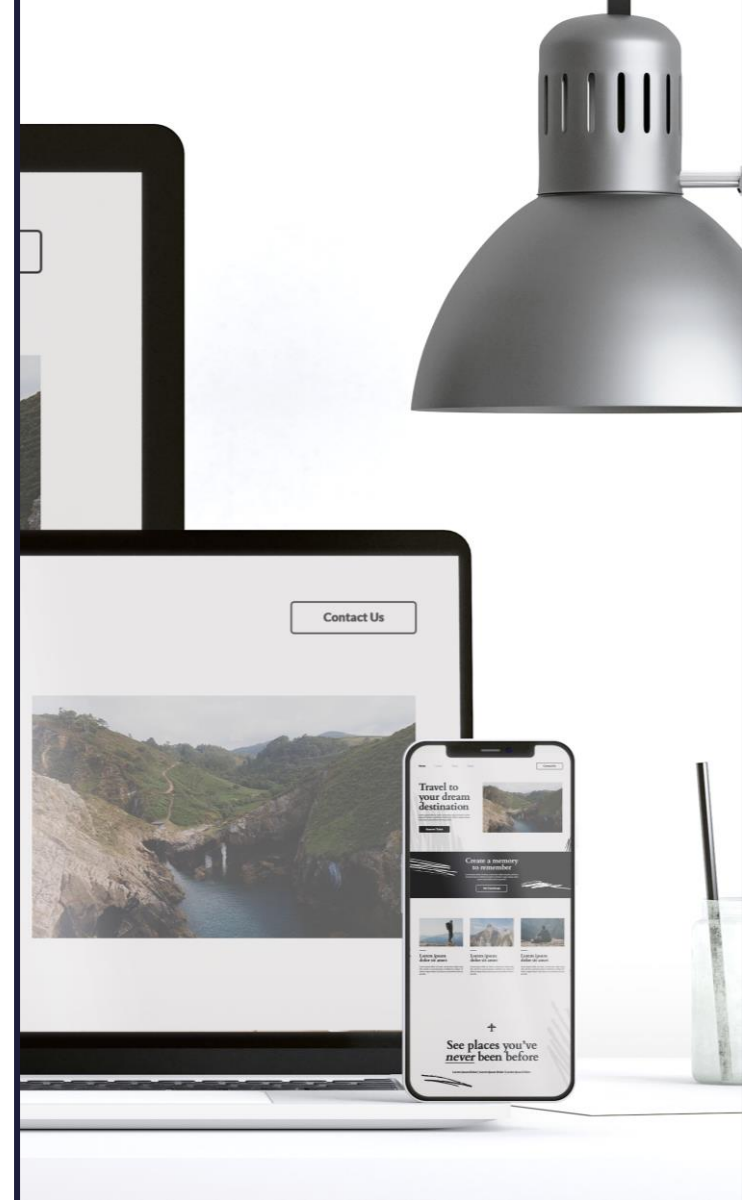


Cursus SALESFORCE

M2I - Formation - 2023

Donjon Audrey

Module HTML – CSS – Responsive





1. Introduction
2. Préparer l'environnement
3. Le HTML (6 à 19)
4. Le CSS (20 à 30)
5. Responsive Web Design (31 à 36)

Introduction

Technologies du web qui seront abordés :

- HTML (HyperText Markup Language) : Structure ma page Web
- CSS (Cascading StyleSheets) : Met en forme ma page Web
- Responsive : Utilisation des media queries

Outils :

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

Préparer l'environnement



1. Installer VSCode :

<https://code.visualstudio.com/Download>

2. Installer Google chrome ou Mozilla firefox :

https://www.google.com/intl/fr_fr/chrome/

<https://www.mozilla.org/fr/firefox/new/>



3. Télécharger L'extension chrome ou Firefox Web Developer :

<https://chrome.google.com/webstore/detail/web-developer/>

<https://addons.mozilla.org/fr/firefox/addon/web-developer/>

4. Installer les extensions vs code suivantes :

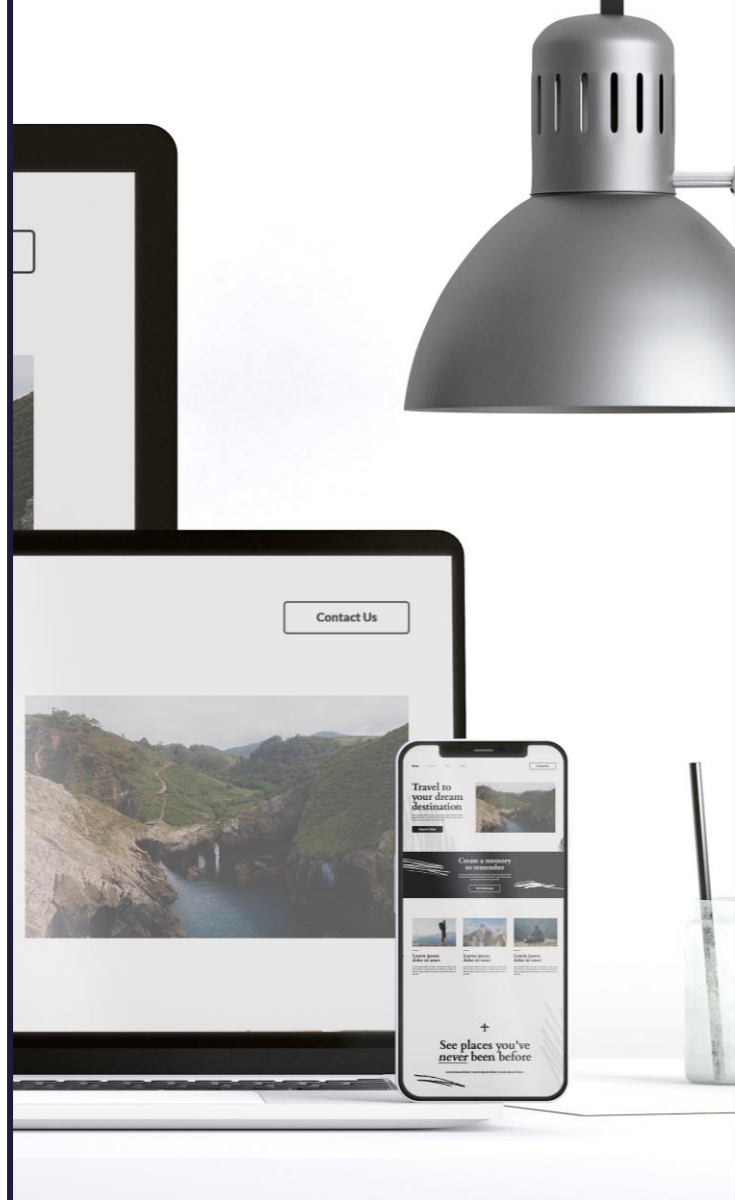
a) French Language

b) Trailing spaces

c) Auto Rename tag



Le HTML



Introduction HTML

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

Les **bases** du HTML

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

```
<balise-double></balise-double> ou <balise-orpheline>
```

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

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

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

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


Les balises

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

Paragraphe	<code><p>Je suis un paragraphe</p></code>
Titres (6 niveaux)	<pre> <h1>Je suis un Titre niveau 1</h1> <h2>Je suis un Titre niveau 2</h2> <h3>Je suis un Titre niveau 3</h3> <h6>Je suis un Titre niveau 6</h6> </pre>
Saut de ligne	<code>
</code>
Listes (ordonnées et non ordonnées)	<pre> je suis la première ligne de la liste à puce je suis la 2ème ligne de la liste à puce je suis la 3ème ligne de la liste à puce je suis la 4ème ligne de la liste à puce je suis la première ligne de la liste numérotée je suis la 2ème ligne de la liste numérotée je suis la 3ème ligne de la liste numérotée je suis la 4ème ligne de la liste numérotée </pre>

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

Image	<code></code>
Lien (relatif ou absolu)	<code></code> lien absolu : construit avec un protocole et nom de domaine <code></code> <code></code> lien relatif : construit par rapport à la page actuelle <code></code>
Balises sémantiques de texte	<code></code> Accentuation du texte <code></code> <code></code> Texte important <code></code> <code><mark></code> Texte pertinent dans son contexte <code></mark></code>

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

Les attributs

- ▶ Un attribut est une **information additionnelle/spécification** liée à une balise.
- ▶ Les attributs se déclarent dans **les balises ouvrantes** et sont **cumulables**.
- ▶ La syntaxe d'un attribut est la suivante :

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

- ▶ Quelques exemples :

```
> <h1 color="red" align="center"></h1>  
>   
> <a href="https://developer.mozilla.org/fr/docs/Web">Lien vers developer.mozilla.org</a>
```

Exercice 01

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

- ▶ Un titre d'onglet "CRM"
- ▶ Une icône de CRM dans l'onglet
- ▶ Un titre de niveau 1: "MyCRM"
- ▶ Un titre de niveau 2: "Description"
- ▶ Une description: "Cette application permet de faire de la gestion de commande et de client."
- ▶ Les mots "commande" et "client" doivent ressortir en important



Exercice 02

Reprendre la page et ajouter :

- ▶ Une petite image de CRM
- ▶ Deux titres de niveau 3: “Clients” et “Commandes”
- ▶ Une liste de client: “Jean Dupont”, “Sarah Abdram” et “Rachel Taeck”
- ▶ Une liste de commande: “CMD01”, “CMD02 ”, “CMD03” et “CMD04”
- ▶ Un lien de redirection vers le site du CRM



Les spécificités du **tableau**

▸ Principales balises pour afficher un tableau :

- **Structure** (englobe le tableau) :
- **En-tête de tableau** (header) :
- **Contenu de tableau** (body) :
- **Ajout d'une ligne** :
- **Ajout de cellule dans une ligne** :

>	<table> </table>		
>	<thead></thead>		
>	<tbody></tbody>		
>	<tr></tr>		
>	<th></th>	>	<td></td>

▸ Exemple d'attribut :

- **Bordures** (border=0 par défaut !) :
- **Fusionner les cellules** (horizontalement) :
- **Fusionner les cellules** (verticalement) :
- **Gérer l'alignement** :
- **Fixer la dimension** de la cellule :

>	<code><table border="1"></code>
>	<code><td colspan="2">Cellule qui prend 2 colonnes</td></code>
>	<code><td rowspan="2">Cellule qui prend 2 lignes</td></code>
>	<code><td align="right">Cellule</td></code>
>	<code><td width="50%">Cellule</td></code>

Note : attention l'utilisation d'attribut pour mettre en forme le tableau (bordures, largeur et alignement sont utilisés ici à titre d'exemple avant d'aborder le css, en tant normal il est déconseillé d'utiliser ces attributs directement sur les balises mais plutôt de le faire en css (ce qui sera vu plus tard dans le cours) !

Exercice 03

Reprendre la [page index.html](#) en remplaçant les listes à puces des clients et des commandes par **2 tableaux**, **centré sur la page** et affiché **l'un en dessous de l'autre**.

Les tableaux doivent avoir le même visuel que ci-après :

Clients

Nom	Prénom
Dupont	Jean
Abdram	Sarah
Taeck	Rachelle

Commandes

Numéro de commande	État
CMD01	Payé
CMD02	Payé
CMD03	Non payé
CMD04	En attente de paiement

Exercice 04

Reprendre la page [index.html](#) et reproduire le rendu suivant avec les tableaux HTML.

SPA de Chataîgneraï		SPA de Chambort	
Chat	4	Chat	2
Chien	2	Chien	5
Chaton	3	Chaton	6
Chiot	5	Chiot	2



Les spécificités du formulaire

► Principales balises pour afficher un formulaire :

- Structure : `<form></form>`
- Attribut de Nom unique : `name="form_name"`
- Attribut de méthode: `method="GET"`

► Balises de formulaires :

- Étiquette de champ : `<label> </label>`
- Champs de saisie : `<input>` (text, number, date, checkbox, password ...)
- Liste déroulante:
`<select><option></option></select>`
- Bouton envoyer: `<input type="submit">`
- Champs de saisie spécial:
`<textarea></textarea>`

```
<form action="cible.php" method="GET">
  <!--Champ(input) Prénom(de type text) avec son étiquette(label)-->
  <div>
    <label for="form_id">Prénom :</label><!--Attribut for(label) et
    id(input) pour identifier le champ et son étiquette-->
    <input id="form_id" type="text" name="form_name"><!--Attribut name :
    envoie des données avec le nom(name) du champ-->
  </div>
  <!--Champ(input) Nom (de type text) avec son étiquette(label)-->
  <div>
    <label for="form_id">Nom :</label>
    <input id="form_id" type="text" name="form_name">
  </div>
  <!--Liste déroulante-->
  <div>
    <label for="color">Quelle est votre couleur préférée ?</label>
    <select name="color" id="color">
      <option value="rouge">Rouge</option>
      <option value="vert">Vert</option>
      <option value="bleu">Bleu</option>
      <option value="violet">Violet</option>
      <option value="aucune">Je n'ai pas de couleur préférée</option>
    </select>
  </div>
  <!--Champ de texte multi-ligne-->
  <div>
    <label for="message">Votre message :</label>
    <!-- rows = hauteur en nombre de lignes de texte, cols = largeur en
    nombre de caractères -->
    <textarea name="message" id="message" cols="30" rows="10"></textarea>
  </div>
  <!--Bouton envoyer-->
  <div>
    <input type="submit" value="Envoyer">
  </div>
</form>
```

Exercice 05

Réaliser une **fiche d'inscription** des animaux sur un site de garde d'animaux.

Critères:

- ☐ Nom / Prénom du propriétaire
- ☐ Date de naissance du propriétaire
- ☐ Type d'animal domestique que je souhaite faire garder avec plusieurs choix possible (chien, chat, rongeur, poisson, autre)
- ☐ Si autre animal que la liste préciser ici
- ☐ Adresse email du propriétaire
- ☐ Numéro de téléphone du propriétaire
- ☐ Dernière page de vaccin de l'animal

Inscription de l'animal à faire garder

Nom du propriétaire :
 Prénom du propriétaire :
 Date de naissance du propriétaire :
 Type d'animal à faire garder

Si vous avez choisi "autre" précédemment, veuillez préciser de quel animal il s'agit :

Adresse mail du propriétaire :

Numéro de téléphone du propriétaire :

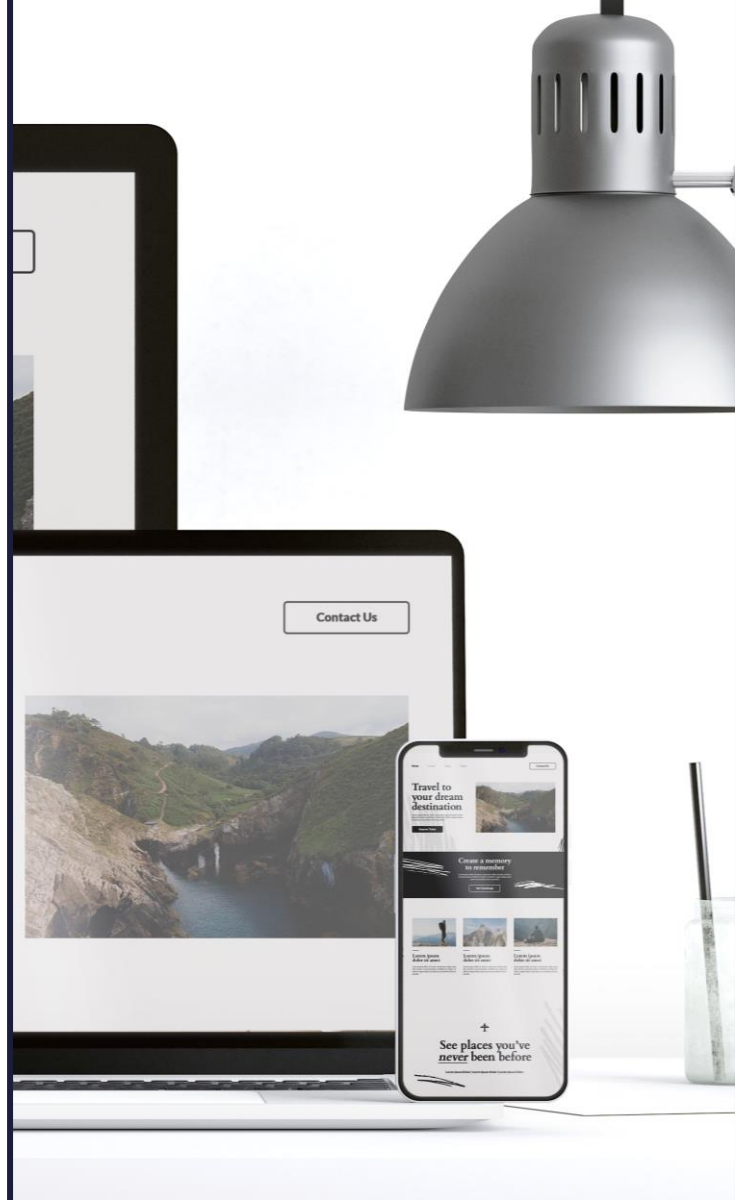
Dernière page de vaccin de l'animal : Aucun fichier sélectionné.

Les balises **sémantiques de structure**

<code><header></header></code>	Représente l'en-tête d'une page ou d'une section (peut contenir des éléments comme des logos, des titres, des menus de navigation)
<code><nav></nav></code>	Menu de navigation (contient généralement un ensemble de liens vers d'autres pages ou sections du site)
<code><main></main></code>	Contenu principal de la page (Spécifie le contenu principal d'un document. Il y a généralement un seul élément <code><main></code> par page.)
<code><section></section></code>	Sert à regrouper des contenus en fonction de leur thématique (Définit une section dans un document, généralement avec un thème ou une signification cohérente, et souvent accompagnée d'un en-tête)
<code><article></article></code>	La balise <code><article></code> Utilisé pour un composant autonome d'une page qui est suffisamment indépendant pour être distribué ou réutilisé (comme un article de blog, un post de forum, un widget)
<code><aside></aside></code>	Informations complémentaires au document (comme des barres latérales, des encadrés publicitaires.)
<code><figure></figure></code> <code><figcaption></figcaption></code>	Utilisé pour marquer une illustration, un diagramme, des photos, ou un code, généralement accompagné d'une légende
<code><footer></footer></code>	Représente le pied de page d'une page ou d'une section (comme les droits d'auteur, des liens vers des politiques de confidentialité, ou des crédits)

Note : Ces balises **améliorent la lisibilité du code** pour les développeurs, mais elles **aident aussi les moteurs de recherche** et les **lecteurs d'écran à comprendre la structure et le contenu d'une page web**, ce qui est essentiel pour le **SEO** et **l'accessibilité**.

Le CSS



Introduction au CSS

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

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

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

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

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

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

CSS

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

Quelques **sélecteurs CSS**

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

<p>Sélecteur d'élément ou balise HTML</p> <p>Ici : Touchera toutes les balises <code><p></p></code> de la page web</p>	<div>CSS</div> <pre>p{ color:■red; }</pre>
<p>Sélecteur de class</p> <p>Ici : Touchera la/les balises ayant la class <code>"text-red"</code></p>	<div>CSS</div> <pre>.text_red{ color:■red; }</pre>
<p>Sélecteur d'ID</p> <p>Ici : Touchera la balise ayant l'id <code>"color_main_title"</code></p>	<div>CSS</div> <pre>#color_main_title{ color:■brown; }</pre>
<p>Sélecteurs universels</p> <p>Ici : Touchera tous les éléments HTML de la page</p>	<div>CSS</div> <pre>*{ color:■red; }</pre>
<p>Sélecteur d'attribut</p> <p>Ici : Touchera toutes les balises <code><a></code> ayant comme attribut : <code>href="https://google.com"</code></p>	<div>CSS</div> <pre>a[href="https://google.com"]{ color:■green; }</pre>

Quelques Combinateurs

(sert à combiner plusieurs sélecteurs)

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

Quelques exemples de **propriété CSS**

Propriété	Syntaxe	Exemples
Couleur couleur de fond	color background-color	<code>color: red; color: rgb(255, 99, 71); color: #F10606;</code> <code>background-color: gray; background-color: #808080;</code> <code>background-color: rgba(255, 99, 71, 0.80);</code>
Police	font-family	<code>font-family: Times; font-family: Impact, Verdana, "Arial Black";</code>
Taille texte	font-size	<code>font-size: 15px; font-size: small; font-size: 1,5em;</code>
Styles de police	font-style font-weight ...	<code>font-style: italic; font-weight: bold;</code> <code>text-decoration: underline;</code>
Alignement	text-align	<code>text-align: center; text-align: right;</code> <code>text-align: justify;</code>
Marges	margin padding	<code>margin: 5px 5px 5px 5px; margin-top: 10px; margin-bottom: 30px; </code> <code>padding: 10px; padding-left: 15px;</code>

Quelques exemples de **propriété CSS**

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

Exercice

Choisissez un exercice effectué en html puis essayez de le styliser le plus possible en utilisant les propriétés vu précédemment



BOX MODEL

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

▸ Élément HTML de type **Block**

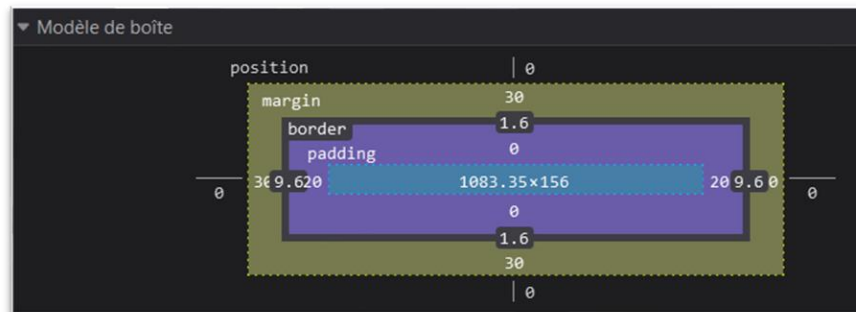
- **Occupe toute la largeur** disponible de son parent
- **L'élément suivant ou précédent** l'élément block se positionnera **au dessus ou en dessous de celui-ci** (retour à la ligne)
- Width & Height modifiables

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

▸ Élément HTML de type **Inline**

- **Occupe la place de leurs contenu**
 - L'élément inline suivant se **positionne à la suite**
 - **Width & Height non modifiables**
- et pas de **padding et margin (top et bottom)**

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



▸ Élément HTML de type **Inline-block**

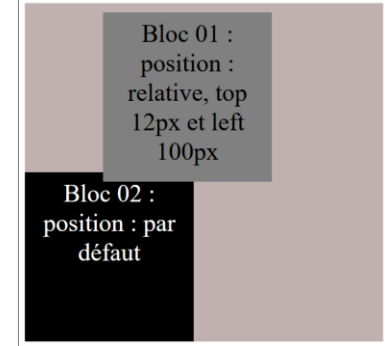
- Se comporte **en partie comme un inline** :
Occupe la place de leurs contenu / L'élément inline suivant se **positionne à la suite**
- Se comporte **en partie comme un block** :
Width & Height modifiables et padding et margin (top et bottom) possible

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

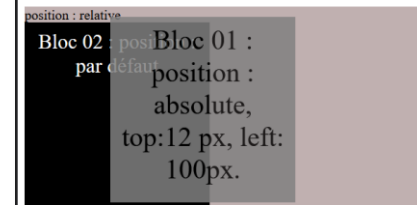
Flux et Positionnement

- ▶ Le **flux HTML** correspond à l'écoulement des informations (ou données) que le navigateur va interpréter et afficher. **Un navigateur commence par le haut de la page**, place les éléments HTML qu'il rencontre les uns à la suite des autres, **de gauche à droite** puis **de haut en bas**, à ceci près que cela dépend si ce sont des balises bloc ou en-ligne.
- ▶ La propriété CSS **position** permet de **gérer le positionnement** des éléments.
- ▶ On distingue **4 types principaux** de positionnement :
 - ▶ **(static)**: Valeur par défaut - Positionnement **statique** des éléments **dans le flux**
 - ▶ **relative** : Positionnement ajustée par rapport à la **position initiale** de l'élément **dans le flux**
 - ▶ **absolute** : Positionnement ajustée par rapport au **parent relative** le plus proche **hors flux** (chevauchement d'éléments alors possible)
 - ▶ **fixed** : **semblable à absolute**, mais en s'appuyant sur la **fenêtre visible** du navigateur et **hors flux**.
- ▶ **Positionnement ajustée avec les propriétés top, left, bottom et right**. Les valeurs négatives sont possibles.
- ▶ La propriété **z-index** permet de prioriser l'élément à afficher (**en cas de chevauchement**)

Positionnement relatif



Positionnement absolue



Exercice 06

Créer **2 pages web** :

- Page contact
- Page new-contact

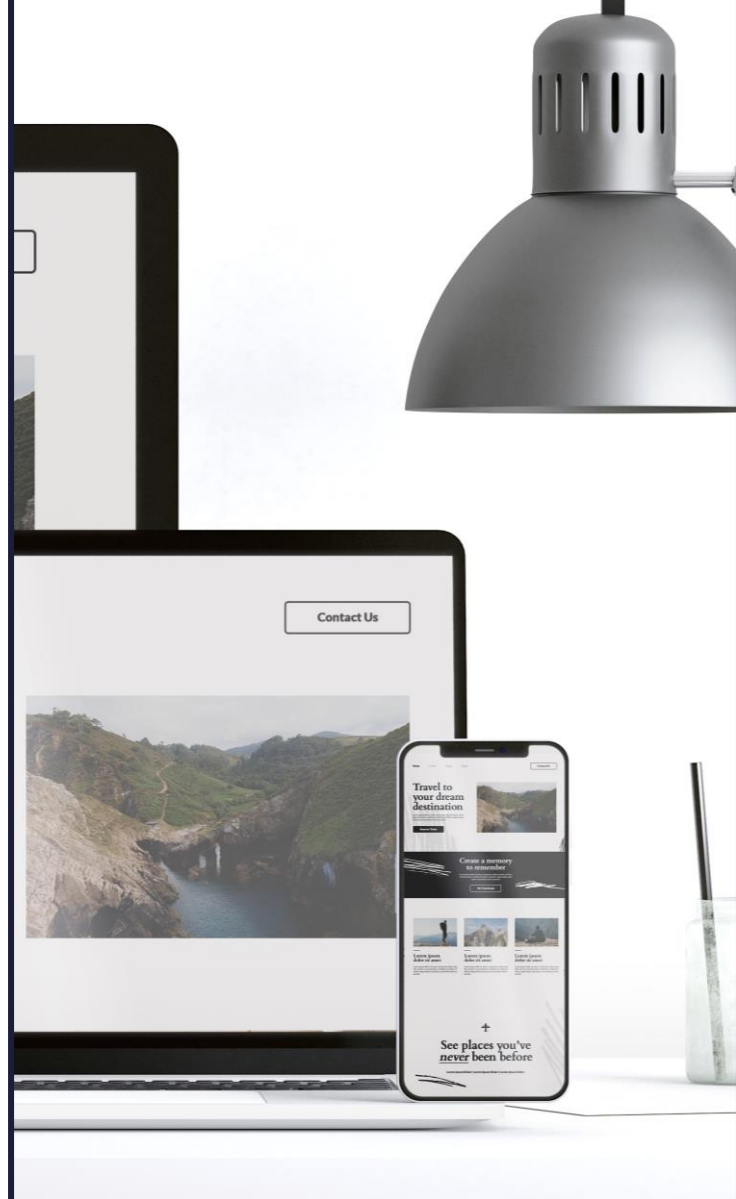
Voici les **codes de couleur** :

- Noir : #222222
- Vert : #578957
- Vert foncé : #406040
- Jaune : #c7b878

Aidez vous des 2 maquettes des 2 pages à reproduire qui vous seront fournies



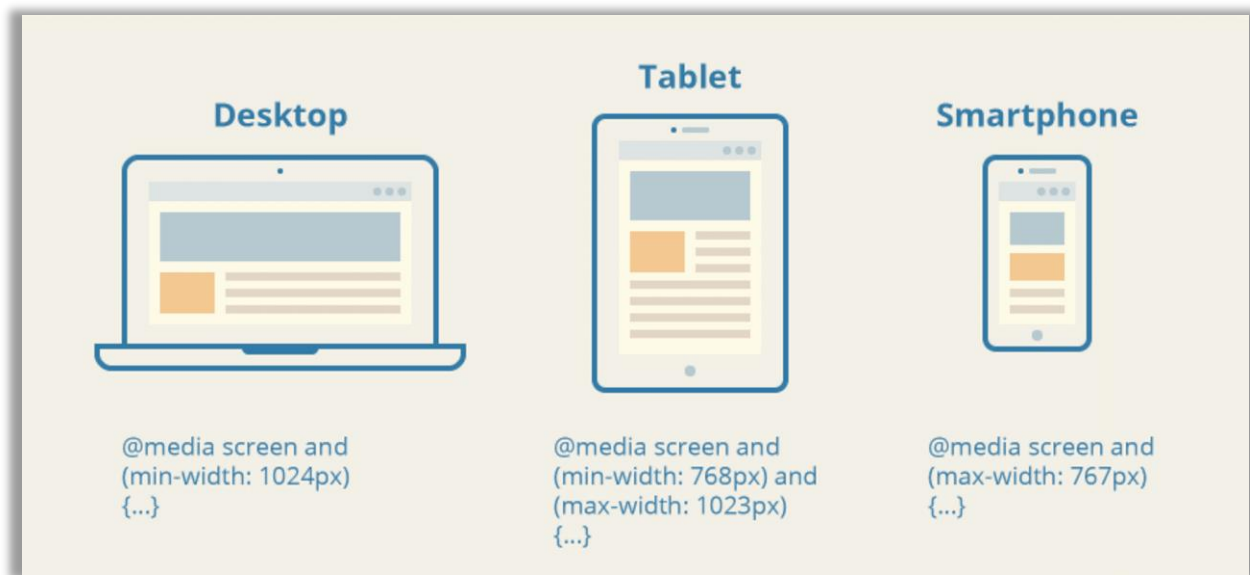
Le Responsive web design



Responsive web design

La **responsive web design** est une approche qui permet de créer des pages web qui sont toujours bien rendu quelque soit la taille du périphérique d'affichage.

En css nous allons mettre en place des sites responsives grâce aux **média queries**.



Mise en place des media queries

CSS

```
@media screen and (min-width:480px) {  
    /* Mon code css ici avec Sélecteurs, propriétés et valeurs */  
}  
  
@media screen and (min-width:768px) {  
    /* Mon code css ici avec Sélecteurs, propriétés et valeurs */  
}  
  
@media screen and (min-width:992px) {  
    /* Mon code css ici avec Sélecteurs, propriétés et valeurs */  
}  
  
@media screen and (min-width:1200px) {  
    /* Mon code css ici avec Sélecteurs, propriétés et valeurs */  
}
```

Synthèse sur les FlexBox

Flexbox est un **modèle de boîte flexible** permettant de gérer le positionnement des éléments dans un conteneur d'une manière plus fluide qu'avec les modèles de boîte classique.

Pour qu'un conteneur devienne un conteneur flex, il faut changer son type.

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

row : horizontale, c'est la valeur par défaut

column : verticale

row-reverse : horizontale inversé

column-reverse : verticale inversé

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

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

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

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

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

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

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

- **center** : éléments centrés au milieu de l'axe.

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

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

CSS

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

Css

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

Css

```
div{
  display: flex;

  /* Tous les éléments enfants seront disposés en
  ligne côte à côte */
  flex-direction: row;

  /* Si il y a un manque de place, les éléments
  passeront à la ligne du dessous */
  flex-wrap: wrap;

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

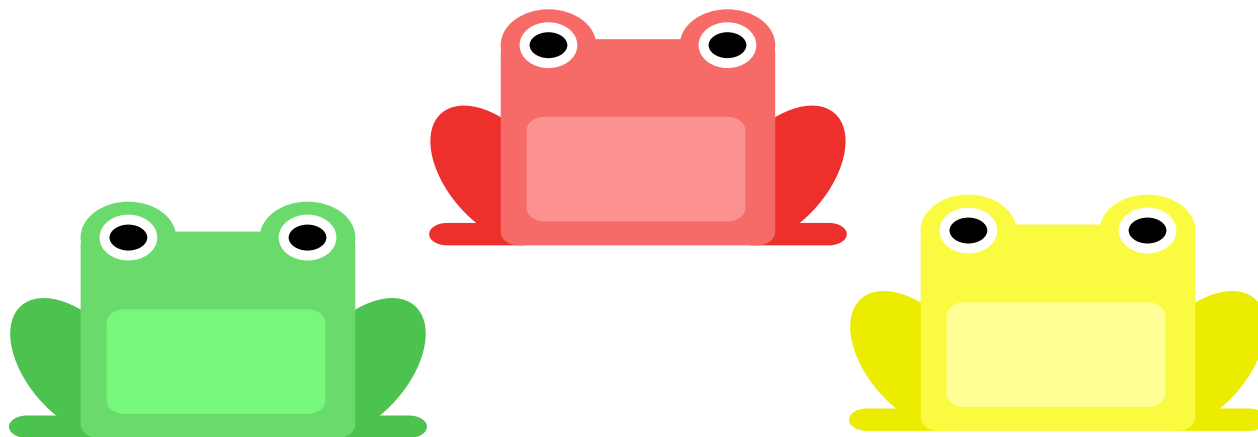
  /* Les éléments seront centrés sur l'axe
  secondaire (sur l'axe vertical dans cet exemple) */
  align-items: center;
}
```

Entraînement Flexbox : Jeux Flexbox Froggy

Outil pour tester les propriétés css qui existent :

<https://codepen.io/enxaneta/full/adLPwv/>

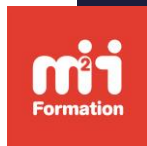
Lien vers le jeu : <https://flexboxfroggy.com/#fr>



Exercice 07

À partir de l'exercice 6 réalisé précédemment pour la page de la liste des contacts, adapter le pour avoir le même aperçu que les maquettes qui vous seront donné en utilisant les media queries, vous pouvez également utiliser les flex.





Fin du module



m2information.fr