responsive design avec les Media Queries

Les media queries font partie des nouveautés de CSS3. Il ne s'agit pas de nouvelles propriétés, mais de *règles* que l'on peut appliquer dans certaines conditions. Concrètement, vous allez pouvoir dire « Si la résolution de l'écran du visiteur est inférieure à tant, alors applique les propriétés CSS suivantes »

Vous pouvez changer l'apparence de votre site en fonction d'autres critères comme le type d'écran (smartphone, télévision, projecteur...)

Appliquer une media query

- en chargeant une feuille de style `.css` différente en fonction de la règle (ex. : « Si la résolution est inférieure à 1 280 px de large, charge le fichier `petite_resolution.css` ») ;

- en écrivant la règle directement dans le fichier `.css` habituel (ex. : « Si la résolution est inférieure à 1 280 px de large, charge les propriétés CSS ci-dessous »).

Chargement d'une feuille de style différente

On peut lui ajouter un attribut `media` , dans lequel on va écrire la règle qui doit s'appliquer pour que le fichier soit chargé:
<link rel="stylesheet" media="screen and (max-width: 1280px)" href="petite_resolution.css" />

ou normale :
<link rel="stylesheet" href="style.css" />

Chargement des règles directement dans la feuille de style

Une autre technique, que je préfère personnellement pour des raisons pratiques, consiste à écrire ces règles dans le même fichier CSS que d'habitude:

```
@media screen and (max-width: 1280px)
{
   /* Rédigez vos propriétés CSS ici */
}
```

Les règles disponibles

- `color` : gestion de la couleur (en bits/pixel) ;
- `height` : hauteur de la zone d'affichage (fenêtre) ;
- `width` : largeur de la zone d'affichage (fenêtre) ;
- `device-height` : hauteur du périphérique ;
- `device-width` : largeur du périphérique ;
- `orientation` : orientation du périphérique (portrait ou paysage) ;
- `media` : type d'écran de sortie. Quelques-unes des valeurs possibles :
    - `screen` : écran « classique »,
    - `handheld` : périphérique mobile,
    - `print` : impression,
    - `tv` : télévision,
    - `projection` : projecteur,
    - `all` : tous les types d'écrans.

On peut rajouter le préfixe `min-` ou `max-` devant la plupart de ces règles. Ainsi, `min-width` signifie « largeur minimale », `max-height` « hauteur maximale », etc.
La différence entre `width` et `device-width` se perçoit surtout sur les navigateurs mobiles des smartphones ; nous en reparlerons plus loin.

```
/* 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 les téléviseurs */
@media tv

/* Sur tous types d'écrans orientés verticalement */
@media all and (orientation: portrait)
```

Tester les media queries : exemple de code

```
/* Paragraphes en bleu par défaut */
p
{
   color: blue;
}

/* Nouvelles règles si la fenêtre fait au plus 1024px de large */
@media screen and (max-width: 1024px)
{
   p
   {
      color: red;
      background-color: black;
      font-size: 1.2em;
   }
}
```

La page

```
#bloc_page
{
   width: 900px;
   margin: auto;
}


@media all and (max-width: 1024px)
{
   #bloc_page
   {
      width: auto;
   }
}
```

`auto` est la valeur par défaut de la propriété `width` . Par défaut, les blocs ont une largeur automatique (ils prennent toute la place disponible). Cette valeur « écrase » celle que nous avions forcée à 900 px quelques lignes plus haut : nous revenons donc au comportement par défaut du bloc.

Le menu de navigation

```
@media all and (max-width: 1024px)
{
   nav
   {
      width: auto;
      text-align: left;
   }

   nav ul
   {
      flex-direction: column;
   }

   nav li
   {
      padding-left: 4px;
   }

   nav a
   {
      font-size: 1.1em;
   }

   nav a:hover
   {
      border-bottom: 0;
   }
}
```

Nous voulons que le menu de navigation prenne moins de place sur les petites résolutions. Plutôt que de lui donner une dimension fixe, nous allons lui redonner sa dimension automatique flexible d'origine. Chaque élément du menu s'écrira en dessous du précédent : pour cela, nous demandons à ce que les éléments de la Flexbox soient organisés en colonne.

La bannière

```
@media all and (max-width: 1024px)
{
   #banniere_image
   {
      display: none;
   }
}
```

Pour retirer la bannière, rien de plus simple : nous utilisons la propriété `display` , à laquelle nous affectons la valeur `none` . Si la fenêtre est trop petite, nous préférons masquer complètement la bannière

En résumé

- Les media queries permettent de charger des styles CSS différents en fonction de certains paramètres.
- Les paramètres autorisés par les media queries sont nombreux : nombre de couleurs, résolution de l'écran, orientation... En pratique, on s'en sert surtout pour modifier l'apparence du site en fonction des différentes résolutions d'écran.
- On crée une media query avec la directive `@media` suivie du type d'écran et d'une ou plusieurs conditions (comme la largeur maximale d'écran). Le style CSS qui suit sera activé uniquement si les conditions sont remplies.
- Les navigateurs mobiles simulent une largeur d'écran : on appelle cela le *viewport*.
- On peut cibler les smartphones grâce à une règle basée sur le nombre réel de pixels affichés à l'écran : `max-device-width` .