

Responsive design

Le responsive design est la capacité d'un design de site à s'adapter à n'importe quel terminal, à ses capacités d'affichage (la dimension de l'écran, sa résolution par exemple). Les terminaux se diversifient, l'arrivée massive des smartphones et l'explosion de leur utilisation n'est que le début de cette révolution de la conception web.

Sujet de débat à la mode il y a quelques années qui est devenu un sujet indispensable mais qui pose quelques questions tant par l'aspect technique (beaucoup de problèmes ont été déjà résolus), mais aussi par l'aspect du mode de conception classique que certains remettent en cause.

La conception

Le web de Mamie

Jusqu'à maintenant la conception graphique d'un site se résumait à penser un seul design, un seul format d'écran, à résoudre quelques petits problèmes d'interaction et d'accessibilité.

Le *webdesigner* faisait quelques croquis, pensait plus précisément à deux ou trois problématiques que le site allait lui poser et se lançait rapidement (pour la plupart) sur *la créa* (un ou plusieurs documents Photoshop censés représenter le design final d'un site).

Le web était à une seule dimension, quasiment tout le web était conçu pour du 960 pixels de large, on a même pensé à utiliser des systèmes de grilles pour accélérer la production de ces sites.

Le web d'aujourd'hui

Le web est aujourd'hui à plusieurs dimensions et ne consomme plus forcément assis devant un quinze pouces avec une bonne connexion ADSL.

Le concepteur, désigner web doit s'adapter à ces nouvelles pratiques pour offrir le meilleur service aux utilisateurs. Il doit imaginer un web qui s'adapte au mode de consommation de ce média et penser des supports, des outils, du web qui s'ajustent à la situation de l'utilisateur.

Une créa par taille d'écran ?

Le modèle de conception utilisé jusqu'à maintenant ne marche plus.

Comment imaginer, présenter et surtout réaliser autant de *créa* qu'il y a de formats d'écran possibles ? (Et je n'évoque même pas le client capricieux qui fait reprendre 4 ou 5 fois toute une série d'écrans).

Pour mieux travailler, de manière plus intelligente et aussi peut-être intelligible, il faut sans doute essayer d'imaginer de nouvelles manières de créer et de faire comprendre à un client ce que l'on souhaite créer et ce que l'on imagine.

Un nouveau mode de conception, de nouveaux outils ?

La mise en cause de la manière de travailler et les outils qui y participent.

Jusqu'à maintenant, l'outil de choix a été Photoshop (parfois Illustrator, et quelque-fois même inDesign). C'est un moyen permettant d'apercevoir un résultat et de pouvoir imprimer un support au format A3 présentable à un décideur.

Cette étape de maquette est parfois le résultat d'une réflexion effectuée en amont, dans le meilleur des cas.

Dans tous les cas ce travail peut être long et le résultat obtenu n'est pas forcément à la mesure des attentes car, simplement, une page A3 n'est pas un écran et ne permet pas d'acquérir une idée valable de ce qu'un site peut être, de ses couleurs, de la taille de la typographie... Le papier donne une impression de site, tout au plus il donne un aperçu.

Mais la papier est quelque chose qui permet de travailler rapidement, de concevoir avec quelques traits de crayons, c'est un outil rapide qui permet d'avancer rapidement dans les idées par itération.

À partir de ces croquis plusieurs choix sont possibles :

1. On passe directement à un maquettage *réel*, une page HTML et des CSS, cela implique de maîtriser ces outils, d'être peut-être aussi expérimenté.
2. On passe à la conception dans Photoshop. (Sans oublier les déclinaisons possibles sur les différentes dimensions d'écran...)

Ne jetez pas Photoshop (ou Illustrator), pas tout de suite, il est toujours utilisable, mais peut-être plus tardivement dans la réflexion.

- <http://37signals.com/svn/posts/1061-why-we-skipphotoshop>

Et le contenu ?

Un site se conçoit par son contenu, c'est un point essentiel et le fait de le rendre *responsive* rend le sujet encore plus sensible.

Le contenu doit s'adapter au mode de consultation, il faut aussi penser à ce qu'un utilisateur peut rechercher dans un site lorsqu'il le consulte avec un mobile ou avec un iPad.

Qu'est-ce qu'on peut lui offrir de plus ou de moins ?

Quelques idées sur le sujet : * Un mobile peut être localisé par des moyens GPS; * Un mobile permet de faire des photos, des vidéos sur un lieu donné; * Une tablette peut tourner, peut utiliser une caméra, peut être localisé...

Les modes de consultation évoluent mais les outils aussi, il faut penser à ça dans cette conception.

De la technique

Média queries

Les *media queries* sont un moyen d'adapter le design d'un site au terminal qui s'y connecte.

Jusqu'à maintenant

En CSS2, il était possible d'indiquer des feuilles de styles différentes pour un écran et pour une imprimante par exemple, mais aussi pour de nombreux autres terminaux.

```
<link rel="stylesheet" media="screen" href="style_ecran.css" />
<link rel="stylesheet" media="print" href="style_imprime.css" />
```

L'attribut *media* permet de spécifier le support de destination des feuilles de styles, dans l'exemple précédent, la première feuille de style sera utilisé pour le rendu sur un écran et la deuxième feuille de style sera utilisée lorsque le document courant sera imprimé.

Il est possible de spécifier les supports suivants :

- screen : Écrans
- handheld : Périphériques mobiles ou de petite taille, ignorée la plupart du temps par les mobiles...
- print : Impression
- speech : Synthèses vocales
- braille : Plages braille
- embossed : Imprimantes braille
- projection : Projecteurs (ou présentations avec slides)
- tty : Terminal/police à pas fixe
- tv : Écran télé
- all : Tous les supports

Ces *media* sont ici des attributs mais peuvent être aussi utilisés directement dans la feuille de style :

```
@media screen {  
    /* Ici les styles pour les écrans */  
}  
@media print {  
    /* Ici les styles imprimés */  
}
```

Dans l'exemple précédent il est ainsi possible de fournir dans la même feuille de style, un style pour l'écran et un autre pour les imprimantes.

À noter que si l'exemple précédent est utilisé il est nécessaire de ne pas spécifier de *media* dans la balise important le fichier.

```
<link rel="stylesheet" href="style_complet.css" />
```

Aujourd'hui

Les media queries ont évolué et permettent maintenant une sélection fine des terminaux et de leur fournir des feuilles de styles adaptées de manière très précise, selon leur dimension ou voire même leurs capacités techniques.

Pour cela la syntaxe utilisée permet d'indiquer un ou plusieurs critères de filtres, ces critères sont les suivants :

- *only*, qui signifie *seulement* ou *uniquement*;
- *and*, pour ajouter, ce qui signifie *et*;
- *not* pour exclure, *non*.

Le *ou* n'existe pas, c'est le comportement par défaut d'une *media query*. Les conditions sont signalées les unes après les autres et la règle est appliquée dès qu'un des condition est rencontrée.

De manière générale, les *query* sont composées d'un media (screen, print...) et d'une expression, souvent la dimension de l'écran.

(À noter que ce que nous allons voir ensuite peut-être utilisé dans la valeur d'attribut *media* de la feuille de style importée via la balise *link* ou directement dans la feuille de style.)

Exemple

```
@media screen and (max-width: 500px) {  
    /* Des styles pour les écrans inférieurs à 500px */  
}
```

Qui peut aussi se manipuler de la manière suivante :

```
<link rel="stylesheet" media="screen and (max-width: 500px)"  
href="ecran-mini.css" />
```

Ces règles peuvent se cumuler :

```
@media screen and (min-width: 200px) and (max-width: 400px) { /* Des styles
```

pour les écrans avec des fenêtres de dimension 200px à 400 px */ }

Il existe de nombreuses expressions permettant de filtrer les terminaux selon leur capacités de couleur (*color*), leur format (*aspect-ratio*) pour des écrans 4/3 ou 16/9 par exemple, la hauteur et la largeur (*height* et *width*), l'orientation *portrait* ou *paysage*, la résolution du périphérique (pour les écrans Retina par exemple), le type d'écran utilisé (LCD ou autre technologie) avec l'expressin gris.

Les dimensions utilisent les unités de dimensions classiques des CSS (pixels, em, %...) et peuvent être préfixées de *min-* et *max-* pour indiquer des valeurs minimum et maximum. Les résolutions sont à indiquer en DPI (dot per inch) ou DPCM (dot per centimeter).

À noter une différence particulière sur les capacités suivantes : *width* et *height* sont les dimensions d'affichage d'un site (le fenêtre dans laquelle il est affiché) tandis que *device-width* et *device-height* permettent de connaître la dimensions d'affichage maximale d'un périphérique (la dimension réelle du matériel).

Le support des navigateurs

La plupart des smartphones et des navigateurs récents reconnaissent ces queries et ce sont des outils utilisables dès maintenant. Seul Internet Explorer se retrouve en retard sur le sujet. Mais il existe des solutions, la plus utilisée actuellement étant *Respond.js* : <https://github.com/scottjehl/Respond>.

Le problème de la résolution

Les écrans jusqu'à maintenant n'avait qu'une résolution de 72 DPI, avec l'arrivée des smartphones et des écrans *retina* cette résolution a évolué.

Pour résoudre ce problème les fabricants et les concepteurs de logiciels de navigation on inventé un nouveau tag, qui ne fait pas pour le moment partie des standards.

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
```

Votre design qui fait théoriquement 320 pixels sera affiché sur des écrans dont les dimensions réelles font jusqu'à 780 pixels.

Cette balise permet d'indiquer la manière dont le navigateur doit boomer ou non sur le contenu que vous avez rendu *responsive*.

Le *viewport* est la fenêtre virtuelle qui permet d'afficher un site.

Quelques exemples de valeurs possibles pour cet balise

```
<meta name="viewport" content="width=320">
```

Permet de signaler un affichage stricte à 320 pixels de large.

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

Permet de signaler un design dit *fluide*, un design qui s'adapte continuellement à la taille d'écran de l'utilisateur.

Ressources

- <http://css-tricks.com/css-media-queries/>
- <http://www.alsacreations.com/article/lire/930-css3-media-queries.html>
- <https://github.com/scottjehl/Respond>
- <http://webdesign.tutsplus.com/tutorials/htmlcss-tutorials/quick-tip-dont-forget-the-viewport-meta-tag/>