

Responsive Web Design – Think Mobile First

I▶ Dans ce cours, nous allons aborder le Responsive Web Design et les fameux Média Queries du CSS3.

Qu'est ce que le Responsive Web Design ? Jusque vers la fin 2009-2010, la majorité des sites Web étaient maquetés avec un container centré (un div ou un tableau) dont la largeur maxi était de 1000px. On utilisait en général une image de fond que l'on plaçait dans le background du body en CSS. Cette image assez large occupait plus ou moins d'espace selon la résolution des écrans des internautes.

Mais on ne pensait pas en utilisant cette méthode de conception que de multiples terminaux Web feraient leur apparition ! Le tout 1er SmartPhone est apparu en 2007 et se nommait **iPhone**, puis vinrent tous les autres : les **SmartPhones Android, les WindowsPhone et les Tablettes tactiles** (dont l'IPad).

Les développeurs et les designers Web ont du adapter leur mise en page Web à tous ces différents terminaux. Ce n'était déjà pas simple avec les différents navigateurs présents sur le marché de l'Internet et les différentes versions de tous ces navigateurs, il fallait qu'un élément de plus apparaisse dans la problématique de conception de pages Web !

NB : ON peut consulter les statistiques d'utilisation des navigateurs ici : <http://gs.statcounter.com>

Beaucoup consultent le Web depuis leurs téléphones portables avec de petites tailles d'écran. Les sites conçus en 1000px sont automatiquement réduits pour que la page s'affiche intégralement. l'utilisateur peut donc zoomer avec ses doigts et naviguer dans telle ou telle partie du site. Mais il s'avère que ce n'est pas très pratique et la question se pose

alors de développer plusieurs interfaces du site, une pour le web classique, une pour telle catégorie de SmartPhones, une pour telle autre, etc .. On se rend vite compte que cette manière de faire engendrera beaucoup de travail en temps et en coût.

Heureusement, il y a une solution : le **Responsive Web Design**, un design, une conception de mise en page qui peut s'adapter de manière très fluide à toutes ces différentes largeurs d'écran !

Et le **RWD** s'appuie évidemment sur **HTML5** et **CSS3**, les révolutionnaires de la conception Web !

=> Quelques outils pour tester le responsive :

- Firefox
: <https://addons.mozilla.org/en-us/firefox/addon/firesizer/>
- Chrome
: <https://chrome.google.com/webstore/detail/window-resizer/kkelicaakdanhinjdeammilcgefongh>
- et : <http://lab.maltewassermann.com/viewport-resizer/>

► Principes du Responsive

Nous allons voir comment mettre en place un site adaptatif avec les méthodes du **Responsive Design** :

Les navigateurs des terminaux ont tendance à réduire automatiquement le site Internet conçu pour un affichage classique PC et le rendre complètement illisible.

=> **Exemple** : un site est généralement conçu pour un affichage en 1024 px minimum en largeur.

Le navigateur mobile va essayer de faire rentrer les 1024 px du site dans une taille d'écran qui peut aller de 380 px à 1000 px au plus pour les tablettes...

=> Les tailles d'écran des navigateurs :

Comprendre comment sont gérées de deux manières différentes n'est pas une mince affaire, il existe déjà 2 concepts quant à la gestion des tailles d'écrans :

- La surface réelle [**width**] : c'est la résolution de l'écran, le nombre précis de pixels de l'écran définis par le constructeur.
- La surface en **pixels CSS** [**device-width**] : c'est le nombre de pixels virtuels que pense avoir le terminal du mobile.

Avec la technologie **Retina d'Apple**, la densité des pixels est telle que la pixellisation même de très près est quasiment invisible.

Les Iphones 4 et 5 ont une résolution de 326 ppp (pixels par pouce) et 128 ppcm (pixels par cm), les Ipad ont une résolution de 264 ppp. ce qui fait que les résolutions en taille réelle **width** et **height** deviennent de plus en plus élevées !

Les tailles en **pixels CSS** (device-width et device-height) ramènent les dimensions de l'écran à quelque chose de plus réaliste et plus facile à implémenter dans le développement des pages HTML. Nous verrons plus tard comment l'implémenter avec les média-queries ...

=> *Voici un tableau de comparaison :*



On pourra trouver une liste plus complète http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density ou <http://mobitest.me/devices/>

=> Le concept du viewport des mobiles :

Le navigateur mobile a donc tendance à simuler une largeur d'écran plus grande, justement pour faire rentrer toute la page ! Mais la réduit fortement ...

Le meta-tag **viewport** inventé et propriétaire d'Apple, (totalement étranger à toute norme homologuée !) peut solutionner le problème.

Heureusement, les navigateurs mobiles l'ont adopté en grande majorité...

Le **viewport** par défaut des terminaux mobiles est de 980 px.

=> **Comment manipuler le meta-tag viewport :**

en utilisant ce tag dans l'entête du document HTML

```
<meta name='viewport' content='attributs'>
```

- Liste des attributs du viewport :

- **width** : largeur en pixel de la page
En fixant **width=device-width**, la largeur en pixel du navigateur s'adapte au nombre de pixels simulés par la largeur du terminal (device)
Soit imaginez un site avec un container principal de **1000 px**, visionné sur un mobile de **320px** de large (Iphone 3 4 et 5), on ne voit que les 320 premiers pixels ...
- **initial-scale** : niveau de zoom initial => **initial-scale='1'**
- **minimum-scale** : valeur la plus petite du zoom => **minimum-scale='0.5'**
- **maximum-scale** : valeur la plus grande du zoom => **maximum-scale='1.5'**
- **user-scalable** : autorisation ou pas de zoomer => **user-scalable='yes' | 'no'**



Et sur une tablette de **600px** de large :



Il faut maintenant grâce aux media-queries, redimensionner ou faire disparaître certaines zones :

La détection se fera avec la largeur de l'écran :

```
/*————— Iphone Max 380px —————*/  
@media only screen and (max-width:320px) and  
(orientation:portrait)  
{  
section {width:90%; margin-top:25%; float:left;}  
aside { display:none;}  
#container { width:95%;padding:1%;}  
nav li { clear:left; margin:0;}  
nav li:first-child {margin:0;}  
header { height: auto; line-height: normal; padding: 0; }  
}
```



Michel BOCCIOLESI