



insy2s

# NURSERIE

A LA DÉCOUVERTE DES MÉTIERS DU DIGITAL

# Le parcours

Découvrir différents métiers et leurs  
composantes au travers des  
présentations et de la pratique  
personnelle

## Découverte du métier **d'intégrateur web** via la pratique

### Sommaire :

- Présentation du HTML
- Premiers essais en HTML
- La structure d'une page HTML
- Créer une structure en HTML
- Créer une maquette
- Découverte du CSS et application

## Présentation du WEB

Il faut savoir que le Web fait partie d'Internet.

Internet est un grand ensemble qui comprend, entre autres : le Web, les e-mails, la messagerie instantanée, etc.

Les langages HTML et CSS sont à la base du fonctionnement de tous les sites web.

Le langage HTML a été inventé par un certain Tim Berners-Lee en 1991, le CSS et le JavaScript en 1996.

La 1ère page web créée :

<http://info.cern.ch/hypertext/WWW/TheProject.html>

## Présentation du WEB

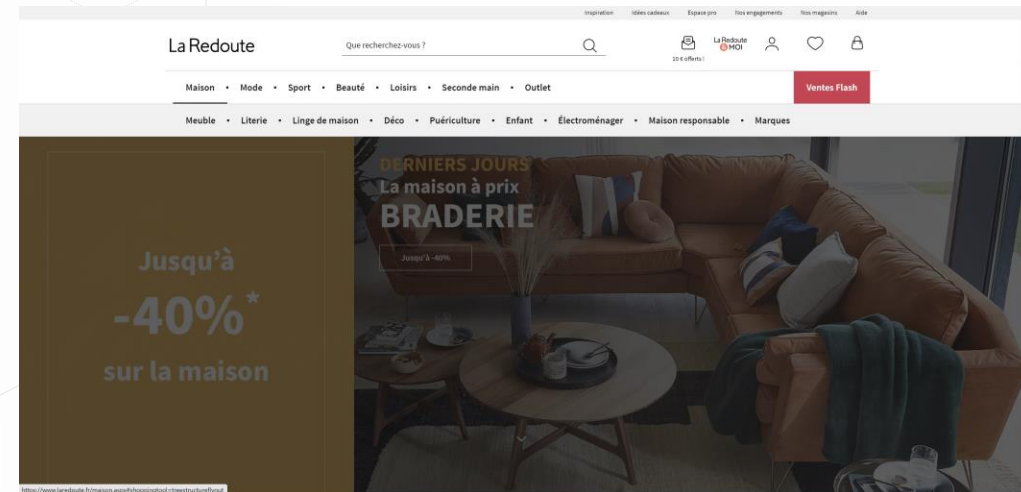
HTML, CSS et JavaScript sont les seuls langages interprétés par les navigateurs (Firefox, Safari, Edge, Opera, Chrome...).

Code



Interprétation

Affichage



(Clic droit dans une page web -> afficher le code source)

## Présentation du WEB

**HyperText Markup Language (HTML)** est le code utilisé pour structurer une page web et son contenu.

Par exemple, le contenu de votre page pourra être structuré en un ensemble de paragraphes, une liste à puces ou avec des images et des tableaux de données.

## Présentation du WEB

**HTML n'est pas un langage de programmation. C'est un *langage de balises* qui définit la structure de votre contenu.**

L'« hypertexte » désigne les liens qui relient les pages web entre elles, que ce soit au sein d'un même site web ou entre différents sites web. Les liens sont un aspect fondamental du Web. Ce sont eux qui forment cette « toile » (ce mot est traduit par *web* en anglais).

En téléchargeant du contenu sur l'Internet et en le reliant à des pages créées par d'autres personnes, vous devenez un participant actif du World Wide Web.

## Présentation du HTML

### Composition d'une balise

#### 1. Balise paire :

**<balise>** contenu.... **</balise>**

Balise ouvrante

Balise fermante

< : chevron ouvrant  
> : chevron fermant

exemple : **<p>**Ceci est un paragraphe.**</p>**

#### 2. Balise orpheline ou auto-fermante :

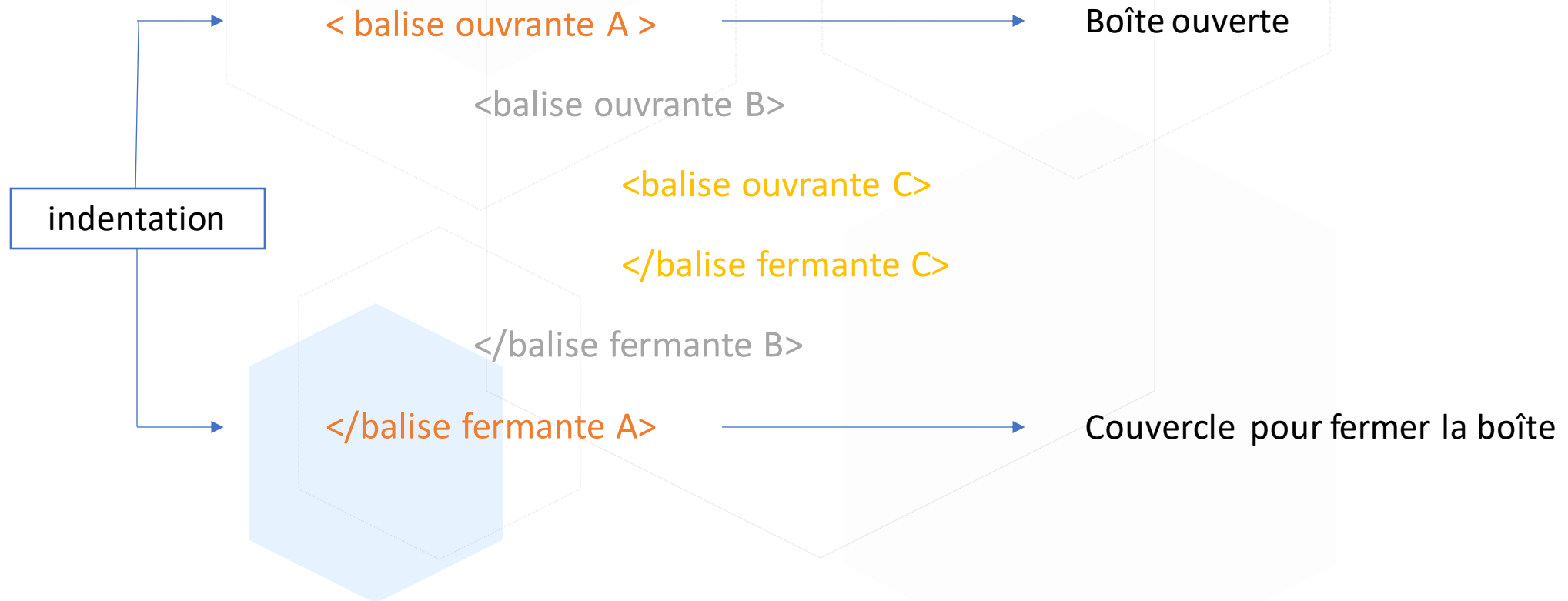
**<balise>** ou **<balise />**

exemple : **<br />** (balise de retour à la ligne)



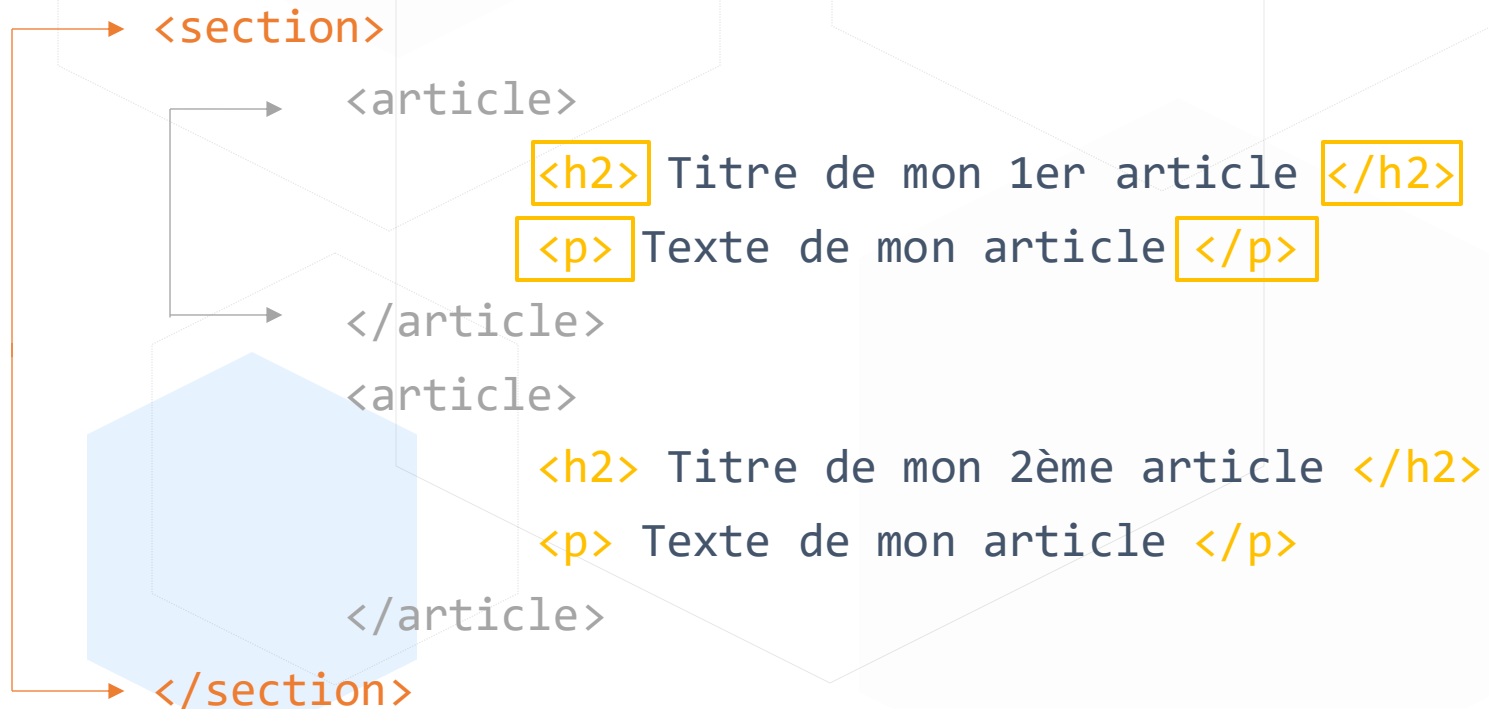
## Présentation du HTML

### Imbrication des balises



## Présentation du HTML

### Imbrication des balises : exemple



```

<section>
  <article>
    <h2> Titre de mon 1er article </h2>
    <p> Texte de mon article </p>
  </article>
  <article>
    <h2> Titre de mon 2ème article </h2>
    <p> Texte de mon article </p>
  </article>
</section>
  
```

## Présentation du HTML

### 1<sup>er</sup> test : Créer un nouveau document html avec VSCode

#### Exemple text brut

```
test1.html
1 Hello World !
2 Je suis un paragraphe.
3 Je suis la deuxième ligne du paragraphe.
4 Je suis la troisième et dernière ligne du paragraphe.
```



Hello World ! Je suis un paragraphe. Je suis la deuxième ligne du paragraphe. Je suis la troisième et dernière ligne du paragraphe.

#### Résultats dans le navigateur

#### Exemple text avec balises

```
test1.html > p
1 <h1>Hello World !</h1>
2 <p>Je suis un paragraphe.<br />
3 Je suis la deuxième ligne du paragraphe.<br />
4 Je suis la troisième et dernière ligne du paragraphe.
5 </p>
```



# Hello World !

Je suis un paragraphe.  
Je suis la deuxième ligne du paragraphe.  
Je suis la troisième et dernière ligne du paragraphe.

## Présentation du HTML

### Les attributs

- ◆ Tous les éléments HTML peuvent avoir des **attributs**
- ◆ Les attributs fournissent **des informations supplémentaires** sur les éléments
- ◆ Les attributs sont toujours spécifiés dans la **balise de début**
- ◆ Les attributs se présentent généralement sous forme de paires nom/valeur telles que : **attribut="valeur"**

## Présentation du HTML

### Les attributs

Exemples

```
<a href="https://www.w3schools.com">Visiter Notre Site</a>
```

```

```

**NB :** On peut avoir autant d'attributs qu'on veut pour une balise donnée

```

```

## Présentation du HTML

### NOTRE PREMIER CODE

Structure de base d'une page web

Notre première page web

Zoom sur le « Head »

Zoom sur le « Body »

## Présentation du HTML

### Structure d'une page HTML

Structure minimaliste d'une page web :

**<!DOCTYPE html>** : document de type HTML5

**<html> ... </html>** : racine du document. Indique où commence et fini notre page.

**<head> ... </head>** : l'en-tête contient des informations sur notre page.

Il peut contenir des liens vers des feuilles de style et du JavaScript.

**<body> ... </body>** : les informations dans ces balises seront affichées aux utilisateurs

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</body>

</html>
```

## Présentation du HTML

### Structure complète

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="css/style.css">
  <title>Hello, world!</title>
</head>
<body>
  <h1>Hello, world!</h1>

  <script src="js/main.js"></script>
</body>
</html>
```



## Présentation du HTML

### Notre première page web

Créer un fichier .html avec VSCode, générer la structure de base, enregistrer et l'ouvrir dans le navigateur.

( raccourci emmet : `! + entrer` ou `html:5 + entrer` )

## Présentation du HTML

### La balise « head »

La balise "head", contient des informations au sujet de la page ou l'intégration de fichiers utiles à la page. Ce bloc n'est pas affiché sur l'écran.

**<title>Titre de ma page</title>**

Indique le titre de la page, généralement visible dans l'onglet de la page ou lorsque la page est ajoutée dans les favoris.

**<meta charset="utf-8" />**

Indique que le jeu de caractère de la page. UTF-8 qui est maintenant le jeu le plus utilisé sur internet permet d'écrire du texte dans quasiment toutes les langues.

**<meta name="keywords" content="restaurant pizza tiramisu">**

Meta pour les moteurs de recherche. Le site sera mieux référencé lors des recherches.

**<meta name="description" content="Joli restaurant en bord de plage">**

Les balises méta fournissent des informations, soit pour le navigateur soit pour les moteurs de recherche.

## Présentation du HTML

### La balise « head »

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

**link** : que le navigateur doit chercher un document

**rel** : le document est une feuille de style (stylesheet)

**href** : chemin vers le fichier

```
<script type="text/javascript" src="js/main.js"></script>
```

**script** : permet d'ajouter un script dans la page

**type** : Par défaut le langage est le JavaScript mais il est conseillé de l'indiquer

**src** : le chemin vers le fichier.

Nous plaçons généralement la balise "script" dans le corps (<body></body>) de la page.

## Présentation du HTML

### La balise « body »

La balise "**body**" contient les éléments devant être affichés à l'utilisateur.

En reprenant le code ci-dessus, il est normal de ne rien voir. (la balise body est vide)

HTML ne dit pas comment afficher les balises, il dit seulement : voilà comment est structurée ma page.

Deux grandes classes d'éléments :

#### Block

Prennent toute la largeur disponible

Provoque un retour à la ligne

Peuvent contenir des éléments inline

#### Inline

Ne provoque pas de retour à la ligne

Ne peuvent pas contenir des éléments block

Peuvent contenir des éléments inline

BLOCK-LEVEL ELEMENTS:



INLINE ELEMENTS:



## Présentation du HTML

### LES BALISES

Les titres

Les paragraphes

Les liens

Les images

Les tableaux

Les listes

Les commentaires

# Présentation du HTML

## LES TITRES

Les titres HTML sont définis avec les balises `<h1>` à `<h6>`.

`<h1>` définit le titre le plus important, `<h6>` définit le titre le moins important.

`<h1>`Heading 1`</h1>`

`<h2>`Heading 2`</h2>`

`<h3>`Heading 3`</h3>`

`<h4>`Heading 4`</h4>`

`<h5>`Heading 5`</h5>`

`<h6>`Heading 6`</h6>`

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

## Présentation du HTML

### LES PARAGRAPHES

La balise `<p>` définit un paragraphe.

`<p>`C'est un paragraphe`</p>`

**Exercice titres / paragraphes**

# Présentation du HTML

## LES LIENS

Les liens permettent aux utilisateurs de cliquer dessus pour se déplacer de page en page.

La balise `<a>` en HTML définit un lien hypertext. Il a la syntaxe suivante :

```
<a href="url">texte du lien</a>
```

L'attribut le plus important de l'élément `<a>` est le `href`, qui indique la destination du lien. Le *texte du lien* est la partie qui sera visible par l'utilisateur.

En cliquant sur le texte du lien, l'utilisateur sera envoyé à l'adresse URL spécifiée.

### Exemple

```
<a href="https://www.w3schools.com/">Visit W3Schools.com !</a>
```

L'attribut `target` spécifie où ouvrir le document lié. Il accepte plusieurs valeurs dont :

- `_self` : Par défaut. Ouvre le document dans la page en cours
- `_blank` : Ouvre le document dans un nouvel onglet

### Exemple

```
<a href="https://www.google.fr" target="blank">lien dans un nouvel onglet</a>
```



# Présentation du HTML

## LES LIENS

### - URL absolues vs relatives :

**<h2>Absolute URLs</h2>**

**<a href="https://www.w3.org">W3C</a>**

**<a href="https://www.google.com">Google</a>**

Liens vers d'autres sites

**<h2>Relatives URLs</h2>**

**<a href="page2.html">Page 2</a>**

**<a href="contacts.html">Contacts</a>**

Liens vers d'autres pages du site

Pour utiliser une image comme lien, il suffit de mettre la balise **<img>** à l'intérieur de la balise **<a>** :

**<a href="https://www.google.com">**

****

**</a>**

Il est également possible de gérer le **href** pour créer un lien directement vers une adresse mail, ou un numéro de tel.

# Présentation du HTML

## LES IMAGES

La balise `<img />` a deux attributs obligatoires :

- `src` : spécifie le chemin d'accès à l'image
- `alt` : spécifie un texte alternatif pour l'image

**n.b** : La balise `<img />` doit souvent être à l'intérieur d'une balise de type bloc, soit à l'intérieur de la balise `<p></p>` soit à l'intérieur de la balise `<figure></figure>`

### Exemple

```

```

Taille de l'image - Largeur et hauteur :

```

```

Exercice liens / images

# Présentation du HTML

## LES TABLEAUX

Structure de base d'un tableau HTML :

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

**table** : début du tableau

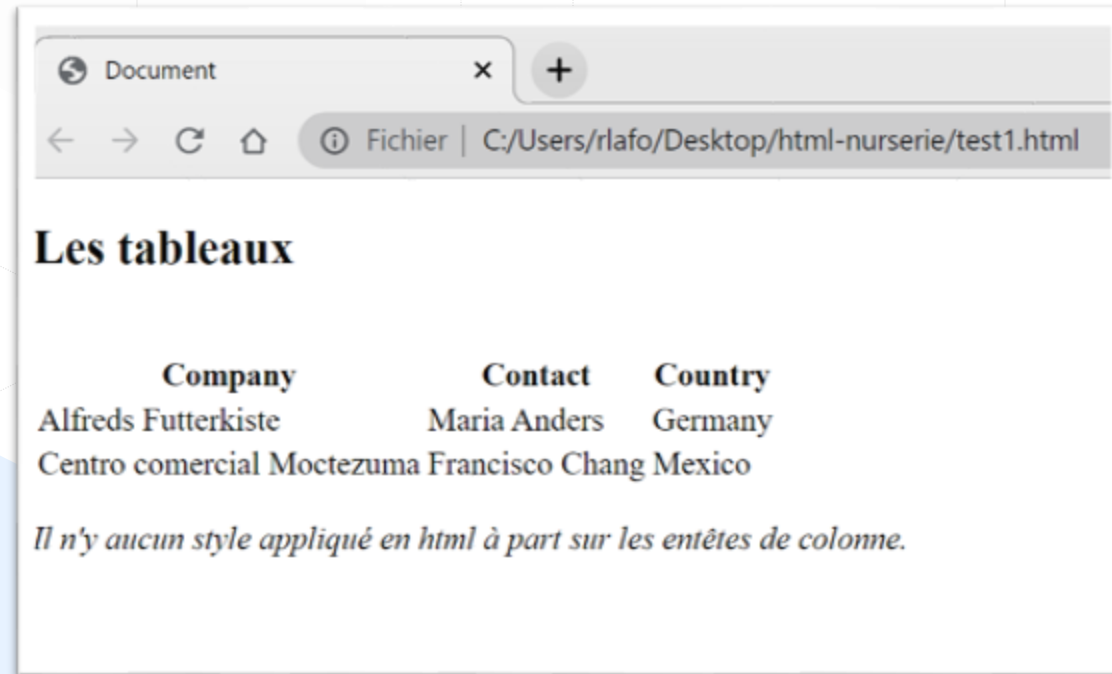
**tr** : une ligne

**th** : en-tête de colonne

**td** : une colonne

# Présentation du HTML

## LES TABLEAUX



# Présentation du HTML

## LES LISTES

### Non ordonnée

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

### Ordonnée

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

1. Coffee
2. Tea
3. Milk

### Listes imbriquées

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

## Présentation du HTML

### LES COMMENTAIRES

```
<!-- Write your comments here -->
```

Les commentaires ne sont pas affichés à l'écran mais sont visibles dans le html.  
Ils servent à donner des renseignements utiles sur la structure et permettent de rapidement comprendre le code des uns et des autres.

! Ne pas mettre « - - > » dans les commentaires

#### Raccourcis :

Sélectionner les lignes, ou placer le curseur sur la ligne à mettre en commentaire, puis :

Win => Ctrl + :

Mac => Cmd + Shift + :

# Présentation du HTML

## EXERCICE PRATIQUE

Recréer cette page en HTML

Sera fourni un dossier avec :

- 1 fichier image du modèle
- 1 fichier image à insérer
- 1 dossier « css »
- 1 fichier test2.html pour débiter

### Mon site web

Lorem ipsum dolor sit amet. Qui eaque voluptas hic voluptas veritatis aut eveniet quia? Eos aperiam repudiandae et pariatur beatae aut assumenda pariatur est autem quod ut vitae ipsum eum culpa molestiae eum modi impedit? Sed asperiores suscipit a quam error in temporibus dolore.

33 nesciunt velit et voluptas aliquid non voluptas voluptas et cumque tempora. Ea impedit unde rem repellat atque rem accusamus blanditiis. Aut libero laborum est explicabo aperiam nam error facere et cupiditate omnis sit Quis excepturi est illum galisum ut voluptatum suscipit.



- Développement Front-End
- Développement Back-End
- Optimisation
- Référencement

Matière	Note	Note Max	Note Min
Front	16	18	14
Back	15	17	12
Design	14	19	10

## Présentation du HTML

### LES FORMULAIRES

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.

La balise **<form>** est utilisée pour créer un formulaire HTML pour la saisie de l'utilisateur

L'attribut **action** définit l'action à effectuer lors de la soumission du formulaire.

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 de post-transaction 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>
```

Ici, nous ne verrons pas la partie **traitement**, donc on ne précisera pas l'attribut **action**.

Par contre, **methode="get"** nous permettra de visualiser l'envoi des données du formulaire.



## Présentation du HTML

### LES FORMULAIRES

L'élément **<form>** en HTML peut contenir un ou plusieurs des éléments de formulaire suivants :

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

Pour en savoir plus :

<https://developer.mozilla.org/fr/docs/Web/HTML/Element#formulaires>

# Présentation du HTML

## LES FORMULAIRES

Quelques-uns des différents types d'**input** :

```
<input type="text">
```

Affiche un champ de saisie de texte sur une seule ligne

```
<input type="number">
```

Affiche un champ qui n'accepte que des chiffres

```
<input type="radio">
```

Affiche un bouton radio (pour sélectionner l'un des nombreux choix)

```
<input type="checkbox">
```

Affiche une case à cocher (pour sélectionner zéro ou plusieurs choix)

```
<input type="submit">
```

Affiche un bouton de soumission (pour soumettre le formulaire)

```
<input type="button">
```

Affiche un bouton cliquable

# Présentation du HTML

## LES FORMULAIRES

Liste complète d'**input** :

- `<input type="text">`
- `<input type="email">`
- `<input type="tel">`
- `<input type="password">`
- `<input type="checkbox">`
- `<input type="radio">`
- `<input type="number">`
- `<input type="range">`
- `<input type="url">`
- `<input type="file">`
- `<input type="image">`

- `<input type="color">`
- `<input type="search">`
- `<input type="time">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="month">`
- `<input type="week">`
- `<input type="button">`
- `<input type="submit">`
- `<input type="reset">`
- `<input type="hidden">`

Page listant les différents types d'input, et leurs attributs

<https://developer.mozilla.org/fr/docs/Web/HTML/Element/input>

# Présentation du HTML

## LES FORMULAIRES

L'élément **<select>** :

```
<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>
```

Les éléments **<option>** définissent une option qui peut être sélectionnée.

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

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

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

# Présentation du HTML

## LES FORMULAIRES

Utilisez l'attribut **size** pour spécifier le nombre de valeurs visibles.

Utilisez l'attribut **multiple** pour permettre à l'utilisateur de sélectionner plusieurs valeurs.

(Il conviendra de prévenir l'utilisateur que la sélection **multiple** est activée, et qu'il lui suffit d'appuyer sur la touche **Ctrl** en plus du click)

```
<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>
```

# Présentation du HTML

## LES FORMULAIRES

L'élément **<textarea>** :

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

```
<textarea name="message" rows="10" cols="30">
```

The cat was playing in the garden.

```
</textarea>
```

L'attribut **rows** spécifie le nombre de lignes visibles dans une zone de texte.

L'attribut **cols** spécifie la largeur visible d'une zone de texte.

# Présentation du HTML

## LES FORMULAIRES

```
<form method="get">
  <label for="fname">First name :<br>
    <input type="text" name="fname" value="John">
  </label><br>
  <label for="html">
    <input type="radio" name="fav_language" value="html">HTML
  </label><br>
  <label for="css">
    <input type="radio" name="fav_language" value="css">CSS
  </label><br>
  <label for="javascript">
    <input type="radio" name="fav_language" value="javascript">JavaScript
  </label><br>
  <label for="vehicle1">
    <input type="checkbox" name="vehicle1" value="bike">I have a bike
  </label><br>
  <label for="vehicle2">
    <input type="checkbox" name="vehicle2" value="car">I have a car
  </label><br>
  <label for="vehicle3">
    <input type="checkbox" name="vehicle3" value="boat">I have a boat
  </label><br>
  <input type="submit" value="submit">
  <input type="reset" value="reset">
</form>
```

First name:

☐ HTML  
☐ CSS  
☐ JavaScript  
☐ I have a bike  
☐ I have a car  
☐ I have a boat

# Présentation du HTML

## LES FORMULAIRES : EXERCICE

Recréer cette page en HTML

Sera fourni :

- 1 fichier image du modèle
- 1 fichier formulaire.html pour débiter

### Balises utilisées :

```
<label for="">Titre du champ de formulaire</label>
<input type="text" placeholder="exemple de contenu">
<input type="tel">
<input type="email">
<select><option value="">item</option></select>
<input type="date">
<input type="number">
<input type="checkbox">
<textarea>,,,</textarea>
<input type="radio">
<button type="submit" name="submit">Envoyer ma
demande</button>
```

**Votre réservation d'hotel**

\* champs obligatoires

Votre nom\*:  
nom de famille

Votre prénom\*:  
prénom

Votre numéro de téléphone\*:  
0604020103

Adresse e-mail\*:

Sélectionner la ville :  
Paris

Date d'arrivée:  
jj/mm/aaaa

Nombre de personnes\*:

Vos options:  
Climatisation ☐ Mini Bar ☐ Jacuzzi ☐

Ajouter un commentaire.

Acceptation des Conditions Générales de Vente (CGV)\*  
☒ J'accepte ☐ Je refuse

Envoyer ma demande

Selectionner la ville :  
Lille  
Lille  
Paris  
Bordeaux  
Marseille  
Nantes  
Lyon  
La Rochelle  
Poitiers  
Toulouse



# Présentation du HTML

## LES ANCRES (OU SIGNETS)

Une **ancre** est un lien qui renvoie vers un élément de la page.

Il faut donner un identifiant unique à l'élément cible (un id) que l'on place en tant qu'attribut dans la balise de cet élément cible.

### Exemple :

*Lien* -> `<a href="#p1">Paragraphe 1</a>` Pour sélectionner un **id** il faut ajouter **#** devant la valeur

*Cible* -> `<p id="p1"> Lorem ipsum... </p>` La valeur d'un **id** ne doit pas contenir d'espace  
On peut ajouter un **id** à n'importe quelle balise html

# Présentation du HTML

## LES ANCRES (OU SIGNETS)

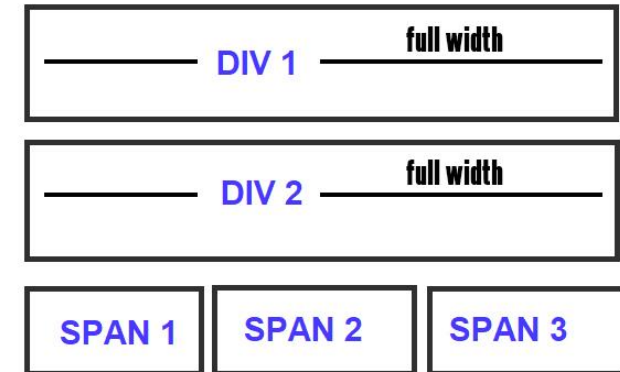
Balises utilisées lorsqu'il n'y a pas d'autre balise qui correspond au besoin.

**<div></div>**

- élément contenant d'autres éléments, servant à définir un bloc
- servira essentiellement à faire du positionnement de bloc

**<span></span>**

- élément contenant d'autres éléments, inline
- servira essentiellement à regrouper des caractéristiques globales pour des éléments textuels



## Présentation du HTML

### HTML 5 LES NOUVEAUTÉS

Nouvelles balises (balises sémantiques ou d'organisation) : section, video, article, audio, aside, header, footer ...

De nouvelles balises audio et vidéo qui ne nécessitent plus l'appel à des plug-ins dédiés.

Le dessin 2D et 3D par la nouvelle balise <canvas>.

Une profusion de formulaires novateurs comme les curseurs ou les calendriers et la prise en charge de façon native par les navigateurs de la validation des données.

Les nouveaux éléments de formulaire : datetime, date, time, month, number...  
et bien d'autres fonctionnalités...

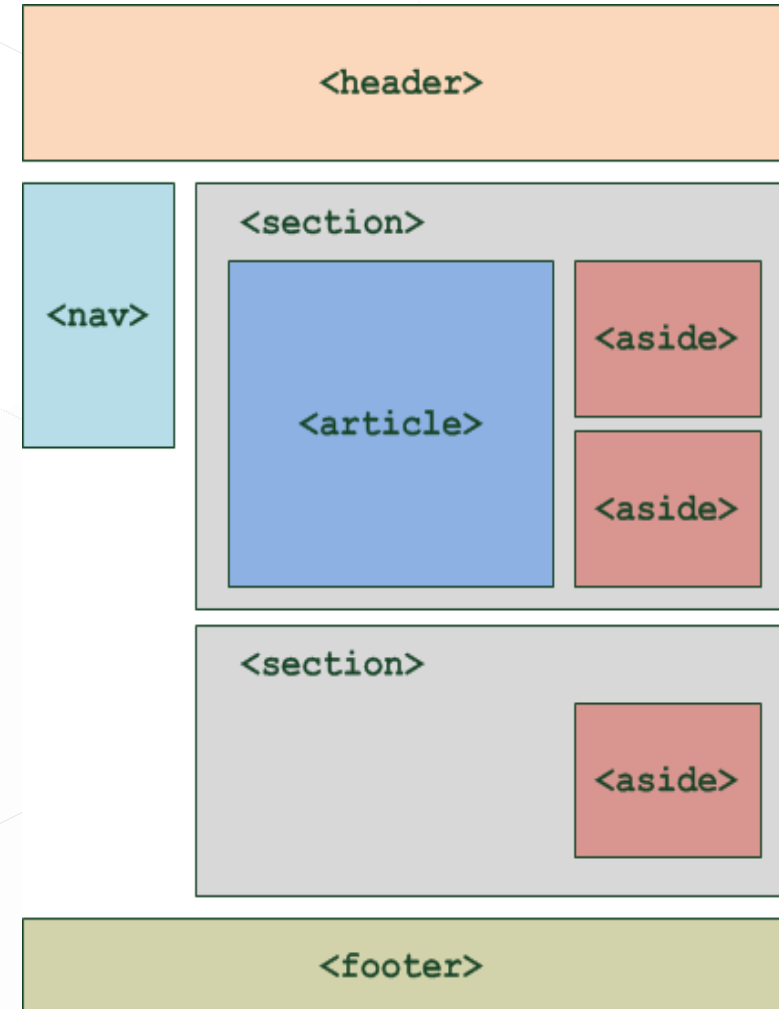
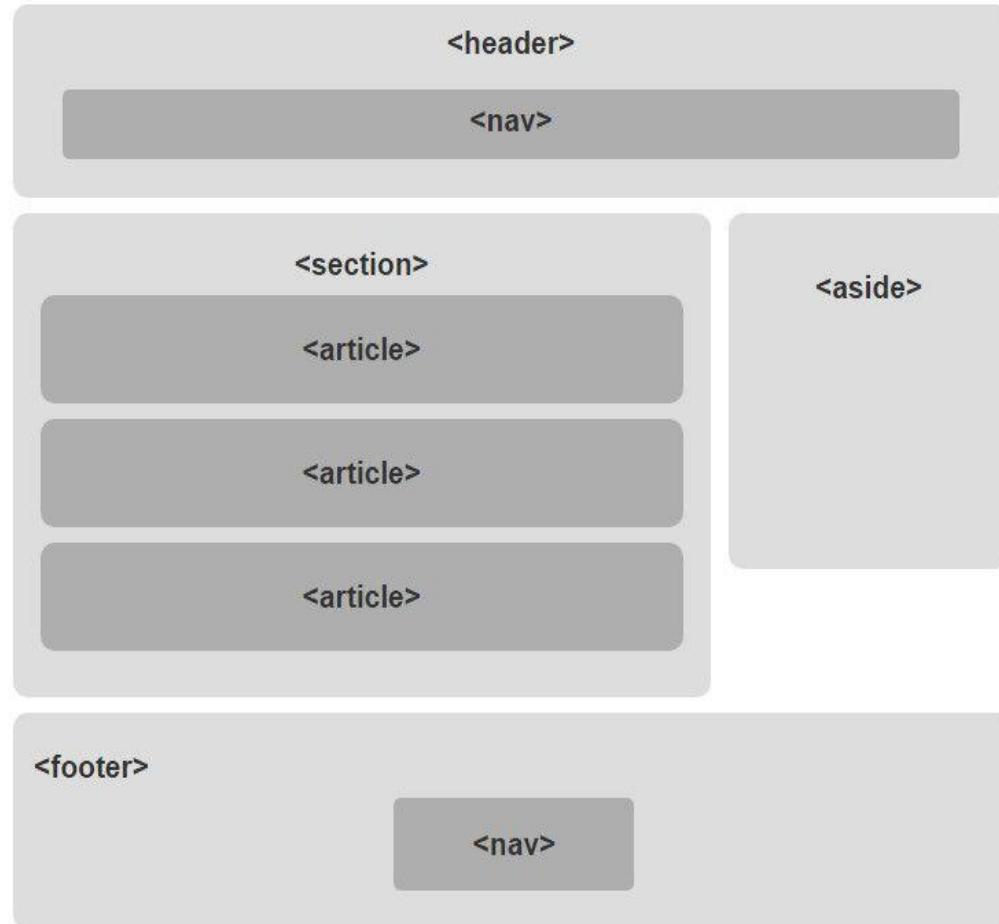
Liste complète des ajouts de HTML 5 :

<https://fr.wikipedia.org/wiki/HTML5>

<https://www.aurone.com/les-nouveautes-apportees-par-html-5>

# Présentation du HTML

## LES BALISES SÉMANTIQUES



# Présentation du HTML

## LES BALISES SÉMANTIQUES

### **<header>**

Section d'introduction d'un article, d'une autre section ou du document entier (en-tête de page).

### **<nav>**

Section possédant les liens de navigation principaux (au sein du document ou vers d'autres pages)

### **<section>**

Section générique regroupant un même sujet, une même fonctionnalité, de préférence avec un en-tête, ou bien une section d'application web

### **<article>**

Section de contenu indépendante, pouvant être extraite individuellement du document ou syndiquée (flux RSS ou équivalent), sans pénaliser sa compréhension

### **<aside>**

Section dont le contenu est un complément par rapport à ce qui l'entoure, qui n'est pas forcément en lien direct avec le contenu mais qui peut apporter des informations supplémentaires.

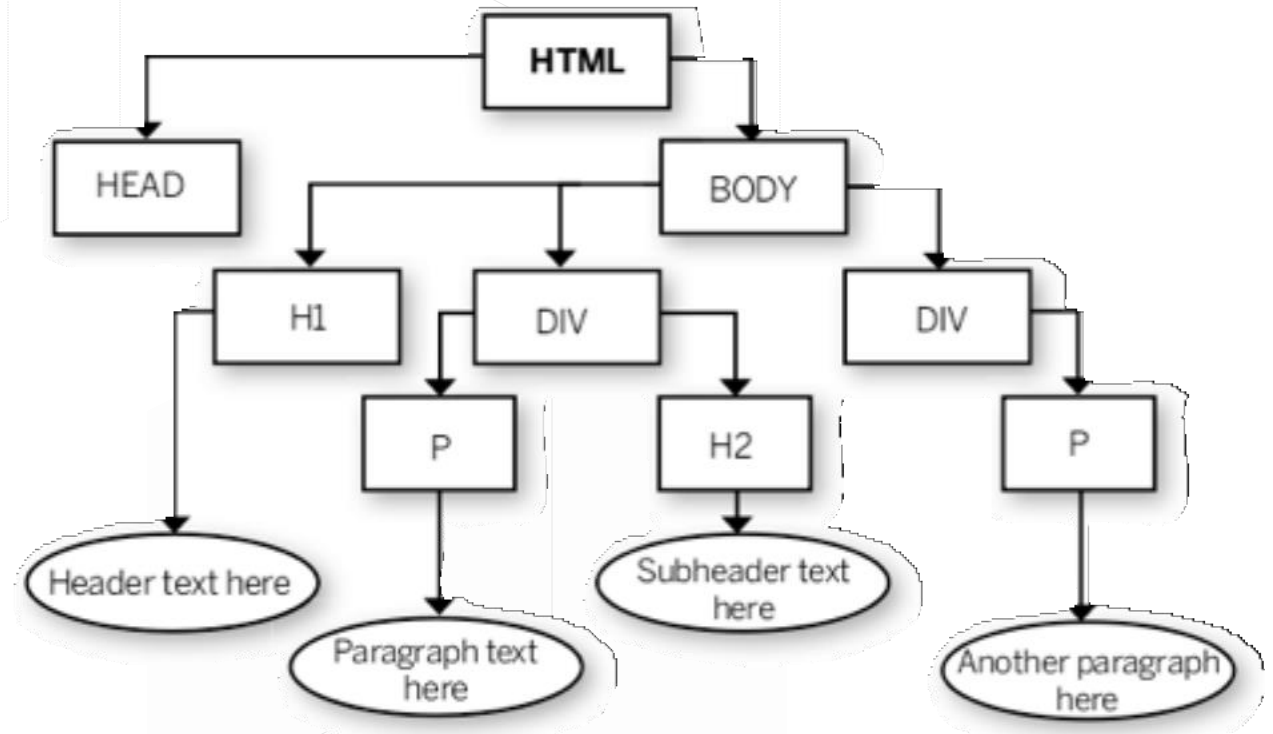
### **<footer>**

Section de conclusion d'une section ou d'un article, voire du document entier (pied de page).

# Présentation du HTML

## ARBORESCENCE HTML

- L'arborescence HTML contient une racine, des parents, des frères, des enfants
- comme un arbre généalogique
- Appellation technique : « **DOM** » (Document Object Model)



# Présentation du HTML

## EXERCICE CV

Créer votre CV en HTML (structure uniquement)

Étape 1 : Créez une page HTML avec une structure de base.

Étape 2 : Dans un `<header></header>`, ajoutez votre nom et votre prénom en titre principal et une image d'avatar.

Étape 3 : A la suite, dans une `<div></div>`, ajoutez 3 `<section></section>` avec leur propre titre secondaire (mon expérience, mes compétences, ma formation).

Étape 4 : Chaque expériences et formations pourra être placée dans un `<article></article>`.

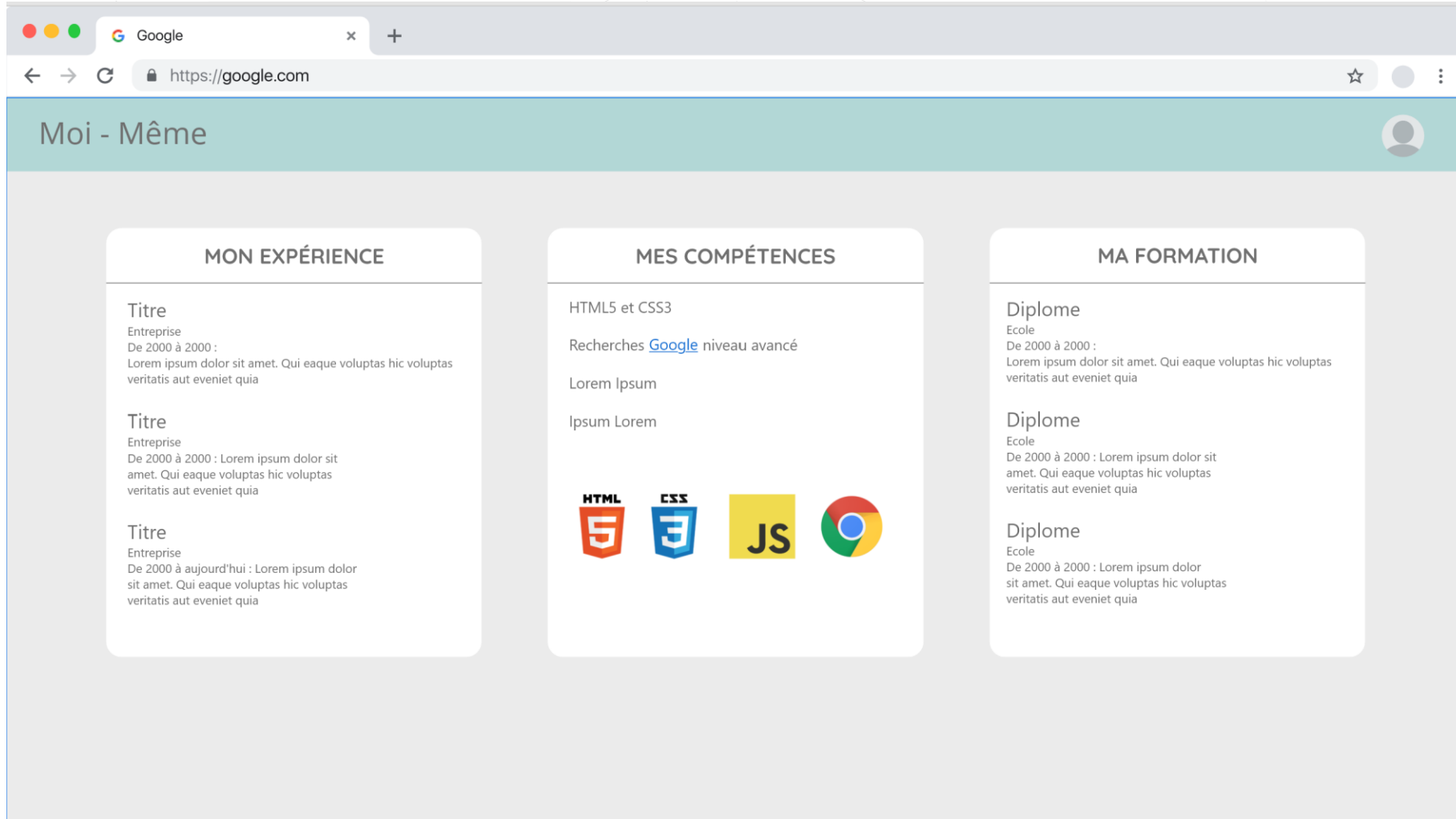
Etape 5 : Suivre les indications sur l'aperçu fourni





# Présentation du HTML

## Aperçu du cv une fois stylisé



## Découverte du CSS et application

Séparation de la structure logique (HTML) et de la présentation (CSS)

Présentation suivant une feuille de style (style sheet) qui traite les éléments de contenu en éléments de présentation

CSS = Cascading Style Sheets (feuilles de style en cascade)

En cascade :

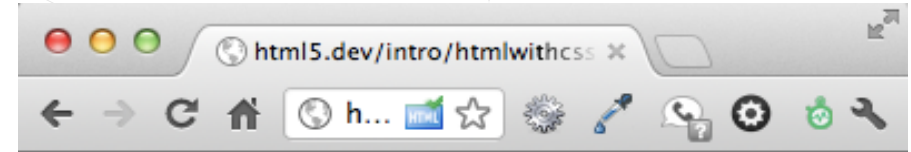
- on peut utiliser plusieurs feuilles de styles
- il y a un degré d'importance pour chaque feuille de style

## Découverte du CSS et application

```
h1{
  color: red;
}

h2{
  color: blue;
  font-style: italic;
}

p{
  color: white; background-
color: black;
}
```



### whichElement?

*Trying to answer that age old question:*

*Should that be a div, a span, or something else?*

- Home
- Contribute
- About

One of the main challenges we see in building semantic content is picking what tag to use when. This site seeks to help with that. Now, before we get all judgy and preachy let me get a few tenants out there:

## Découverte du CSS et application

### Ajouter des styles

Dans le head avec une balise `<style></style>` :

```
<head>
<style type="text/css">
    #boite1 {
        position: relative;
        margin-left: 20px;
        height: auto;
        width: 643px;
        float: left;
    }
</style>
</head>
```

Feuille de style externe (recommandé) :

```
<head>
    <link rel="stylesheet" href="css/st
yle.css" />
</head>
```

## Découverte du CSS et application

### Sélecteur CSS

**Syntaxe:**

```
Sélecteur {  
  Attribut-css : valeur ;  
}
```

**Selecteur de type (nom des balise):**

```
p {  
  text-align: center;  
  color: red;  
}
```

## Découverte du CSS et application

### Le sélecteur d' identifiant CSS

Dans le html :

```
<p id="para1"> para1 </p>
```

Dans le css :

```
#para1 {  
    text-align: center;  
}
```

## Découverte du CSS et application

### Le sélecteur de classe CSS

Dans le html :

```
<p class="bg-blue center "> para1 </p>
```

Dans le css :

```
.center {
  text-align: center;
  color: red;
}
.bg-blue {
  background-color:blue;
}
```

L'attribut class, peut supporter plusieurs valeurs:

```
<p class="center large">This paragraph refers to two classes.</p>
```

```
.center {
  text-align: center;
}
.large {
  width :100% ;
}
```

Il est également possible de sélectionner tous les éléments d'un type contenant une certaine class:

```
p.bg-blue {
  Background-color: blue;
}
```

## Découverte du CSS et application

### Le sélecteur de regroupement CSS

Le selecteur de regroupement,  
espacé par une virgule

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

Sélectionne tous les **h1**, tous les **h2**, et tous les **p**.

Le selecteur d' "enfant",  
juste séparé par un espace

```
div p {
  text-align: center;
  color: red;
}
```

Sélectionne tous les **p** enfant d'une **div**

Mini-jeu sur les selecteurs CSS  
css diner: <https://flukeout.github.io/>

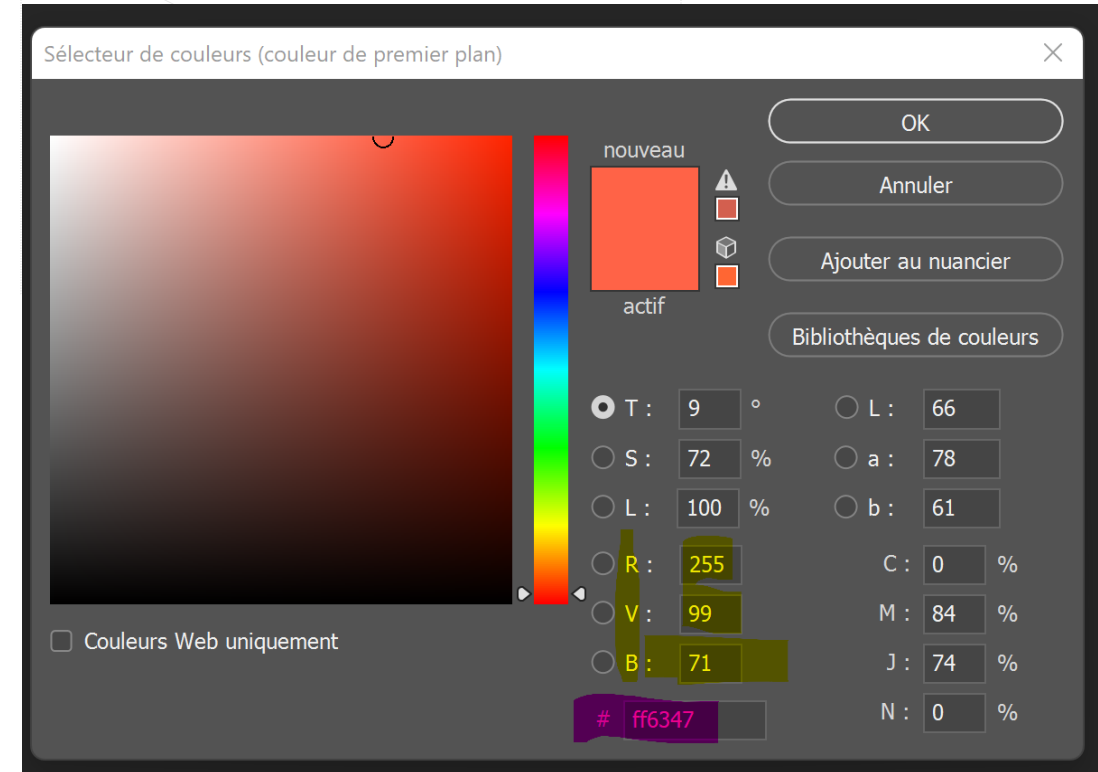


# Découverte du CSS et application

## Les couleurs

```
p {
  color: rgb(255, 99, 71);
  color: #ff6347; (hexadecimal)
  color: rgba(255, 99, 71, 0.5);
}
```

Couche Alpha = transparence  
De 0 (objet invisible) à 1 (100% opaque),  
Exemple 0.5 = 50% d'opacité



## Découverte du CSS et application

### Couleurs d'arrière-plan

```
body {  
    background-color: lightblue;  
    background: rgba(0, 128, 0, 0.3);  
}
```

## Découverte du CSS et application

### Image d'arrière-plan

```
body {  
  Background-image: url("images/paper.gif");  
}
```

## Découverte du CSS et application

### Répétition d'arrière-plan

```
body {  
    background-image: url("images/gradient_bg.png");  
    background-repeat: repeat-x;  
    background-repeat: repeat-y;  
    background-repeat: no-repeat;  
}
```

## Découverte du CSS et application

### Défilement d'arrière-plan

```
body {  
    background-attachment: fixed;  
}
```

## Découverte du CSS et application

### Le modèle des boîtes

Le modèle des boîtes CSS se base sur le principe que tous les éléments utilisés en HTML sont mis en forme sur la base de zones rectangulaires. Ainsi chaque élément lorsqu'il est rendu dans le navigateur aura les contours suivants :

- Le contenu de l'élément dont la taille est définie par sa largeur width et sa hauteur height
- Son espacement intérieur : propriété padding
- Sa marge : propriété margin
- Sa bordure : propriété border

## Découverte du CSS et application

### Les bordures

```
div {  
    border: 2px solid gray;  
    border: 5px dotted #ff0000;  
}
```

2px : taille de la bordure

solid : style (normal, pointillés, tirets, double...)

gray : couleur

# Découverte du CSS et application

## Les bordures arrondies

```
div {  
  
border-radius: 5px; /* les coins seront tous arrondis */  
  
border-radius: 5px 10px 15px 20px; /* valeurs différentes pour chaque coin en partant du haut  
gauche puis      dans le sens des aiguilles d'une montre */  
}
```

**NB :** pour créer un cercle, on peut mettre :

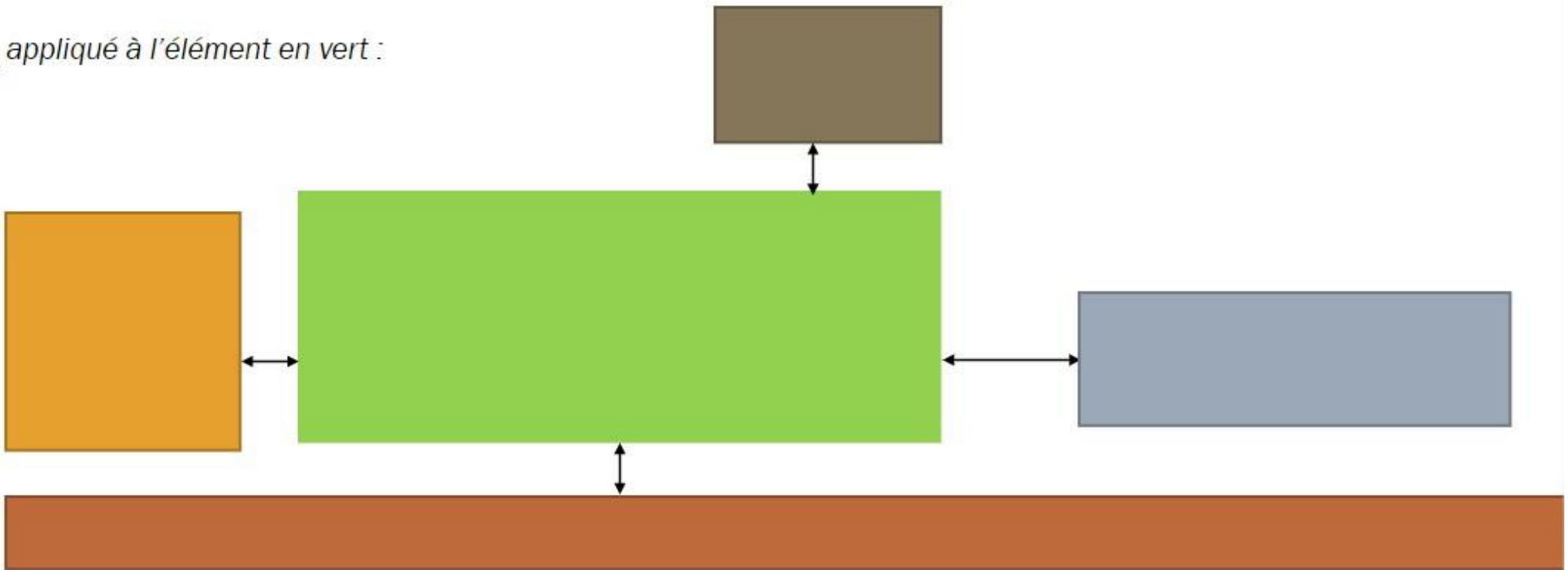
```
div {border-radius: 50%;}
```



# Découverte du CSS et application

## Les marges

margin appliqué à l'élément en vert :



# Découverte du CSS et application

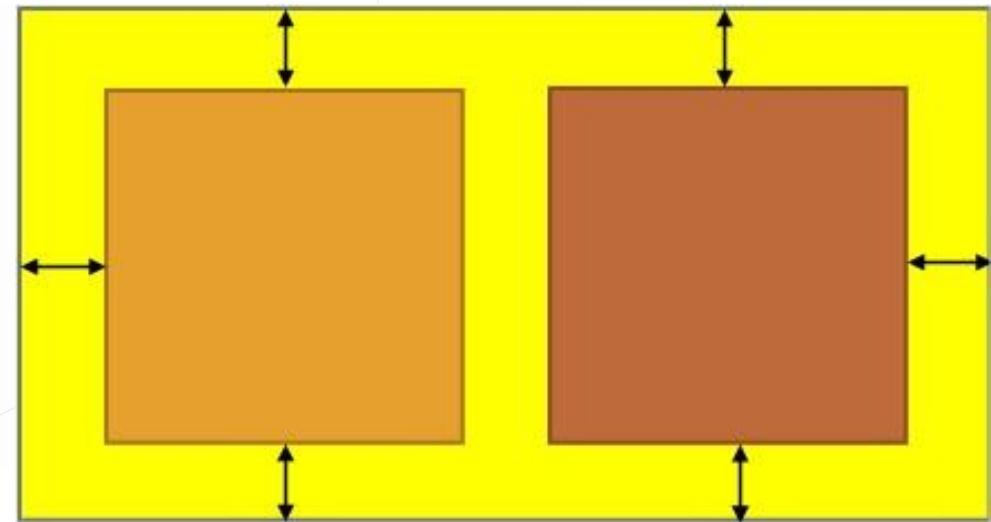
## Le padding

Les propriétés CSS padding, sont utilisées pour générer de l'espace autour du contenu d'un élément, à l'intérieur de toute bordure définie.

Même syntaxe que les marges :

padding-top  
padding-right  
padding-bottom  
padding-left  
padding

*Padding appliqué à l'élément en jaune, ses enfants ne sont donc pas collés aux bords*



## Découverte du CSS et application

### La hauteur

```
div {  
  height: 200px;  
  max-height: 400px;  
  min-height: 50%;  
}
```

## Découverte du CSS et application

### La largeur

```
div {  
    width: 100%;  
    max-width: 400px;  
    min-width: 100px;  
}
```

## Découverte du CSS et application

### Décoration du texte

```
a {  
  text-decoration: overline;  
  text-decoration: line-through;  
  text-decoration: underline;  
  text-decoration: none; /* utile pour supprimer le style des  
    liens par défaut */  
}
```

Lien par défaut : [Google](#)

Et en utilisant la valeur "none" : [Google](#)

## Découverte du CSS et application

### Aligner les textes

```
p {  
    text-align: left; /* alignement à gauche (c'est la valeur par  
défaut) */  
}  
  
p {  
    text-align: center; /* texte centré dans son conteneur */  
}  
  
p {  
    text-align: right; /* alignement à droite */  
}  
  
p {  
    text-align: justify; /* le texte est justifié et prend toute la  
largeur du conteneur */  
}
```

## Découverte du CSS et application

### Les polices de caractères (fonts)

#### Police de texte

```
p {
    font-family: Arial, Helvetica, sans-serif; /* choix 1, choix 2,
catégorie par défaut */
    font-style: italic;
    font-style: oblique;
    font-weight: bold;
}
```

#### Utiliser les polices Google

Dans le Head du HTML

```
: <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia" >
```

En CSS : `h2 {font-family: 'Sofia', cursive; }`

-> <https://fonts.google.com/>

## Découverte du CSS et application

### Les liens

```
/* unvisited link */  
a:link {  
  color: red;  
}
```

```
/* visited link */  
a:visited {  
  color: green;  
}
```

```
/* mouse over link */  
a:hover {  
  color: hotpink;  
}
```

```
/* selected link */  
a:active {  
  color: blue;  
}
```



# Découverte du CSS et application

## Le display : rappel

Exemples :

<div>

<P>

<H1,2,3...>

### Block

Prennent toute la largeur disponible

Provoque un retour à la ligne

Peuvent contenir des éléments inline

Exemples :

<img>

<a>

<span>

<button>

<input>

### Inline

Ne provoque pas de retour à la ligne

Ne peuvent pas contenir des éléments block

Peuvent contenir des éléments inline

## BLOCK-LEVEL ELEMENTS:



## INLINE ELEMENTS:



## Découverte du CSS et application

### Le display

```
div {  
  display: inline;  
  display: block;  
  display: inline-block;  
  display: none;  
  visibility: hidden;  
}
```

- **NB :** Masquer un élément peut être fait en définissant la propriété `display` sur `none`. L'élément sera masqué et la page s'affichera comme si l'élément n'existait pas.
- Par contre avec `visibility :hidden`, l'élément occupera toujours le même espace qu'auparavant. L'élément sera masqué, mais affectera toujours la mise en page.

## Découverte du CSS et application

### Le positionnement

```
div {  
  position: static; /* valeur par défaut */  
  position: relative;  
  position: fixed;  
  position: absolute;  
  position: sticky;  
  
  bottom: 0; /* ou top: ; */  
  right: 0; /* ou left: ; */  
}
```

Position : absolute  
top : 0; left : 0;

Position : relative

## Découverte du CSS et application

### Overflow

```
div {
  overflow: visible;
  overflow: hidden;
  overflow: scroll;
  overflow: auto;
  overflow-x: hidden; /* Hide horizontal scrollbar */
  overflow-y: scroll; /* Add vertical scrollbar */
}
```

Lorem ipsum dolor, sit amet consectetur adipisicing elit.  
 Consequuntur, sunt. Eligendi atque fugiat nemo odit!  
 Voluptas, adipisci exercitationem veniam ullam aut architecto  
 nam consequuntur blanditiis qui ad maxime molestias facilis?  
 Lorem ipsum dolor, sit amet consectetur adipisicing elit.  
 Consequuntur, sunt. Eligendi atque fugiat nemo odit!  
 Voluptas, adipisci exercitationem veniam ullam aut architecto  
 nam consequuntur blanditiis qui ad maxime molestias facilis?

## Découverte du CSS et application

### Affichage flottant

La propriété CSS float spécifie comment un élément doit flotter.

```
div {  
    float: right;  
  
    float: left;  
  
    float: none;  
  
}
```

Float : left

Float : right

**Arrêter l'affichage flottant :** `div { clear: both; }`

## Découverte du CSS et application

### Pseudo élément

Un pseudo-élément CSS est utilisé pour styliser des parties spécifiées d'un élément. Par exemple, il peut être utilisé pour :

1. Styliser la première lettre, ou ligne, d'un élément
2. Insérer du contenu avant ou après le contenu d'un élément

```
selector::pseudo-element {  
  property: value;  
}
```

Exemple : `p::first-line {color: red; text-transform: uppercase;}`

-> La première ligne de chaque paragraphe sera en rouge et en majuscules.

Liste complete des pseudo-éléments : [https://www.w3schools.com/css/css\\_pseudo\\_elements.asp](https://www.w3schools.com/css/css_pseudo_elements.asp)

## Découverte du CSS et application

### L'opacité

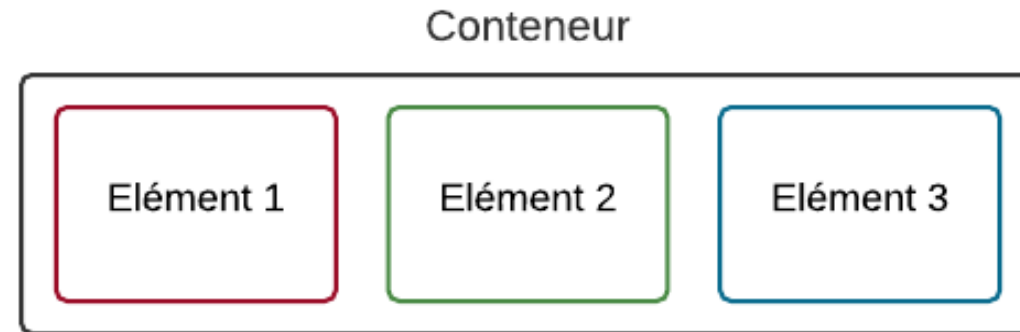
```
img {  
  opacity: 0.5; /* opacité de 50% */  
}
```

```
Img:hover {  
  opacity: 1.0; /* opacité de 100% */  
}
```

## Découverte du CSS et application

### Flexbox : agencement du contenu

Le principe de la mise en page avec Flexbox est simple : vous définissez un conteneur, et à l'intérieur vous y placez plusieurs éléments :



Le conteneur est une balise HTML, et les éléments sont d'autres balises HTML à l'intérieur :

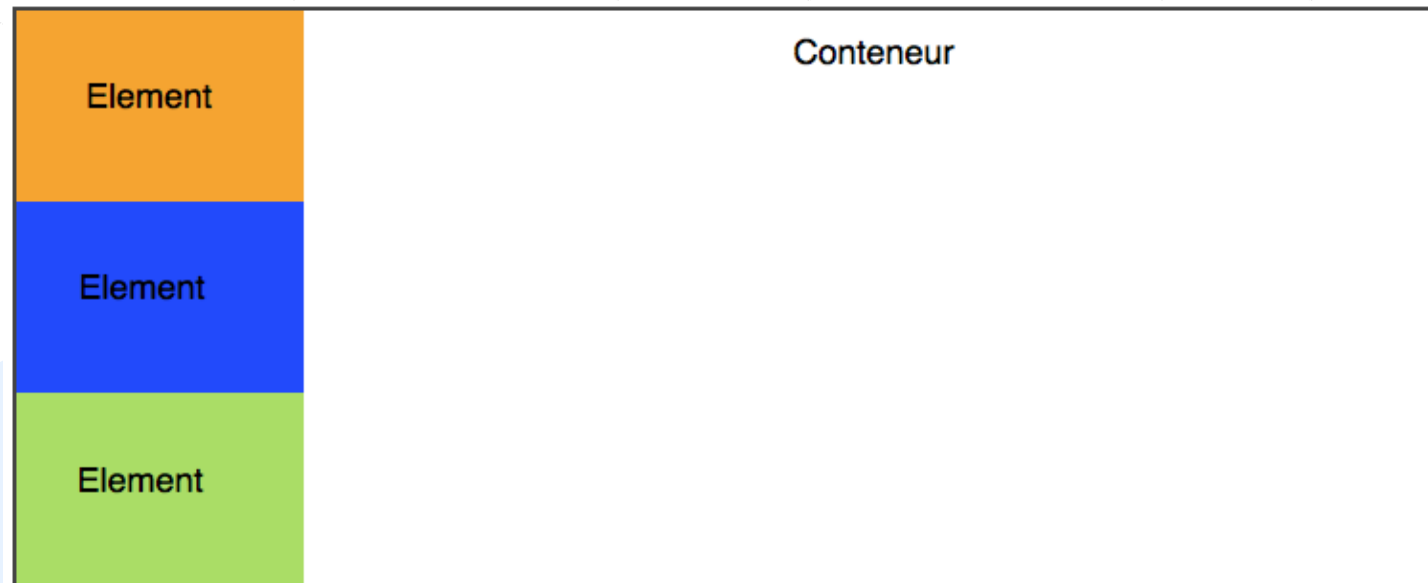
```
<div id="conteneur">  
  <div class="element e1">Element 1</div>  
  <div class="element e2">Element 2</div>  
  <div class="element e3">Element 3</div>  
</div>
```



## Découverte du CSS et application

### Flexbox : agencement du contenu

Les éléments ici sont de type « block » (div) ils vont donc se placer par défaut les uns en dessous des autres :



## Découverte du CSS et application

### Flexbox : agencement du contenu

En ajoutant à notre conteneur la propriété « flex », la disposition va changer :

```
#conteneur  
{  
  display: flex;  
}
```



## Découverte du CSS et application

### Flexbox : colonne ou ligne

#### Modifier la direction

Flexbox nous permet d'agencer ces éléments dans le sens que l'on veut.

Avec **flex-direction**, on peut les positionner verticalement ou encore les inverser. Il peut prendre les valeurs suivantes :

- row : organisés sur une ligne (par défaut) ;
- column : organisés sur une colonne ;
- row-reverse : organisés sur une ligne, mais en ordre inversé ;
- column-reverse : organisés sur une colonne, mais en ordre inversé.

## Découverte du CSS et application

### Flexbox : colonne ou ligne

Exemple :

```
#conteneur  
{  
  display: flex;  
  flex-direction: column-reverse;  
}
```

Les éléments sont disposés en colonne mais l'ordre a été inversé sans modifier le fichier html :



## Découverte du CSS et application

### Flexbox : retour à la ligne

#### Gérer le retour à la ligne

Flexbox nous permet d'agencer ces éléments dans le sens que l'on veut.

Par défaut, les blocs essaient de rester sur la même ligne s'ils n'ont pas la place (ce qui peut provoquer des bugs de design, parfois). Si vous voulez, vous pouvez demander à ce que les blocs aillent à la ligne lorsqu'ils n'ont plus la place, avec **flex-wrap** qui peut prendre ces valeurs :

- nowrap : pas de retour à la ligne (par défaut) ;
- wrap : les éléments vont à la ligne lorsqu'il n'y a plus la place ;
- wrap-reverse : les éléments vont à la ligne, lorsqu'il n'y a plus la place, en sens inverse.

# Découverte du CSS et application

## Flexbox : retour à la ligne

```
#conteneur
{
  display: flex;
  flex-wrap: wrap;
}
```

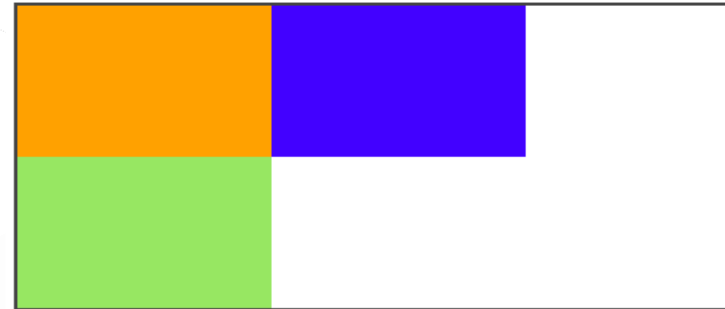
Voici l'effet que prennent les différentes valeurs sur une même illustration :

nowrap



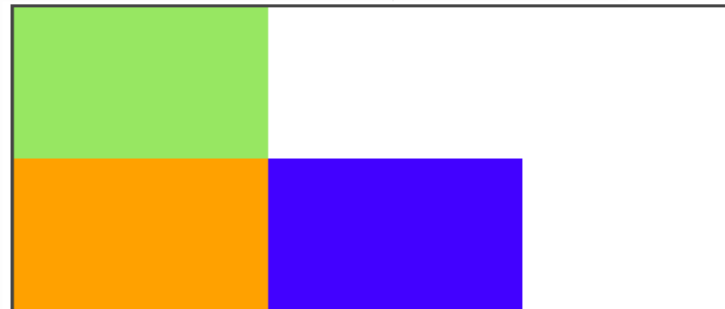
Les éléments se resserrent tant qu'ils peuvent

wrap



Les éléments passent à la ligne

wrap-reverse



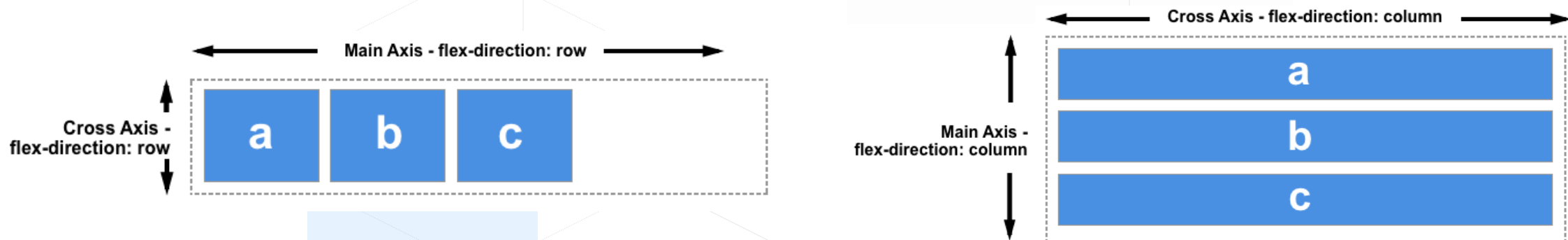
Les éléments passent à la ligne à l'envers

## Découverte du CSS et application

### Flexbox : les alignements

Les éléments sont organisés soit horizontalement (par défaut), soit verticalement. Cela définit ce qu'on appelle **l'axe principale**. Il y a aussi un axe secondaire (*cross axis*) :

- si vos éléments sont organisés horizontalement, l'axe secondaire est l'axe vertical ;
- si vos éléments sont organisés verticalement, l'axe secondaire est l'axe horizontal.



## Découverte du CSS et application

### Flexbox : aligner sur le Main Axis

Pour changer l'alignement des éléments disposés horizontalement (disposition flex par défaut), on va utiliser « **justify-content** », qui peut prendre ces valeurs :

- **flex-start** : alignés au début (par défaut) ;
- **flex-end** : alignés à la fin ;
- **center** : alignés au centre ;
- **space-between** : les éléments sont étirés sur tout l'axe (il y a de l'espace entre eux) ;
- **space-around** : idem, les éléments sont étirés sur tout l'axe, mais ils laissent aussi de l'espace sur les extrémités.



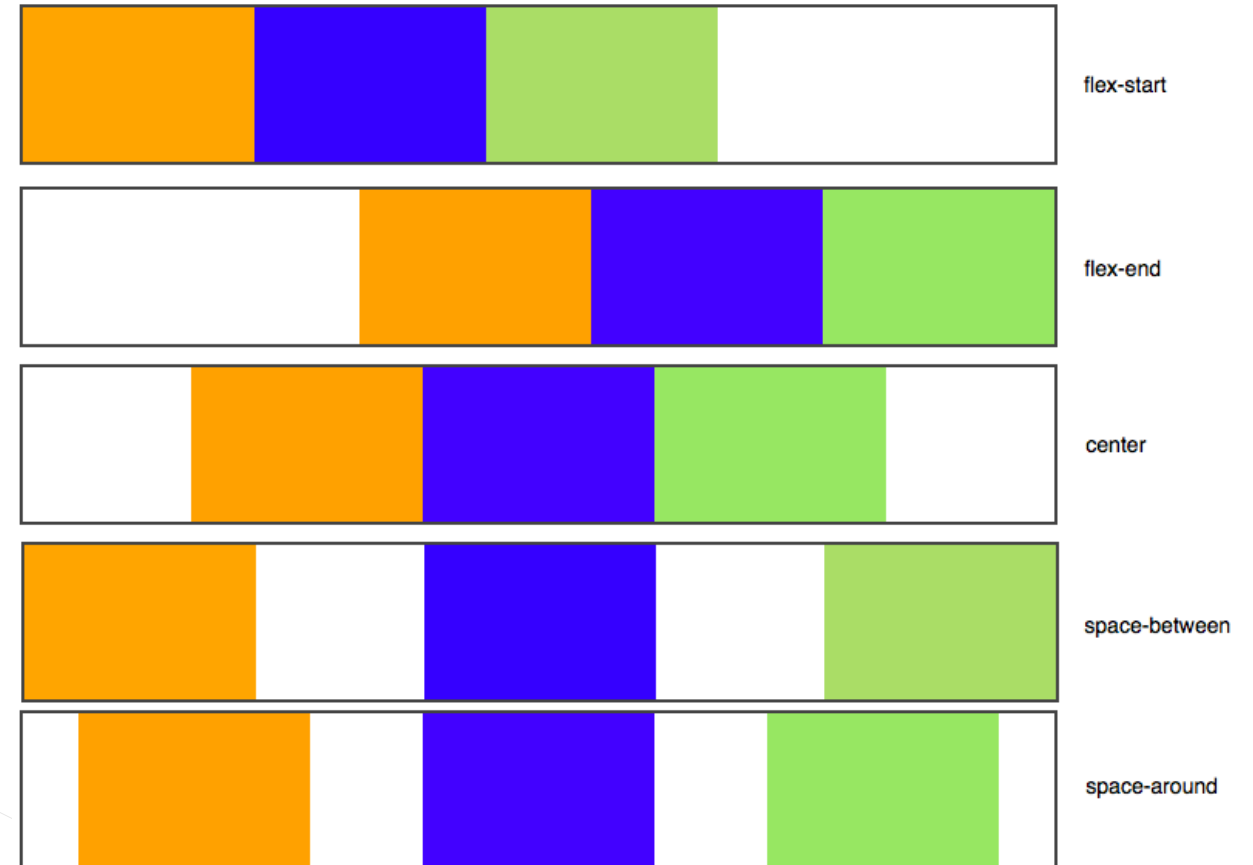
# Découverte du CSS et application

## Flexbox : aligner sur le Main Axis

Exemple :

```
#conteneur
{
  display: flex;
  justify-content: space-around;
}
```

Voici toutes les valeurs possibles :



## Découverte du CSS et application

### Flexbox : aligner sur le Cross Axis

Avec **align-items**, nous pouvons changer leur alignement sur l'**axe secondaire**.

Il peut prendre ces valeurs :

- **stretch** : les éléments sont étirés sur tout l'axe (valeur par défaut) ;
- **flex-start** : alignés au début ;
- **flex-end** : alignés à la fin ;
- **center** : alignés au centre ;
- **baseline** : alignés sur la ligne de base (semblable à flex-start).

## Découverte du CSS et application

### Flexbox : aligner sur le Cross Axis

Pour ces exemples, nous allons partir du principe que nos éléments sont dans une direction horizontale :

```
#conteneur
{
  display: flex;
  justify-content: center;
  align-items: center;
}
```



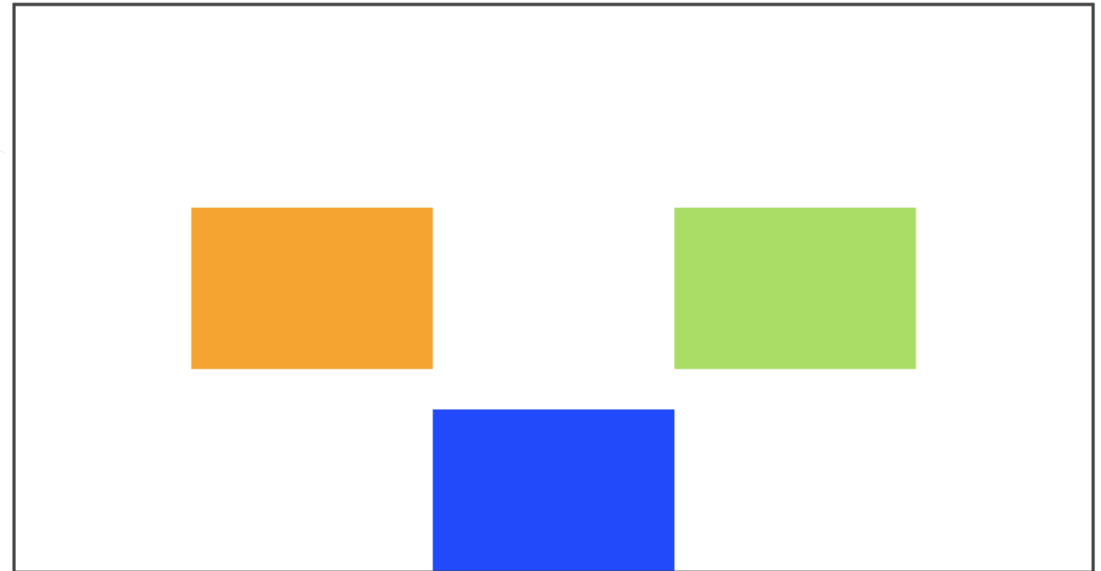
*Un alignement sur l'axe secondaire avec align-items nous permet de centrer complètement l'élément dans le conteneur.*

# Découverte du CSS et application

## Flexbox : alignement sélectif

Aligner un seul élément sur l'axe secondaire

```
#conteneur {
  display: flex;
  flex-direction: row;
  justify-content: center;
  align-items: center;
}
.element:nth-child(2) /* On sélectionne le deuxième bloc élément */
{
  background-color: blue;
  align-self: flex-end; /* Seul ce bloc sera aligné à la fin */
}
```



*Un élément aligné différemment des autres avec align-self.*

## Découverte du CSS et application

### Flexbox : l'ordre des éléments

Sans changer le code HTML, nous pouvons modifier l'ordre des éléments un par un en CSS grâce à la propriété **order**. Indiquez simplement un nombre, et les éléments seront triés du plus petit au plus grand nombre.

Reprenons une simple ligne de 3 éléments :

```
#conteneur  
{  
  display: flex;  
}
```



Une ligne de 3 éléments

# Découverte du CSS et application

## Flexbox : l'ordre des éléments

```
.element:nth-child(1) /* selection du 1er enfant du conteneur
appelé element */
{
  order: 3;
}
.element:nth-child(2) /* selection du 2ème enfant du
conteneur appelé element */
{
  order: 1;
}
.element:nth-child(3) /* selection du 3ème enfant du
conteneur appelé element */
{
  order: 2;
}
```

NB : order est différent de flex-direction : wrap-reverse et flex-direction : column-reverse car il permet de modifier le rang de chaque élément individuellement, ce n'est donc pas une simple inversion de l'ordre initial.

*Avant*



*Après*



Avec order, nous pouvons réordonner les éléments en CSS

## Découverte du CSS et application

### Flexbox : faire grossir ou maigrir

Avec la propriété flex , nous pouvons permettre à un élément de grossir pour occuper tout l'espace restant.

```
.element:nth-child(2) /* sélection du 2ème enfant du conteneur */  
{  
  flex: 1;  
}
```



Le deuxième élément s'étire pour prendre tout l'espace

## Découverte du CSS et application

### Flexbox : faire grossir ou maigrir

Le nombre que vous indiquez à la propriété flex indique dans quelle mesure il peut grossir par rapport aux autres.

```
.element:nth-child(1) /* sélection du 1er enfant de notre conteneur */  
{  
  flex: 2;  
}  
.element:nth-child(2) /* sélection du 2ème enfant de notre conteneur */  
{  
  flex: 1;  
}
```

Ici, le premier élément peut grossir 2 fois plus que le deuxième élément :



Le troisième élément s'adapte pour occuper l'espace restant



# Découverte du CSS et application

Pour aller plus loin

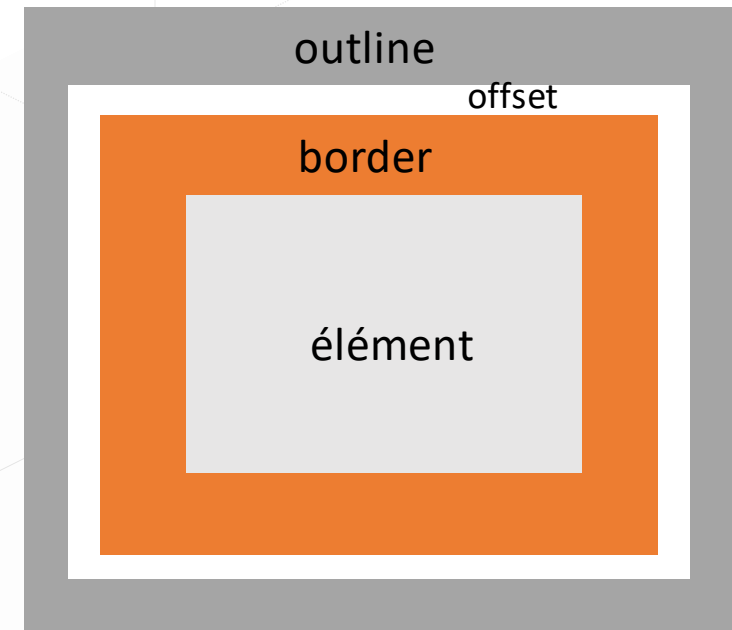


## Découverte du CSS et application

### Le contour

Un contour est une ligne qui est tracée autour des éléments, EN DEHORS des bordures, pour faire démarquer l'élément.

```
p {  
  
  outline: red solid 10px;  
  
  outline-offset: 35px; /* décalage du contour */  
  
}
```



# Découverte du CSS et application

## Espacement du texte

```
p {
text-indent: 50px;
/* création de padding sur
la 1ere ligne du paragraphe */
}
```

### L'espacement des lettres (dans le sens horizontal)

```
p {
letter-spacing: 3px;
letter-spacing: -3px; /* chevauchement des Lettres */
}
```

### Hauteur de la ligne

```
p {
line-height: 0.8;
}
```

### Espacement des mots

```
p {
word-spacing: 10px;
}
```

### Ombre de texte

```
p {
text-shadow: 2px 2px 5px red;
}
```

### Taille de la police

```
p {
font-size: 40px;
font-size: 2.5em;
font-size: 2.5rem; /* 40px/16=2.5em */
}
```



# Découverte du CSS et application

## Formatage des caractères

### Textes en MAJUSCULES

```
p {  
  text-transform: uppercase;  
}
```

### Majuscules En Début De Mot

```
p {  
  text-transform: capitalize;  
}
```

### Textes en miniscules

```
p {  
  text-transform: lowercase;  
}
```

### PETITES MAJUSCULES

```
p {  
  font-variant: small-caps;  
}
```



# Découverte du CSS et application

## Les polices de caractère (fonts)

### Utiliser une police téléchargée sur un site :

Placer d'abord les fichiers de police dans un dossier 'fonts' dans le dossier de votre projet.

Puis, en css :

En haut du fichier on met :

```
@font-face {  
  font-family: "Open Sans";  
  src: url("/fonts/OpenSans-Regular.ttf") format("truetype");  
}
```

Puis on attribue la police à un ou plusieurs éléments :

```
body {  
  font-family: 'Open Sans', sans-serif; /* en attribuant au body cette Font elle sera utilisée par défaut sur tous les textes. */  
}
```



# Découverte du CSS et application

## Les media queries

Les requêtes multimédias dans CSS3 ont étendu l'idée des types de médias CSS2 : au lieu de rechercher un type d'appareil, elles examinent la capacité de l'appareil.

Les requêtes multimédias peuvent être utilisées pour vérifier de nombreuses choses, telles que :

- largeur et hauteur de la fenêtre
- largeur et hauteur de l'appareil
- orientation (la tablette/le téléphone est-il en mode paysage ou portrait ?)
- résolution



# Découverte du CSS et application

## Les media queries

Les media queries, introduites dans CSS2, ont permis de définir différentes règles de style pour différents types de médias.

Exemples : vous pouvez avoir un ensemble de règles de style pour les écrans d'ordinateur, un pour les imprimantes, un pour les appareils portables, un pour les appareils de type télévision, et ainsi de suite.



# Découverte du CSS et application

## Les media queries

### Syntaxe de la media query

- Dans un fichier css

```
@media not|only mediatype and ( expressions) {  
  CSS-Code;  
}
```

- Dans un fichier HTML

```
<link rel="stylesheet" media=" mediatype and|not|only ( expressions)"  
      href=" print.css">
```





# Découverte du CSS et application

## Les media queries

```
@media screen and (min-width: 480px) {  
  body {  
    background-color: lightgreen;  
  }  
}
```

```
@media screen and (max-width: 900px) and (min-width: 600px) {  
  div.example {  
    font-size: 50px;  
    padding: 50px;  
    border: 8px solid black;  
    background: yellow;  
  }  
}
```

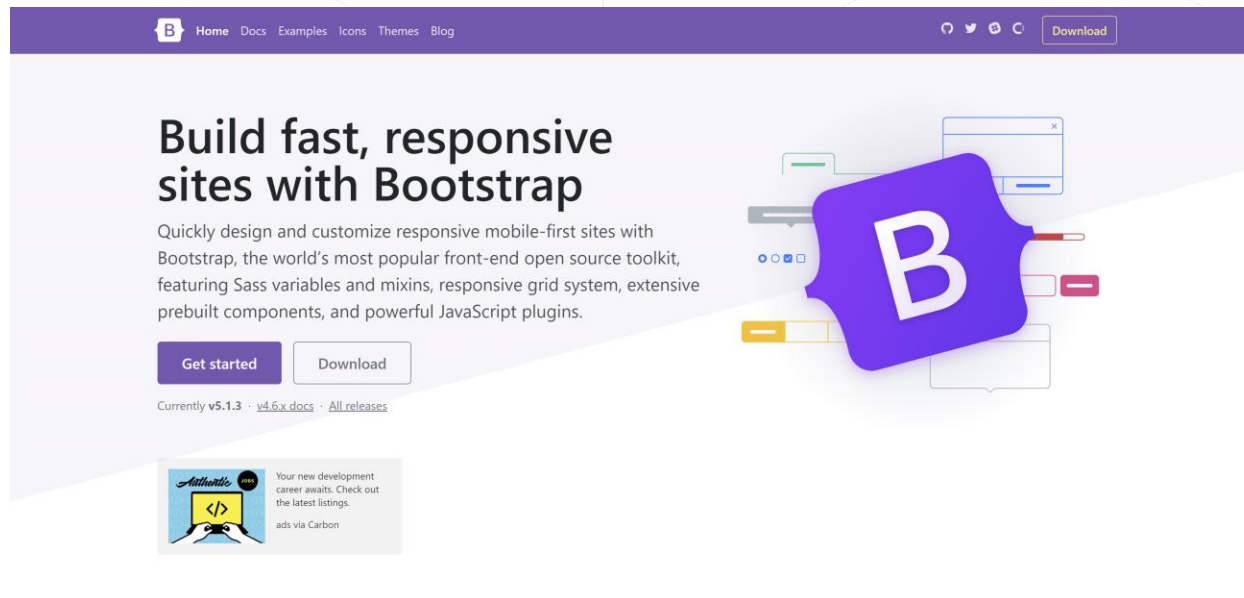


## Découverte du CSS et application

### Bootstrap

Bootstrap est un Framework CSS.

Il permet de créer des pages web rapidement grâce à ses composants qui sont pré-fabriqués (structure + css)  
Les classes de Bootstrap permettent d'utiliser un css déjà existant et optimisé pour différentes tailles d'écran.



- Créé en 2011
- Développé par Twitter
- Gratuit (open-source)

<https://getbootstrap.com/>