



Franck CELLIER

Formateur en Informatique

Technicien Qualité d'Organisme de Formation

<http://cellierfranck.alwaysdata.net>



MS LANGAGE & DEVELOPPEMENT POUR LE WEB

LE RESPONSIVE DESIGN VIA LES REQUETES MEDIA QUERIES

A. PRÉSENTATION

Définition du « responsive design »

Le « responsive design » désigne généralement l'ensemble des techniques permettant d'adapter les pages web à toutes les tailles d'écrans afin d'avoir un bon rendu.

La problématique du responsive design est apparue avec l'apparition des tablettes et des Smartphones pouvant se connecter à Internet : en effet, comment faire pour qu'un site s'affiche aussi bien sur un petit écran de téléphone portable que sur un ordinateur de bureau ?

Trois façons d'adapter son site sur tous les écrans

- Créer un site dédié pour chaque terminal différent (un site mobile, un site pour ordinateur, etc.) ;
- Créer une application mobile ;
- Utiliser les outils offerts par le HTML et le CSS et créer une version « responsive » du site.
-

B. LE VIEWPORT ET L'UTILISATION DES POURCENTAGES

Le « viewport » correspond à la taille de la fenêtre de l'écran utilisé par l'internaute.

Une première méthode, très simple, pour commencer à optimiser son site web pour un affichage sur mobile ou tablette est d'utiliser et de manipuler le viewport.

En HTML5, nous pouvons prendre « contrôle » de la taille de la fenêtre grâce à l'élément meta et aux attributs name et content :

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

Il est demandé à ce que les pages web s'adaptent à la taille de la fenêtre de l'écran de l'internaute, afin que le contenu s'adapte.

Le code `width=device-width` va faire en sorte que la page s'adapte à la taille de l'écran avec les éléments HTML prenant la place disponible.

Le code `initial-scale=1.0` définit le niveau de zoom initial lorsque la page est chargée par le navigateur.

C. UTILISATION DES VALEURS EN POURCENTAGE EN CSS

En plus de l'emploi de l'élément `meta`, il est judicieux d'utiliser dans le CSS d'utiliser des valeurs en pourcentages pour définir les tailles de certains éléments plutôt que des valeurs en pixels.

Cela aura pour effet de redimensionner les différents éléments des pages en même temps que la fenêtre va se rétrécir ou s'agrandir.

En utilisant cette méthode, ne perdez pas de vue que les tailles exprimées sont un pourcentage de la taille de l'élément parent des éléments que vous ciblez actuellement.

Définir une taille égale à 100% pour l'élément `body`, une largeur de 50% pour un div enfant direct du `body`, la taille du div sera toujours égale à 50% de celle du `body`.

Si maintenant il est défini une largeur égale à 50% pour les paragraphes à l'intérieur du div, les paragraphes occuperont 50% de la largeur du div.

D. LIMITES DE CES DEUX PREMIÈRES MÉTHODES

Le fait d'utiliser la `meta name="viewport"` et des tailles relatives dans le code CSS devrait déjà fournir une bonne base pour adapter le site web aux différentes tailles d'écran.

Cela reste toutefois insuffisant. Selon comment sont construites les pages, il sera peut-être nécessaire de réagencer totalement les blocs selon la taille de l'écran d'internaute.

Par exemple, pour un affichage sur mobile, on préférera souvent enlever du contenu et afficher nos différents éléments en longueur plutôt qu'en largeur.

La meilleure méthode pour faire tout cela va être d'utiliser une nouvelle fois la règle CSS `@` ainsi que ce qu'on appelle les "media queries".

E. PRÉSENTATION DES MEDIA QUERIES

Les media queries correspondent à une technique introduite en CSS3 et qui va se servir de la règle CSS `@media`.

Avec les media queries, il est possible d'appliquer différentes propriétés ou différentes valeurs à des éléments HTML selon la taille de l'écran et le type de media utilisé (écran, imprimante, etc).

Cela va permettre par exemple de ne pas afficher certains éléments pour des tailles d'écran trop petites ou de réorganiser les pages web grâce aux propriétés CSS.

F. UTILISATION DE @MEDIA EN CSS

Il est impératif de préciser une chose avec @media, soit le type de media pour lequel les déclarations CSS doivent s'appliquer.

Dans l'immense majorité des cas, on utilise @media screen, afin d'appliquer les propriétés à tous les medias dotés d'un écran (téléphone portable, tablette, ordinateur de bureau, etc.).

Cependant, il existe aussi @media print, afin de n'appliquer certaines règles que pour les imprimantes ou encore @media speech, pour les ordinateurs disposant d'une synthèse vocale.

Finalement, on peut également utiliser @media all pour appliquer des déclarations à tous les types de media.

Avec @media screen, on précise une taille ou un intervalle de tailles d'écran au sein duquel les propriétés vont s'appliquer.

Exemple : **LE CODE HTML**

```
<!DOCTYPE HTML>
<html>

  <head> <!-- Entête de la page WEB -->
    <meta charset="utf-8" />

    <!--Le initial-scale=1.0 signifie au navigateur qu'il doit initialiser les dimensions en se basant sur
    le pixel-ratio du système, et le user-scalable=yes autorise l'utilisateur à zoomer s'il le souhaite.
    Mettre cette dernière variable à "no" empêcherait tout zoom, ce qui est mauvais pour
    l'ergonomie et l'accessibilité de la page.-->
    <meta name="viewport" content="initial-scale=1.0, user-scalable=yes" />

    <title>LES MEDIA QUERIES ET LE RESPONSIVE DESIGN</title>
    <link rel="stylesheet" href="css\style.css" />
    <link rel="icon" type="objets/png" href="objets/favicon.png" />
    <link href="https://fonts.googleapis.com/css?family=Happy+Monkey" rel="stylesheet">

  </head>

  <body>

    <header>

      <h1>Mon site Responsive</h1>
      <p>
        
      </p>

    </header>

  </body>

</html>
```

Exemple : LE CSS

```
/* Pour les écrans ayant une taille inférieur à 681px la couleur de fond de la page sera orange */

@media screen and (max-width: 680px) {
  body {
    background-color: orange;
  }
}

/* Pour les écrans ayant une taille entre 681px et 1279px, la couleur de fond de la page sera bleue */

@media screen and (min-width: 681px) and (max-width: 1279px) {
  body {
    background-color: blue;
  }
}

/* Pour les écrans de taille supérieur à 1280px, la couleur de fond de la page sera verte */

@media screen and (min-width: 1280px) {
  body {
    background-color: green;
  }
}

img {
  max-width: 100%;
  height: auto;
}
```

G. UTILISER LES MEDIA QUERIES POUR CHANGER L'AGENCEMENT DES ÉLÉMENTS HTML

Les Media Queries permettent de masquer certains éléments ou modifier leur agencement selon la taille de l'écran.

Il faut utiliser les propriétés `display`, `position` avec des valeurs relatives en pourcentages de façon générale.

Exemple une page web composée de trois div côte à côte et d'un pied de page.

Exemple : LE CODE HTML

```
<!DOCTYPE HTML>
<html>
<!-- Site WEB de présentation du travail effectué entre le 17 et le 21 décembre 2018 - 1ère semaine -->
<!-- A l'attention des nouveaux arrivants -->

<head> <!-- Entête de la page WEB -->
  <meta charset="utf-8" />

  <!--Le initial-scale=1.0 signifie au navigateur qu'il doit initialiser les dimensions en se basant sur le pixel-
  ratio du système, et le user-scalable=yes autorise l'utilisateur à zoomer s'il le souhaite. Mettre cette
  dernière variable à "no" empêcherait tout zoom, ce qui est mauvais pour l'ergonomie et l'accessibilité de
  la page.-->
  <meta name="viewport" content="initial-scale=1.0, user-scalable=yes" />

  <title>LES MEDIA QUERIES ET LE RESPONSIVE DESIGN</title>
  <link rel="stylesheet" href="css/style.css" />
  <link rel="icon" type="image/png" href="objets/favicon.png" />
  <link href="https://fonts.googleapis.com/css?family=Happy+Monkey" rel="stylesheet">

</head>

<body>

  <header>

    <h1>Mon site Responsive</h1>
    <p>
      
    </p>

  </header>

  <main>
    <div class="block1">
      <p>Block N° 1</p>
    </div>

    <div class="block2">
      <p>Block N° 2</p>
    </div>

    <div class="block3">
      <p>Block N° 3</p>
    </div>

    <div class="pied">
      <p>Pied de page</p>
    </div>
  </main>

</body>

</html>
```

Exemple : LE CSS

```
body {
  margin: 0px;
  padding: 0px;
  font-family: 'Happy Monkey', cursive;
  color: darkblue;
  text-align: center;
}

.block1, .block2, .block3 {
  display: inline-block;
  width: 30%;
  height: 150px;
  border: 3px solid #AAA;
  margin: 1%;
}

.block1 {
  background-color: orange;
}

.block2 {
  background-color: #28B;
}

.block3 {
  background-color: #2B8;
}

.pied {
  width: 100%;
  height: 200px;
  margin-top: 20px;
  /*text-align: center;*/
  background-color: #CCC;
}

img {
  max-width: 50%;
  height: auto;
}

@media screen and (max-width: 780px) {
  .block1, .block2, .block3 {
    width: 100%;
    border-right: none;
    border-left: none;
    margin-left: 0px;
    margin-right: 0px;
  }

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

```
@media screen and (orientation:portrait) {  
  .block1, .block2, .block3 {  
    width: 100%;  
    margin-left: 10px;  
    margin-right: 10px;  
  }  
}
```

H. @MEDIA ET ORIENTATION

Des règles spécifiques doivent être mises en place selon l'orientation de l'écran (utile pour l'affichage sur tablette ou smartphone entre autres).

« orientation : landscape » ou « orientation : portrait » définissent des règles spécifiques pour les écrans tournés en mode « paysage » ou « portrait ».