

# Les media queries

Les media queries sont des règles CSS que l'on peut appliquer dans certaines conditions.

Concrètement, nous allons pouvoir dire « Si la résolution de l'écran du visiteur est inférieure à tant, alors applique les propriétés CSS suivantes ».

Grâce à ces règles nous pourrions modifier l'apparence (taille, couleur, position, etc..) de notre site selon la taille, la résolution ou encore l'orientation de l'écran. Les possibilités sont très nombreuses !

## Utilisation media queries

On peut utiliser les media queries de deux façons :

- en chargeant une feuille de style `.css` différente en fonction de la règle (ex : « Si la résolution est inférieure à 1280px 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 1280px de large, charge les propriétés CSS ci-dessous »).

## Chargement d'une feuille de style différente

On va ajouter un lien pour charger une feuille CSS différente selon la règle que l'on souhaite appliquer.

Exemple :

Si la résolution est inférieure à 1280px de large, charge le fichier `petite_resolution.css`

```
<link rel="stylesheet" media="screen and (max-width: 1280px)" href="petite_resolution.css" />
```

On ajoute un attribut `media`, dans lequel on va écrire la règle qui doit s'appliquer pour que le fichier soit chargé. On dit qu'on fait une « requête de media » (*media query* en anglais).

Ce qui donne :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <link rel="stylesheet" href="style.css" /> <!-- Pour tout le monde -->
6     <link rel="stylesheet" media="screen and (max-width: 1280px)" href="petite_resolution.css" /> <!-- Pour
ceux qui ont une résolution inférieure à 1280px -->
7     <title>Media queries</title>
8   </head>
```

## Chargement des règles directement dans la feuille de style

La méthode à privilégier consiste à écrire ces règles dans le même fichier CSS que d'habitude.

On écrit la règle dans le fichier `.css` comme ceci :

```
1 @media screen and (max-width: 1280px)
2 {
3     /* Rédigez vos propriétés CSS ici */
4 }
```

## Les principales règles

- `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'écran.

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

- `only`: « uniquement » ;

- **and**: « et » ;
- **not**: « non ».

## Media queries et navigateurs mobiles

Les écrans des smartphones sont beaucoup moins larges que nos écrans habituels (seulement quelques centaines de pixels de large). Pour s'adapter, les navigateurs mobiles affichent le site en « dézoomant », ce qui permet d'avoir un aperçu de l'ensemble de la page. La zone d'affichage simulée est appelée le **viewport** : c'est la largeur de la fenêtre du navigateur sur le mobile.

En CSS, avec les media queries, si nous ciblons l'écran avec `max-width` sur un mobile, celui-ci va comparer la largeur que nous indiquons avec celle de son viewport. Le problème, c'est que le viewport change selon le navigateur mobile utilisé !

Navigateur	Largeur du <i>viewport</i> par défaut
Opera Mobile	850 pixels
iPhone Safari	980 pixels
Android	800 pixels
Windows Phone	1024 pixels

Un iPhone se comporte comme si la fenêtre faisait 980 px de large, tandis qu'un Android se comporte comme si la fenêtre faisait 800 px !

Pour obtenir un rendu facile à lire, sans zoom, nous pouvons demander à ce que le viewport soit le même que la largeur de l'écran :

```
1 <meta name="viewport" content="width=device-width" />
```

Pour tester les différents affichages sur firefox, faites [Ctrl + Shift + m](#)