

## Semaine 04 - Mise en page adaptative - suite | Sujets complémentaires

Cette semaine, nous allons continuer à travailler la mise en page adaptative avec l'approche «Mobile avant tout» (*Mobile First*). Nous allons également voir 2 petits sujets complémentaires portant sur des **astuces en HTML/CSS** qui nous serviront notamment à animer ultérieurement un menu adaptatif avec un bouton «burger». Nous allons également voir l'usage de la **police d'icônes de Google**.

### Astuces en HTML-CSS avec input checkbox et :checked

Les langages HTML et CSS ont bien évolué depuis des années et les intégrateurs, web designers et UX designers en ont profité à bien des égards pour trouver, voire créer, de nouvelles techniques de mise en page et de mise en forme. Nous allons voir que nous pouvons **faire des choses intéressantes** sur le plan de l'**interactivité (sans JavaScript)** juste avec un bon vieux `<input type="checkbox" />`, une pseudo-classe `:checked` et les sélecteurs d'adjacence `+` et `~`.

### Quelques rappels de CSS au sujet de `+` et `~`, au cas où...

Pour rappel, les combinateurs «`+`» et «`~`» utilisés en CSS permettent **respectivement** de cibler le **voisin direct** (qui suit) ou les éléments adjacents qui suivent le premier sélecteur. Par exemple, «`li:first-child + li`» vise le «`li`» voisin directement situé après le premier «`li`» d'une liste. En comparaison, «`li:first-child ~ li`» cible tous les «`li`» voisins **directement** situés **après** le premier «`li`». Il ne s'agit donc pas de blocs descendants mais bien de voisins (frères ou sœurs) **situés plus bas** dans le **même conteneur parent**. Le code qui suit illustre cet exemple, à partir de la structure HTML suivante:

```
<ul>
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li>4</li>
</ul>
```

<code>li:first-child + li{</code> <code>background-color: aqua;</code> <code>}</code>	
<code>li:first-child ~ li{</code> <code>background-color: aqua;</code> <code>}</code>	

Avec un peu de **créativité**, les **combinateurs** permettent de faire des modifications d'états CSS complexes comme des **effets de survol éloigné**.

## Astuce CSS avec checkbox

### La formule de base

Les cases à cocher (*checkboxes*) sont puissantes lorsqu'on les combine avec un CSS astucieux. Cette note de cours vise à montrer comment on peut changer le style ou non d'un autre élément, et ce sans JavaScript. Tout commence avec un peu de HTML :

```
<input id="caseAcocher" type="checkbox">
<label for="caseAcocher">Légende</label>
```

L'attribut **for** sur le **<label>** correspond à l'attribut **id** qu'on retrouve sur **<input>**, par conséquent lorsqu'on clique sur **<label>** on bascule l'état de la case à cocher **<input>** (comme si on cliquait sur ce dernier élément). C'est important parce que la **prochaine étape** consistera à **cacher** la case **<input>**.

### Cacher la case à cocher

La prochaine étape vise à **cacher la case à cocher**. Le plus généralement, on se contentera uniquement de ne pas l'afficher. Ainsi, dans le code CSS, on aura le code suivant:

```
#caseAcocher {
    display: none;
}
```

### Gérer l'état de la case à cocher pour appliquer des styles CSS à des éléments HTML voisins

Nous utilisons ensuite un **mélange** de **pseudo-classe :checked** et de combinateur d'adjacence (+ ou ~) pour appliquer des styles à des éléments HTML voisins. Par exemple, en considérant le code HTML:

```
<input type="checkbox" id="caseAcocher">
<label for="caseAcocher"> LÉGENDE DE LA CASE À COCHER</label>
<div>Visible ou non selon l'état de la case à cocher</div>
```

Voici le code CSS pour modifier l'apparence de la balise **<label>** et de faire apparaître ou non la **<div>**:

```
div{
    opacity: 0;
}

#caseAcocher:checked+label {
    font-weight: bolder;
}

#caseAcocher:checked~div {
    opacity: 1;
}
```

Le **raisonnement sous-jacent au code CSS** ci-haut est le suivant: *quand la case à cocher est cochée (checked), trouver le <label> immédiatement après la case et lui appliquer un style. Trouver également les <div> du même conteneur parent qui suit et leur appliquer également un style.*

Voilà la base. Cette astuce permet de faire des effets de transition d'états CSS relativement complexes. Pour avoir une idée des nombreuses possibilités, vous pouvez consulter le blogue de Mathieu Chartier à ce sujet dont la référence est à la fin de la note de cours.

## Police d'icônes de Google (Material Icons)

Une police d'icônes est une police de caractères vectoriels qui utilise des pictogrammes fonctionnels à la place de lettres. À la session dernière vous avez appris à utiliser la police de la bibliothèque **Font Awesome**. Cette session-ci nous allons utiliser celle développée par Google dans le cadre du système de conception **Material Design**, et qui est ainsi nommée **Material Icons**.

La **police d'icônes de Google** est accessible à partir du site **Google Web Font**. Elle ne pèse que 42 Ko dans son plus petit format woff2 et 56 Ko au format woff standard. Ainsi, **plus de 900 icônes** sont accessibles à partir d'un seul petit fichier.

Il existe plusieurs façons d'utiliser les icônes de cette police, nous allons voir ici la plus simple.

## Lier la polices d'icônes

Semblable à l'incorporation d'autres polices de Google Font, pour utiliser celle des icônes, il suffit d'intégrer le lien dans le <head> de la page HTML:

```
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="feuille de style">
```

Une **classe CSS supplémentaire** sera déclarée (appelée **.material-icons**). Cette classe définit le style d'affichage de base des icônes (couleur, taille, etc.). Cette classe CSS peut être modifiée pour répondre aux besoins visuels du projet.

## Incorporer des icônes dans la page HTML

Il est facile d'intégrer des icônes dans une page Web avec cette police. Il suffit d'utiliser l'une ou l'autre des balises <i> ou <span>, d'associer la classe **.material-icons** à la balise et d'intégrer le nom de l'icône. Les différents choix sont accessibles à l'URL suivant: <https://fonts.google.com/icons?icon.set=Material+Icons>  
Voici un petit exemple, pour l'icône dénommée: **face** :

```
<span class="material-icons">face</span>
```

## Quelques références utiles sur les sujets complémentaires de la semaine 04

- Chartier, Mathieu (Mai 2021). Astuces en HTML-CSS avec input checkbox et :checked. repéré à: <https://blog.internet-formation.fr/2021/05/astuces-en-html-css-avec-input-checkbox-et-checked/>
- Google (Juillet 2022). Material Icons Guide. repéré à : <http://google.github.io/material-design-icons/#what-are-material-icons->