

PROGRAMMATION WEB AVANCEE

TP 2

CSS3 SUITE

Objectifs:

- Utiliser JavaScript pour manipuler le contenu et l'apparence de pages Web
- Créer un design responsive

1. Manipulation de classes CSS

En HTML, chaque élément peut être associé à des classes. On les spécifie en les énumérant dans l'attribut class de l'élément.

Exemple:

```
<p id="premier">texte visible</p>
<p id="second" class="hidden">ce texte est caché</p>
```

L'élément <p> est associé à une classe hidden. Pour que tous les éléments associés à cette classe soient cachés, il suffirait alors de définir la règle CSS suivante:

```
.hidden {
  display: none;
}
```

En JavaScript, il est possible de modifier dynamiquement la liste de classes associées à un élément (et donc son affichage, en fonction des règles CSS qui sont associées à ces classes), avec la propriété `classList` et à ses méthodes (fonctions) `add()` et `remove()`.

Exemple:

```
var elem = document.getElementById('un-element');
elem.classList.add('hidden'); // la classe hidden va être associée à l'élément
```

L'objet **classList** a d'autres méthodes utiles, comme **toggle** (qui ajoute la classe si elle n'est pas présente, et la retire sinon), et **contains** (qui vérifie si l'élément a la classe).

Exercice 1: Afficher/cacher

- Créer une page web valide contenant les deux paragraphes `<p>` tels que définis plus haut.
- Y associer un fichier CSS contenant la règle `.hidden` tel que définie plus haut.
- Ouvrir la page, vérifier que seul le premier paragraphe est visible.
- Quel code JavaScript faut-il exécuter pour retirer la classe `hidden` du deuxième paragraphe ?

2. Accéder à des éléments par classe

L'API du DOM donne accès à une méthode de document qui permet d'accéder à l'ensemble des éléments qui portent une même classe `class`: `getElementsByClassName()`.

Considérons une page HTML contenant plusieurs paragraphes portant une même classe:

```
<p>C'est l'histoire d'un garçon troublé</p>
<p class="sp">car il s'avère que son ours en peluche a disparu</p>
<p class="sp">mais finalement il le retrouve</p>
```

Pour accéder à tous les éléments portant la classe `spoiler`, nous allons utiliser le code JavaScript suivant:

```
var elements = document.getElementsByClassName(sp); // tableau d'éléments
```

À noter que, contrairement à `getElementById()`, cette fonction ne retourne pas un seul élément, mais un tableau de plusieurs éléments.

Exercice 2 :

- Créer une page web valide contenant les trois paragraphes `<p>` tels que définis dans l'exemple ci-dessus.
- Y associer un fichier CSS contenant la règle `.hidden` tel que définie plus haut.
- Ouvrir la page, vérifier que les trois paragraphes sont visibles.
- Quel code JavaScript faut-il exécuter pour ajouter la classe `hidden` aux paragraphes qui portent la classe `spoiler` ?
- Ajouter un bouton à la page HTML, et y intégrer le code JS nécessaire pour que les spoilers soient cachés seulement une fois que l'utilisateur aura cliqué sur le bouton.

faire un fichier `.js` à part.

3. Manipulation de styles CSS

Il est possible de modifier dynamiquement le style d'éléments HTML sans avoir à manipuler de classes CSS. En effet, tous les éléments HTML fournissent une **propriété JavaScript style**.

La propriété style est de type Object.

L'objet associé à la propriété style est structuré similairement à une règle CSS: il est constitué de propriétés clé-valeur.

Exemple de règle CSS:

```
#mon-element {  
  border: 1 solid black;  
  background-color: red;  
}
```

Exemple de changement de style

Soit le fichier HTML suivant:

```
<p id="premier">texte important</p>  
<p id="second">autre texte</p>
```

Pour modifier la couleur de fond du premier paragraphe, il faut exécuter le code JavaScript suivant:

```
var element = document.getElementById('premier');  
element.style.backgroundColor = 'red';
```

Exercice 3 : Surbrillance au clic

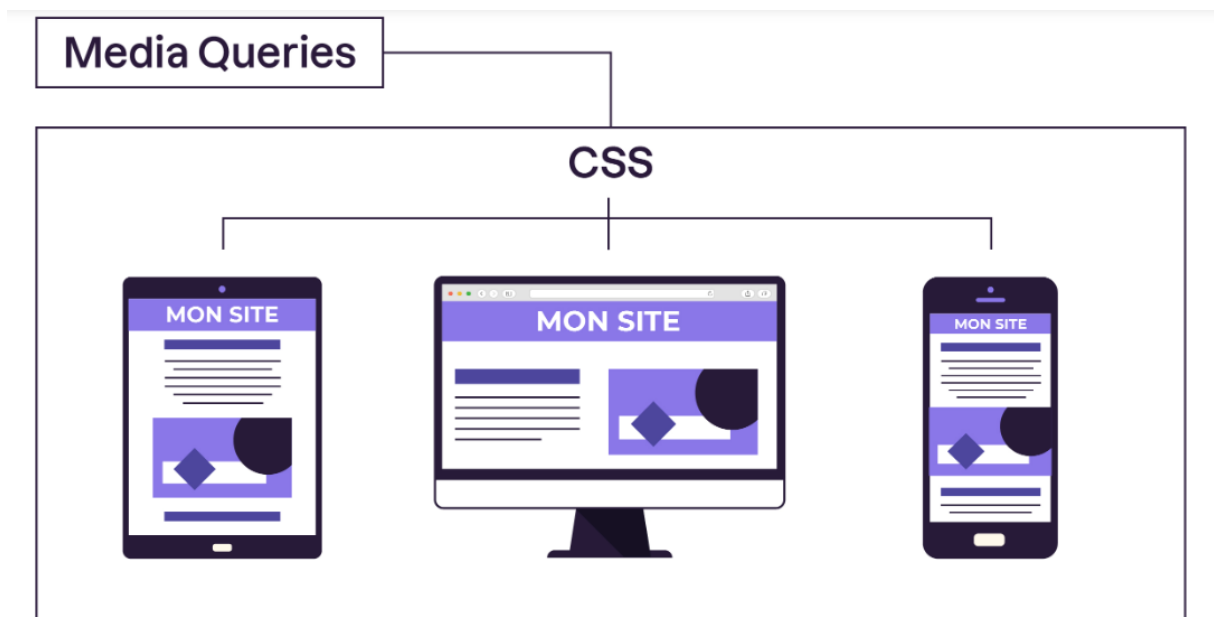
- Créer une page web valide contenant les deux paragraphes <p> tels que définis dans l'exemple ci-dessus.
- Ouvrir la page, vérifier que les deux paragraphes sont visibles.
- (*étape optionnelle pour vous entraîner*) Intégrer le code JavaScript nécessaire pour que la couleur de fond du deuxième paragraphe devienne jaune.
- Intégrer le code JavaScript nécessaire pour que la couleur de fond du deuxième paragraphe devienne jaune, seulement une fois que l'utilisateur aura cliqué dessus.
- Bonus: Intégrer le code JavaScript nécessaire pour que la couleur de fond de n'importe quel paragraphe de la page devienne jaune quand l'utilisateur clique dessus. (indice: il va falloir utiliser le paramètre fourni à l'appel de votre fonction onclick, nous ne l'avons pas encore vu en cours).

4. Créer un design responsive: en utilisant les media queries

Les Media Queries correspondent à des styles CSS conditionnels.

Les Media Queries se basent sur la règle CSS **@media** qui va permettre de définir différents styles CSS pour différents types d'appareils media et selon différents critères.

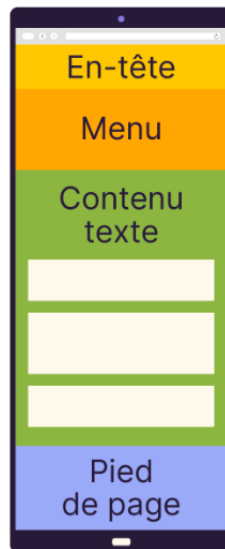
Grâce aux Media Queries, on peut présenter le même contenu HTML de différentes façons en fonction de l'appareil utilisé pour accéder aux pages web.



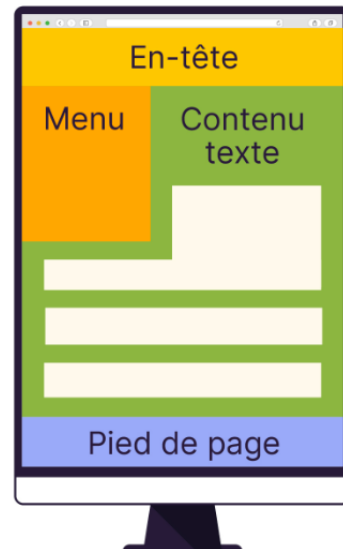
Smartphone



Tablette



Ordinateur



La présentation des blocs varie en fonction de l'appareil utilisé

Le fonctionnement des Media Queries (syntaxe)

Dans la règle **@media**, deux types de conditions ou de contraintes seront placés : une condition sur le media utilisé pour afficher la page et une condition sur les caractéristiques du media.

La syntaxe :

@media screen and (max-width: 1200px) {

/* propriétés CSS ici, avec les sélecteurs*/

Les tailles que nous indiquons avec **min-width** ou **max-width** sont souvent appelées "points de ruptures", ou *breakpoints* en anglais, car ce sont les points où le design change.

Dans l'exemple : cela signifie que le style écrit entre les accolades ne sera appliqué que pour les écrans dont la taille est inférieure ou égale à 1 200 pixels.

Afin que les Media Queries soient prises en compte correctement sur tous les dispositifs, il est faut **obligatoirement ajouter** dans l'en-tête (partie `<head>` d'une page) la ligne suivante :

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

On demande à ce que le "viewport" du navigateur, c'est à dire la surface d'affichage du navigateur s'adapte à la largeur d'affichage de l'appareil. Cela permet notamment un affichage correct sur les téléphones mobiles.

Les règles disponibles :

Il existe de nombreuses règles permettant de construire des media queries. Les plus utilisées :

- **height** : hauteur de la zone d'affichage (fenêtre) ;
- **width** : largeur de la zone d'affichage (fenêtre) ;
- **orientation** : orientation du périphérique (portrait ou paysage) ;
- **media** : type d'écran de sortie. Avec la valeur la plus utilisée :
 - **screen** : écran "classique",
 - **all** : tout type de média,
 - **print** : imprimante (pratique pour formater un contenu pour l'imprimer).

On peut rajouter le préfixe **min-** ou **max-** devant la plupart de ces règles. Ainsi, **min-width** signifie "largeur minimale", **max-height** signifie "hauteur maximale", etc.

Les règles peuvent être combinées à l'aide des mots suivants:

- **only** : "uniquement" ;
- **Or** : il suffira qu'une condition soit vérifiée.
- **and** : "et" ; : les deux conditions devront être vérifiées.
- **not** : "non". En plaçant le mot clef **not** avant le type de media, les règles s'appliqueront à tous les appareils media sauf celui spécifié.

Exemples :

```
/* Sur les écrans, quand la largeur de la fenêtre fait au maximum 1280px */  
@media screen and (max-width: 1280px)  
  
/* Sur tous types d'écran, quand la largeur de la fenêtre est comprise entre 1024px et 1280px */  
@media all and (min-width: 1024px) and (max-width: 1280px)  
  
/* Sur tous types d'écrans orientés verticalement */  
@media all and (orientation: portrait)
```

Ressources à voir

<https://www.alsacreations.com/article/lire/930-css3-media-queries.html>

<https://www.w3.org/TR/CSS21/media.html>

<https://www.w3.org/TR/mediaqueries-3/>

<https://webdesignerwall.com/tutorials/css3-media-queries>

Exercice 4 : Page CV

1. Reprendre l'exercice 2 du TP1, ajouter un lien à la page Moncv.html

Votre page CV doit contenir une photo avec un paragraphe de présentation et des informations de contact (tel, mail, lien vers facebook, twitter). Utiliser les icônes pour vos liens de contact.

Ensuite, vos expériences professionnelles, puis votre formation académique, vos compétences et enfin pour finir avec vos centres d'intérêt.


Ces informations doivent être présentes les unes au dessous des autres pour la version mobile.

La version mobile de votre CV sera la version standard. Pour créer cette version, vous allez utiliser le modèle des boîtes flexibles en choisissant une orientation en colonne principalement.

Pour la version bureau de votre CV HTML et CSS, il faut permettre de placer certains éléments de section côte-côte plutôt que les uns en dessous des autres.

Exemple : Partie d'un CV

John Doe's CV



john.doe@example.com
+00 01 2345 5678

Soft Skills

- Some
- Soft
- Skills

Hard Skills

- Some
- Hard
- Skills

Work experience

2022 - 2023

A beautiful Company
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut scelerisque tristique mauris, a porttitor enim mollis in. Maecenas vel nisi elit. Sed imperdiet tortor nec mauris rhoncus eleifend. Donec id eros laoreet, vulputate sem quis, pellentesque felis.

2021 - 2022

A Very Nice Company
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut scelerisque tristique mauris, a porttitor enim mollis in. Maecenas vel nisi elit. Sed imperdiet tortor nec mauris rhoncus eleifend. Donec id eros laoreet, vulputate sem quis, pellentesque felis.

Education

2019 - 2021

Master Degree in Computer Science
University The Best
Beautiful City, Beautiful Country

2016 - 2019

Bachelor of Technology
In Very Great University
Beautiful City, Beautiful Country

Pour créer cette version bureau, utiliser la règle **CSS @media screen and (min-width: 980px){}**. Les styles définis dans cette règle vont donc s'appliquer pour tous les appareils disposant d'un écran et lorsque la fenêtre est plus large que 980px.