br>
<input type="submit" value="Submit">
</form>
```

L'élément <textarea>

L'élément <textarea> définit un champ de saisie multiligne (une zone de texte):

```
<textarea name="message" rows="10" cols="30">
Le chat est en train de jouer.
</textarea>
```

Vous pouvez également définir la taille de la zone de texte en utilisant du CSS:

```
<textarea name="message" style="width:200px; height:600px;">
Le chat est en train de jouer.
</textarea>
```

L'élément <button>

L'élément <button> définit un bouton cliquable:

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```

Les éléments <fieldset> et <legend>

L'élément <fieldset> est utilisé pour regrouper les données associées dans un formulaire.

L'élément <legend> définit une légende pour l'élément <fieldset>.

```
<form action="/action_page.php">
  <fieldset>
    <legend>Informations</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

L'élément <datalist>

L'élément <datalist> spécifie une liste d'options prédéfinies pour un élément <input>.

Les utilisateurs verront une liste déroulante des options prédéfinies lors de la saisie des données.

L'attribut list du <input> élément, doit faire référence à l'attribut id de l'élément <datalist>.

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

Elément <meter>

```
<label for="disk_c">Disk usage C:</label>
<meter id="disk_c" value="2" min="0" max="10">2 out of 10</meter><br>

<label for="disk_d">Disk usage D:</label>
<meter id="disk_d" value="0.6">60%</meter>
```

L'élément <output>

L'élément <output> représente le résultat d'un calcul (comme celui effectué par un script).

```
<form action="/action_page.php"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>
```

Types d'input

Il existe en HTML 22 types d'input différents. Nous allons voir les principaux, nous avons déjà vu submit et radio par exemple

Input de type texte

<input type="text"> définit un **champ de saisie de texte sur seule ligne** :

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Input de type password

```
<form>
  <label for="username">Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd">
</form>
```

Input de type reset

`<input type="reset">`définit un **bouton de réinitialisation** qui réinitialisera toutes les valeurs du formulaire à leurs valeurs par défaut:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
```

```
<input type="text" id="lname" name="lname" value="Doe"><br><br>
<input type="submit" value="Submit">
<input type="reset">
</form>
```

Input de type checkbox

`<input type="checkbox">` définit une case à cocher .

Les cases à cocher permettent à un utilisateur de sélectionner ZERO ou PLUS d'options d'un nombre limité de choix.

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

Input de type date

Le `<input type="date">` est utilisé pour les champs de saisie qui doivent contenir une date.

En fonction de la prise en charge du navigateur, un sélecteur de date peut apparaître dans le champ de saisie.

```
<form>
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
</form>
```

Vous pouvez également utiliser le min et max attributs pour ajouter des restrictions aux dates:

```
<form>
```

```
<label for="datemax">Enter a date before 1980-01-01:</label>
<input type="date" id="datemax" name="datemax"
max="1979-12-31"><br><br>
<label for="datemin">Enter a date after 2000-01-01:</label>
<input type="date" id="datemin" name="datemin" min="2000-01-02">
</form>
```

Input de type email

Le `<input type="email">` est utilisé pour les champs de saisie qui doivent contenir une adresse e-mail.

En fonction de la prise en charge du navigateur, l'adresse e-mail peut être automatiquement validée lors de la soumission.

Certains smartphones reconnaissent le type d'e-mail et ajoutent «.com» au clavier pour correspondre à l'entrée d'e-mail.

```
<form>
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email">
</form>
```

Type Input Téléphone

```
<form>
  <label for="phone">Enter your phone number:</label>
  <input type="tel" id="phone" name="phone"
pattern="[0-9]{2}-[0-9]{2}-[0-9]{2}-[0-9]{2}-[0-9]{2}">
</form>
```

Input de type Nombres

Le `<input type="number">` définit un champ **numérique** de saisie .

Vous pouvez également définir des restrictions sur les numéros acceptés.

L'exemple suivant affiche un champ de saisie numérique dans lequel vous pouvez saisir une valeur comprise entre 1 et 5:

```
<form>
  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1"
max="5">
</form>
```

Il existe également les types URL, heure, mois, semaine...

Exercice:

Faire un formulaire de réservation de voyage avec :

1. nom
2. prénom
3. Sélectionner une formule de voyage
4. Date d'arrivée, date de départ
5. Nombre de personnes
6. Choix de la chambre : 2, 3 ou 4 lits, vue sur la mer
7. Code de coupon de réduction
8. Accepter les conditions
9. Bouton final pour valider la réservation
10. Ajouter du style

Elément summary

```
<details>
  <summary>Centre de Vacances</summary>
  <p>Notre centre de vacances disposent de supers éléments.</p>
</details>
```

Mise en place d'un grid

Après avoir vu surtout des éléments HTML, nous allons maintenant nous concentrer sur le CSS. Il existe plusieurs moyens de répartir les éléments d'une page HTML en sections. Nous avons vu les tableaux. Il y a également le système FlexBox et Grid. Nous présentons ici un exemple avec le grid qui est très modulable.

```
<div class="container">
  <div class="cell cell-1">1.</div>
  <div class="cell cell-2">2.</div>
  <div class="cell cell-3">3.</div>
  <div class="cell cell-4">4.</div>
  <div class="cell cell-5">5.</div>
  <div class="cell cell-6">6.</div>
  <div class="cell cell-7">7.</div>
  <div class="cell cell-8">8.</div>
  <div class="cell cell-9">9.</div>
</div>
```

```
.container {
  height: 90vh;
  margin: 2rem;
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 1fr 1fr 1fr;
}
.cell {
  color: white;
  font-size: 3rem;
  text-align: center;
  padding: 4rem;
}
.cell-1 {
  background: deepskyblue;
}
.cell-2 {
  background: orangered;
}
.cell-3 {
  background: royalblue;
}
.cell-4 {
  background: gold;
}
.cell-5 {
  background: blueviolet;
}
.cell-6 {
  background: limegreen;
}
.cell-7 {
  background: coral;
}
.cell-8 {
  background: lightseagreen;
}
.cell-9 {
  background: maroon;
}
```

fr signifie fractional. Il représente une fraction de la page répartie équitablement avec les autres.

Nous pouvons modifier à la fois la taille et la disposition des éléments de la grille en ajoutant de nouvelles valeurs aux propriétés `grid-template-columns` et `grid-template-rows`.

```
.container {  
  grid-template-columns: 2fr 1fr 1fr 200px;  
  grid-template-rows: 1fr 1fr 1fr;  
}
```

La propriété `grid-gap` est un raccourci pour `grid-row-gap` et `grid-column-gap`. Nous pouvons l'utiliser pour ajouter rapidement des espaces entre les pistes de la grille.

```
.container {  
  grid-gap: 10px 10px;  
}
```

Nous pouvons spécifier la taille et l'emplacement des cellules individuelles de la grille à l'aide des propriétés de ligne de grille et de colonne de grille. Nous devons ajouter ces deux propriétés aux cellules de la grille que nous voulons positionner.

Voyons d'abord un cas simple. Supposons que nous souhaitons déplacer la cellule 1 vers la deuxième ligne et la troisième colonne de la grille (à l'emplacement actuel de la cellule 6). Pour ce faire, nous devons simplement ajouter les deux règles CSS suivantes à `.cell-1`:

```
.cell-1 {  
  grid-row: 2;  
  grid-column: 3;  
}
```

On veut maintenant que la cellule 1 s'étende sur la première ligne:

```
.cell-1 {  
  grid-row: 1;  
  grid-column: 1/4;  
}
```

Nous pouvons étendre la cellule 2 sur les première et deuxième colonnes et les deuxième et troisième lignes en utilisant le code suivant:

```
.cell-2 {  
  grid-row: 2/4;  
  grid-column: 1/3;  
}
```

Une autre façon d'utiliser le grid :

Nous aimerions élaborer une mise en page de blog simple avec un en-tête, un pied de page, une zone de contenu principale et deux barres latérales (une à gauche et une à droite). Bien que ce ne soit pas une mise en page très compliquée, il peut être encore plus confortable de faire référence aux cellules de la grille par leur nom.

```
<div class="container">  
  <header class="cell cell-1">Header</header>  
  <aside class="cell cell-2">Left sidebar</aside>  
  <main class="cell cell-3">Main content</main>  
  <aside class="cell cell-4">Right sidebar</aside>  
  <footer class="cell cell-5">Footer</footer>  
</div>
```

Dans le CSS, nous devons utiliser deux nouvelles propriétés: **grid-template-areas** sur le conteneur de grille et **grid-area** sur chaque zone de grille individuelle, séparément.

rem signifie racine **em** ce qui est la largeur d'une lettre majuscule M par défaut dans le navigateur.

1 **rem** est égal la taille des lettres html ce qui équivaut en général à 16px.

Il existe aussi la mesure **em**. Mais la différence est que la mesure **em** est relative à la précédente inscrite dans le css. Donc elle ne fera pas référence à une valeur absolue au contraire de **rem**.

vh et **vw** - De nombreuses techniques de conception Web réactive reposent fortement sur des règles de pourcentage. Cependant, les mesures de pourcentage CSS ne sont pas toujours la meilleure solution pour tous les problèmes. La mesure **vh** est égale à 1/100 de la hauteur de la fenêtre. Ainsi, par exemple, si la hauteur du navigateur est de 800px, 1vh équivaut à 8px et, de même, si la largeur de la fenêtre est de 650px, 1vw équivaut à 6,5px.

Vous pouvez aussi déterminer une taille en fonction du parent :

```
div {  
  width: 400px;  
}  
div p {  
  width: 75%;  
}
```

Puisque la largeur du parent div est 400px, la largeur du paragraphe intérieur sera 300px, soit 75% de 400px.

```
.container {  
  height: 90vh;  
  width: 90vw;  
  margin: 2rem auto;  
  display: grid;  
  grid-template-areas: "header header header"  
                      "left main right"  
                      "footer footer footer";  
  grid-template-rows: 1fr 3fr 1fr;  
  grid-template-columns: 1fr 2fr 1fr;  
}  
.cell-1 {  
  background: deepskyblue;  
  grid-area: header;  
}  
.cell-2 {  
  background: orangered;  
  grid-area: left;  
}  
.cell-3 {  
  background: limegreen;  
  grid-area: main;  
}
```

```
.cell-4 {  
  background: gold;  
  grid-area: right;  
}  
.cell-5 {  
  background: blueviolet;  
  grid-area: footer;  
}  
.cell {  
  color: white;  
  font-size: 2.5rem;  
  text-align: center;  
  padding: 4rem;  
}
```

On peut finalement créer des grilles imbriquées les unes dans les autres. Il suffit pour cela d'utiliser les éléments grid qui sont par exemple dans la cellule 5 de la même manière que dans la classe container.

```
.cell-5 {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr 1fr;  
  grid-column-gap: 1rem;  
}  
.cell-5 .cell {  
  background-color: coral;  
}
```

Barre de navigation CSS

Nous allons mettre en place une simple barre de navigation.

html :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Barre de navigation</title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Sofia">
  </head>
  <body>
    <nav id="topnav">
      <a id="logo" class="nav-link active" href="#">LOGO</a>
      <a class="nav-link" href="#">Page 1</a>
      <a class="nav-link" href="#">Page 2</a>
      <a class="nav-link" href="#">Page 3</a>
      <a class="nav-link" href="#">Page 4</a>
      <a class="nav-link" href="#">Page 5</a>

      <a id="about" class="nav-link" href="#">About Me</a>
    </nav>
  </body>
</html>
```

Bien sûr, si vous ne souhaitez pas utiliser id et classes, vous pouvez utiliser plusieurs classes appelées sur un même élément :

```
<div class="class1 class2 class3"></div>
```

ou faire selon cette syntaxe :

```
<div class="social">
  <div class="socialIcon">
    
  </div>
  <div class="socialText">Find me on Facebook</div>
</div>
```

```
.social {
width:330px;
height:75px;
float:right;
text-align:left;
padding:10px 0;
border-bottom:dotted 1px #6d6d6d;
}

.social .socialIcon{
padding-top:0; }

.social .socialText{
border:0;
}
```


style.css

```
#topnav {
  width: 100%;
  position: fixed;
  top: 0;
  left: 0;
  background-color: rgba(45, 86, 175, 0.836);
  font-family: "Sofia";
  font-size: 20px;
  text-shadow: 3px 3px 3px #c48686;
}

.nav-link {
  display: inline-block;
  width: 100px;
  height: 55px;
  color: White;

  /* centering text */
  text-align: center;    /* horizontally */
  line-height: 55px;    /* vertically */

  text-decoration: none;
}

nav a:hover {
  background-color: grey;
  transition: background-color .3s;
  font-size: 20px;
  border-radius : 20px;
}

.active {
  background-color: rgb(38, 216, 47);
}

#logo {
  width: 120px;
  background-color: RoyalBlue;
  font-weight: bold;
}
```

TP : Mettre en place le style du formulaire de contact

Exercice :

Mettre en forme le formulaire de voyage fait précédemment avec le CSS

Voici notre version :

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>My Example</title>

<!-- CSS -->
<style>
.myForm {
font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
font-size: 0.8em;
width: 30em;
padding: 1em;
}

.myForm * {
box-sizing: border-box;
}

.myForm label {
padding: 0;
font-weight: bold;
text-align: right;
display: block;
}
```

```

.myForm input,
.myForm select,
.myForm textarea {
margin-left: 2em;
float: right;
width: 20em;
border: 1px solid #ccc;
font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
font-size: 0.9em;
padding: 0.3em;
}

.myForm textarea {
height: 100px;
}

.myForm button {
padding: 1em;
border-radius: 0.5em;
background: #eee;
border: none;
font-weight: bold;
margin-left: 14em;
margin-top: 1.8em;
}

.myForm button:hover {
background: #ccc;
}
</style>

</head>
<body>

<form class="myForm">

<p>
<label>Name
<input type="text" name="customer_name" required>
</label>
</p>

<p>
<label>Email
<input type="email" name="email_address">
</label>

```

```
</p>

<p>
<label>Enquiry Regarding
<select>
<option value="" selected="selected">Select One</option>
<option value="website" >Our Website</option>
<option value="membership" >Membership</option>
<option value="telepathy" >We'll Guess!</option>
</select>
</label>
</p>

<p>
<label>Enquiry
<textarea name="comments" maxlength="500"></textarea>
</label>
</p>

<p><button>Submit Enquiry</button></p>

</form>

</body>
</html>
```

Mise en place d'un footer avec Grid

Nous allons maintenant progressivement élaborer un footer avec Grid.

Je vous transmets la base HTML sur laquelle nous appliquer le style :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
integrity="sha384-UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5d
Q8lWUE00s/" crossorigin="anonymous">
  <link
href="https://fonts.googleapis.com/css?family=Lato|Merriweather"
rel="stylesheet">
  <link rel="stylesheet" href="style.css">
  <title>Footer1</title>
</head>
<body>
  <footer class="footer">
    <div class="footer-container">
      <div class="footer-logo">
        
      </div>
      <ul class="footer-nav">
        <li><a href="#">Home</a></li>
        <li><a href="#">About Us</a></li>
        <li><a href="#">Case studies</a></li>
        <li><a href="#">Our Work</a></li>
        <li><a href="#">Blog</a></li>
      </ul>
      <ul class="footer-secondary-nav">
        <li><a href="#">Latest News</a></li>
        <li><a href="#">Our Partners</a></li>
        <li><a href="#">Jobs</a></li>
        <li><a href="#">Freebies</a></li>
      </ul>
    </div>
  </footer>
</body>
</html>
```

```

        <li><a href="#">Fun stuff</a></li>
    </ul>
    <div class="contact-block">
        <h3>We're happy to hear from you!</h3>
        <a href="tel:999-999-4321"><i class="fas fa-phone"></i>
999-999-4321</a>
        <a
href="mailto:info@mycompany.xyz">info@mycompany.xyz</a>
    </div>
    <div class="newsletter">
        <div class="newsletter-text">
            <div class="icon">
                <i class="fas fa-paper-plane"></i>
            </div>
            <p>
                Subscribe to our Newsletter
            </p>
        </div>
        <div class="text-input">
            <form action="post">
                <input type="email" name="Enter Your Email
Address" size="20">
                <input type="submit" value="Submit">
            </form>
        </div>
    </div>
    <div class="social-networks">
        <h3>Share our content!</h3>
        <div class="facebook">
            <i class="fab fa-facebook"></i>
        </div>
        <div class="twitter">
            <i class="fab fa-twitter"></i>
        </div>
        <div class="instagram">
            <i class="fab fa-instagram"></i>
        </div>
    </div><!-- .social-networks-->
</div><!-- .footer-container -->
</footer><!-- .footer -->
<div class="bottom-line">

    <div class="legal">
        <a href="#">Terms of use</a>
        <a href="#">Privacy</a>
        <a href="#">Legal</a>
    </div>

```

```
        </div>
    </div>
</body>
</html>
```

Création fichier style.css

```
* {
padding: 0; margin: 0;
}

li {
list-style-type: none;
}

a {
text-decoration: none;
}

body {
font-family: 'Lato', sans-serif;
}

h1, h2, h3, h4, h5, h6 {
font-family: 'Merriweather', serif;
}

.footer {
color: #fff; background: #000; padding: 4em 0;
}
```

Le premier élément que vous devez cibler est le conteneur de pied de page. Il comporte 3 colonnes et 2 lignes. Les première et dernière colonnes sont destinées à servir de marges de la mise en page.

```
.footer {
display: grid;
grid-template-columns: 7vw 1fr 7vw;
grid-template-rows: auto auto;
}
```

Après cela, vous devez cibler l'élément `footer-container`, qui contient tous les autres «blocs» de pied de page, et le placer entre les deuxième et troisième lignes de colonne.

```
.footer-container {  
  grid-column: 2 / 3;  
}
```

Le footer-container est une grille imbriquée avec 4 colonnes et 2 lignes. La mise en page sera construite à l'aide de la propriété `grid-template-areas`, afin de définir les zones auxquelles le contenu sera affecté.

```
.footer-container {  
  grid-column: 2 / 3;  
  display: grid;  
  grid-template-columns: 2fr 2fr 2fr 3fr;  
  grid-template-rows: auto auto;  
  grid-template-areas: "logo navigation secondary social" "contact contact . newsletter";  
}
```

Notez que vous avez centré horizontalement les deux éléments de navigation avec la propriété `justify-self`.

L'élément de réseaux sociaux est une grille imbriquée 2x3. Placez les icônes de titre et de médias sociaux à leur emplacement correspondant sur la grille.

```
.social-networks {  
  grid-area: social;  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-template-rows: auto auto;  
  justify-items: center;  
}  
  
.social-networks h3 {  
  grid-column: 1 / -1;
```



```
}  
  
.facebook {  
  justify-self: right;  
}  
  
.instagram {  
  justify-self: left;  
}
```

Placez le bloc de contacts et les éléments de la **newsletter** au bas de leurs zones, de sorte que les deux s'alignent.

```
.contact-block, .newsletter {  
  align-self: end;  
}
```

Il est maintenant temps de cibler le **bottom-line** container. Vous devez l'aligner avec le contenu **footer-container**.

```
.bottom-line {  
  display: grid;  
  grid-template-columns: 7vw 8fr 4fr 7vw;  
}
```

Le conteneur **legal** avec 3 liens est également une grille imbriquée à 3 colonnes. Les 3 éléments sont justifiés à droite.

```
.legal {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  justify-items: end;  
}
```

Vous avez tous les styles de grille en place. Voyons maintenant comment améliorer un peu le style

Pour la newsletter :

```
.newsletter-text, .text-input {
text-align: center;
}

.text-input input {
border-radius: 0;
}

.text-input input:first-child {
height: 2rem; }

.text-input input:last-child {
height: 2.25rem;
color: #fff;
border: none;
background-color: #f99639;
width: 5rem;
font-weight: bold;
}

.fas, .fab {
font-size: 1.5rem;
}

.newsletter-text icon, .newsletter-text p {
margin: 0 0 1.1rem;
}

.newsletter-text p {
line-height: 2rem;
}
```

Changez la couleur de tous les liens, enveloppez le téléphone et l'adresse e-mail du contact dans une bordure «en forme de bouton» et centrez les deux textes. Donnez au conteneur des réseaux sociaux une bordure en bas. De cette façon, il y aura une séparation claire entre les sections réseaux sociaux et newsletter.

```
.footer a {  
  color: #fff;  
}  
  
.contact-block a {  
  display: block;  
  border: 2px solid #fff;  
  padding: 0.5rem;  
  margin: 10px 0 0;  
  text-align: center;  
  width: 70%;  
}  
  
.contact-block a:hover {  
  border: 2px solid #f99639;  
  color: #f99639;  
}  
  
.social-networks {  
  border-bottom: 2px solid #f99639;  
}
```

Il n'y a que l'arrière-plan et la couleur du texte de la ligne du bas à gauche à styliser.

```
.bottom-line {  
  background-color: #444554;  
}  
  
.bottom-line, .bottom-line a {  
  color: #fff;  
}
```

Bootstrap

Bootstrap est le framework HTML, CSS et JavaScript le plus populaire pour le développement de sites Web réactifs et adaptés aux mobiles.

Bootstrap est entièrement gratuit à télécharger et à utiliser!

- Bootstrap est un framework frontal gratuit pour un développement Web plus rapide et plus facile
- Bootstrap comprend des modèles de conception basés sur HTML et CSS pour la typographie, les formulaires, les boutons, les tableaux, la navigation, les modaux, les carrousels d'images et bien d'autres, ainsi que des plugins JavaScript facultatifs
- Bootstrap vous donne également la possibilité de créer facilement des conceptions réactives

Site : <https://getbootstrap.com/>

Qu'est-ce que le Responsive Web Design?

La conception Web réactive consiste à créer des sites Web qui s'ajustent automatiquement pour avoir une belle apparence sur tous les appareils, des petits téléphones aux grands ordinateurs de bureau.

Historique Bootstrap

Bootstrap a été développé par Mark Otto et Jacob Thornton sur Twitter, et publié en tant que produit open source en août 2011 sur GitHub.

En juin 2014, Bootstrap était le projet n ° 1 sur GitHub!

Pourquoi utiliser Bootstrap?

Avantages de Bootstrap:

- **Facile à utiliser:** toute personne ayant juste des connaissances de base en HTML et CSS peut commencer à utiliser Bootstrap

- **Fonctionnalités** : réactives le CSS réactif de Bootstrap s'adapte aux téléphones, tablettes et ordinateurs de bureau
 - **Approche mobile d'abord**: dans Bootstrap 3, les styles mobiles d'abord font partie du cadre de base
 - **Compatibilité du navigateur**: Bootstrap est compatible avec tous les navigateurs modernes (Chrome, Firefox, Internet Explorer, Edge, Safari et Opera)
-

Système de grille Bootstrap

Le système de grille de Bootstrap autorise jusqu'à 12 colonnes sur la page.

Si vous ne souhaitez pas utiliser les 12 colonnes individuellement, vous pouvez regrouper les colonnes pour créer des colonnes plus larges:

Le système de grille de Bootstrap est réactif et les colonnes se réorganiseront automatiquement en fonction de la taille de l'écran.

Grid Classes

Le système de grille Bootstrap comprend quatre classes:

- xs (pour les téléphones - écrans de moins de 768 pixels de large)
- sm (pour les tablettes - écrans d'une largeur égale ou supérieure à 768 pixels)
- md (pour les petits ordinateurs portables - écrans d'une largeur égale ou supérieure à 992 pixels)
- lg (pour les ordinateurs portables et les ordinateurs de bureau - écrans d'une largeur égale ou supérieure à 1200 pixels)

Les classes ci-dessus peuvent être combinées pour créer des mises en page plus dynamiques et flexibles.

Structure de base d'une grille Bootstrap

Voici une structure de base d'une grille Bootstrap:

```
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  ...
</div>
```

D'abord; créer une ligne (`<div class="row">`). Ensuite, ajoutez le nombre de colonnes souhaité (balises avec `.col-*-*` classes). Notez que les nombres dans `.col-*-*` doit toujours ajouter jusqu'à 12 pour chaque ligne.

Ci-dessous, nous avons rassemblé quelques exemples de dispositions de base de la grille Bootstrap.

Trois colonnes égales

L'exemple suivant montre comment obtenir trois colonnes de largeur égale à partir de tablettes et mise à l'échelle vers de grands bureaux. Sur les téléphones mobiles ou les écrans de moins de 768 pixels de large, les colonnes s'empileront automatiquement:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css">
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"><
/script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js
"></script>
</head>
<body>

<div class="container-fluid">
  <h1>Hello World!</h1>
  <p>Resize the browser window to see the effect.</p>
  <p>The columns will automatically stack on top of each other when the
screen is less than 768px wide.</p>
  <div class="row">
    <div class="col-sm-4"
style="background-color:lavender;">.col-sm-4</div>
    <div class="col-sm-4"
style="background-color:lavenderblush;">.col-sm-4</div>
    <div class="col-sm-4"
style="background-color:lavender;">.col-sm-4</div>
  </div>
</div>

</body>
</html>
```

Deux colonnes inégales

L'exemple suivant montre comment obtenir deux colonnes de différentes largeurs à partir de tablettes et à l'échelle de grands écrans:

```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-8">.col-sm-8</div>
</div>
```

Créer un Jumbotron

Un jumbotron indique une grande boîte pour attirer une attention particulière sur certains contenus ou informations spéciaux.

Un jumbotron s'affiche sous la forme d'une boîte grise aux coins arrondis. Il agrandit également les tailles de police du texte qu'il contient.

Astuce: à l'intérieur d'un jumbotron, vous pouvez mettre presque n'importe quel HTML valide, y compris d'autres éléments / classes Bootstrap.

Utiliser un élément `<div>` avec classe `.jumbotron` pour créer un jumbotron:

Conteneur intérieur Jumbotron

Placez le jumbotron à l'intérieur du `<div class="container">` si vous voulez que le jumbotron NE s'étende PAS jusqu'au bord de l'écran:

```
<div class="container">
  <div class="jumbotron">
    <h1>Bootstrap Tutorial</h1>
    <p>Bootstrap is the most popular HTML, CSS, and JS framework for
developing
    responsive, mobile-first projects on the web.</p>
  </div>
  <p>This is some text.</p>
  <p>This is another text.</p>
</div>
```


Conteneur extérieur Jumbotron

Placez le jumbotron à l'extérieur du `<div class="container">` si vous voulez que le jumbotron s'étende jusqu'aux bords de l'écran:

```
<div class="jumbotron">
  <h1>Bootstrap Tutorial</h1>
  <p>Bootstrap is the most popular HTML, CSS, and JS framework for
  developing responsive,
  mobile-first projects on the web.</p>
</div>
<div class="container">
  <p>This is some text.</p>
  <p>This is another text.</p>
</div>
```

Créer un en-tête de page

Un en-tête de page est comme un séparateur de section.

le `.page-header` classe ajoute une ligne horizontale sous l'en-tête (+ ajoute un espace supplémentaire autour de l'élément):

Exemple d'en-tête de page

Utiliser un élément `<div>` avec la classe `.page-header` pour créer un en-tête de page:

```
<div class="page-header">
  <h1>Example Page Header</h1>
</div>
```

Barre de Navigation Bootstrap

Avec Bootstrap, une barre de navigation peut s'étendre ou se réduire, selon la taille de l'écran.

Une barre de navigation standard est créée avec `<nav class="navbar navbar-default">`.

L'exemple suivant montre comment ajouter une barre de navigation en haut de la page:

```
<nav class="navbar navbar-default">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">Nom du site</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">Page 1</a></li>
      <li><a href="#">Page 2</a></li>
      <li><a href="#">Page 3</a></li>
    </ul>
  </div>
</nav>
```

Ou une barre de navigation noire :

Changez simplement la classe `.navbar-default` par `.navbar-inverse`

Barre de navigation avec liste déroulante

Les barres de navigation peuvent également contenir des menus déroulants.

L'exemple suivant ajoute un menu déroulant pour la "Page 1" bouton:

```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">WebSiteName</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Home</a></li>
      <li class="dropdown">
        <a class="dropdown-toggle" data-toggle="dropdown" href="#">Page 1
        <span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="#">Page 1-1</a></li>
          <li><a href="#">Page 1-2</a></li>
          <li><a href="#">Page 1-3</a></li>
        </ul>
      </li>
      <li><a href="#">Page 2</a></li>
      <li><a href="#">Page 3</a></li>
    </ul>
  </div>
</nav>
```

Vous avez énormément de possibilités différentes pour beaucoup d'éléments sur Bootstrap pour des grilles, des couleurs, des thèmes, des footers...

Nous allons juste en voir quelques-uns :

Formulaires Bootstrap

Les contrôles de formulaire individuels reçoivent automatiquement un style global avec Bootstrap.

Tous les textes <input>, <textarea>, et les éléments <select> avec class = "form-control" sont définis sur width: 100%; par défaut.

Règles standard pour les trois mises en page de formulaire:

- Enveloppez les étiquettes et les contrôles de formulaire dans `<div class="form-group">`(nécessaire pour un espacement optimal)
- Ajouter une classe `.form-control` à tous les éléments `<input>`, `<textarea>`, et `<select>`

L'exemple suivant crée un formulaire Bootstrap simple avec deux champs de saisie, une case à cocher et un bouton d'envoi:

```
<form>
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Remember me</label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

Comment créer un carousel Bootstrap

L'exemple suivant montre comment créer un carrousel de base:

```
<div id="myCarousel" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0"
class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
  </ol>

  <!-- Wrapper for slides -->
  <div class="carousel-inner">
    <div class="item active">
```

```

</div>

<div class="item">

</div>

<div class="item">

</div>
</div>

<!-- Left and right controls -->
<a class="left carousel-control" href="#myCarousel" data-slide="prev">
  <span class="glyphicon glyphicon-chevron-left"></span>
  <span class="sr-only">Previous</span>
</a>
<a class="right carousel-control" href="#myCarousel"
data-slide="next">
  <span class="glyphicon glyphicon-chevron-right"></span>
  <span class="sr-only">Next</span>
</a>
</div>
```

TP FINAL

Réaliser un site CV :

- 3 pages
 - ◆ Accueil
 - ◆ CV
 - ◆ Contact
- Barre de Navigation
- Footer

Voici les éléments de base de la page CV

