Mobile first et responsive design

Adapter le design au support

TOUT EST UNE QUESTION DE SUPPORT

Dès l'origine des technologies de la communication et de l'information, l'humain à chercher à adapter la forme de son discour à l'environnement dans lequel il est reçu. Des plaques Mésopotamiennes aux objet connectés il faut depuis toujours trouver les méthodes et les techniques qui permettent à une information d'atteindre les publiques qu'elle cible. L'impact de la multiplication des supports est encore plus fort dans l'industrie du numérique mais les expériences passées permettent de mettre en place des techniques qui pourront répondre aux futures évolution du responsive design.

PLAQUE EN TERRE CUITE

Mésopotamie 3 300 avant J.-C

OBJET CONNECTÉ

Définition du terme "Internet des objets" en uin 2012

ADTAPTER LE WEB AU SUPPORT QUI LE LIT

Il existe plusieurs moyens de connaître la taille de l'écran d'un utilisateur dans les différents langages Web. Le Reponsive Design est une technique qui permet d'adapter les styles d'une page Web pour qu'elle est un impact optimisé sur ses différents supports de lecture, c'est pourquoi il doit être défini en CSS plutôt qu'en JavaScript par exemple. Il faut alors faire appel aux requêtes Media Queries qui permettent de définir des propriétés différentes selon la taille de l'écran (ou de la fenêtre).

Media Queries dans la balise <head>

Les feuilles de style seront chargées uniquement si la largeur de l'écran respecte les conditions définies dans la balise media="..."

Cette technique implique l'écriture de toutes les propriétés CSS dans plusieurs fichiers CSS

```
<link rel="stylesheet"
      media="screen and (min-width: 980px)"
      href="styleLarge.css"/>

<link rel="stylesheet"
      media="all and (orientation: portrait)"
      href="stylePortraint.css"/>
```

Media Queries dans le fichier CSS

Toutes les tailles sont définies dans le même fichier CSS, il faut intégrer le principe du "Mobile First" en commençant par définir les styles des largeurs les plus petites.

Avec cette technique il est possible de conserver des propriétés communes aux différents formats pour ne redéfinir que certaines propriétés CSS

```
@media screen and (min-width: 980px){
      p {
          font-size: 12px;
          color : red;
      }
}

@media all and (orientation: portrait){
      p {
          font-size: 14px;
      }
}
```

LES DIFFERENTS TYPES DE MEDIA

La règle CSS @media permet de spécifier le support sur lequel doivent s'appliquer des propriétés CSS. Initialement, la règle CSS @media permettait de définir une feuille de style spécifique pour l'impression avec le media print et une autre pour les écrans d'ordinateur avec media screen. Avec l'apparition de nouveaux lecteurs de page Web, la règle CSS @media a intégré de nouveaux types de média pour permettre une plus grandes optimisations.

@media all

Permet de spécifier des propriétés pour tous les types de media

@media print

Permet de spécifier des propriétés destinées à l'impression

@media screen

Permet de spécifier des propriétés destinées aux écrans d'ordinateurs

```
@media all {
   body {
      font-family: 'Arial', sans-serif;
   }
}

@media print {
   body {
      font-size: 14pt
   }
}

@media screen {
   body {
      font-size: 12px
   }
}
```

D'autres types de media permettent par exemple d'optimiser une page Web pour les supports en braille ou pour l'affiche sur des télévisions ou des projecteurs.

CONFIGURER UNE PAGE WEB RESPONSIVE

Associé à la règle @media CSS, la balise HTML meta viewport permet de contrôler la mise en page sur les navigateurs mobiles. Cette balise est essentielle pour maîtrise l'affichage des CSS sur un smartphone car elle permet d'initialiser et de bloquer le zoom. En effet, contrairement à un ordinateur, un smartphone permet de zoomer dans l'écran, il est donc primordial de maîtriser le niveau de zoom initial pour pouvoir configurer efficacement les CSS.

META VIEWPORT
Balise à placer dans la balise <head>..</head>

CONTENT
Permet d'effectuer la configuration du viewport

WIDTH
Largeur de la fenêtre, peut être défini en pixel

HEIGHT
hauteur de la fenêtre, peut être défini en pixel

INITIAL-SCALE
Niveau de zoom de base

MAXIMUM-SCALE
Niveau de zoom maximal

USER-SCALABLE
Possibilité ou non de zoomer dans la page Web

```
<meta name="viewport"
content="
width=device-width,
height=device-height,
initial-scale=1,
maximum-scale=3,
user-scalable=yes" >
```

La configuration de base du viewport pour le reponsive design est width=device-width et initial-scale=1.0

UNITES DE MESURE RELATIVE : EM, REM ET %

Il existe différentes unités de mesure en CSS dites absolues ou relatives. Les unités absolues comme le pixel (px) ou le point (pt) ne seront pas modifiées selon les possibilités d'affichage et sont donc utilisées pour maitriser le rendu d'impression. En Responsive Design il est essentiel d'utiliser des unités de mesure relatives comme le cadratin (em, rem) ou le pourcentage (%) car elles ne sont pas fixes mais proportionnelles à la taille du texte du parent.

Taille par défaut

La taille par défaut des navigateurs est de 16px, pour lui donner une valeur de 10px nous le réduisons à 62.5%

```
html {
    font-size: 62.5%;
}
```

Taille des balises <p>

L'unité rem est proportionnelle à la balise HTML, les balises <p> font donc 12px (10 x 1,2 = 12)

```
p {
    font-size: 1.2rem;
}
```

Taille des balises <h1>

L'unité rem est proportionnelle à la balise HTML, les balises <h1> font donc 22px (10 x 2,2 = 22)

```
h1{
    font-size: 2.2rem;
}
```

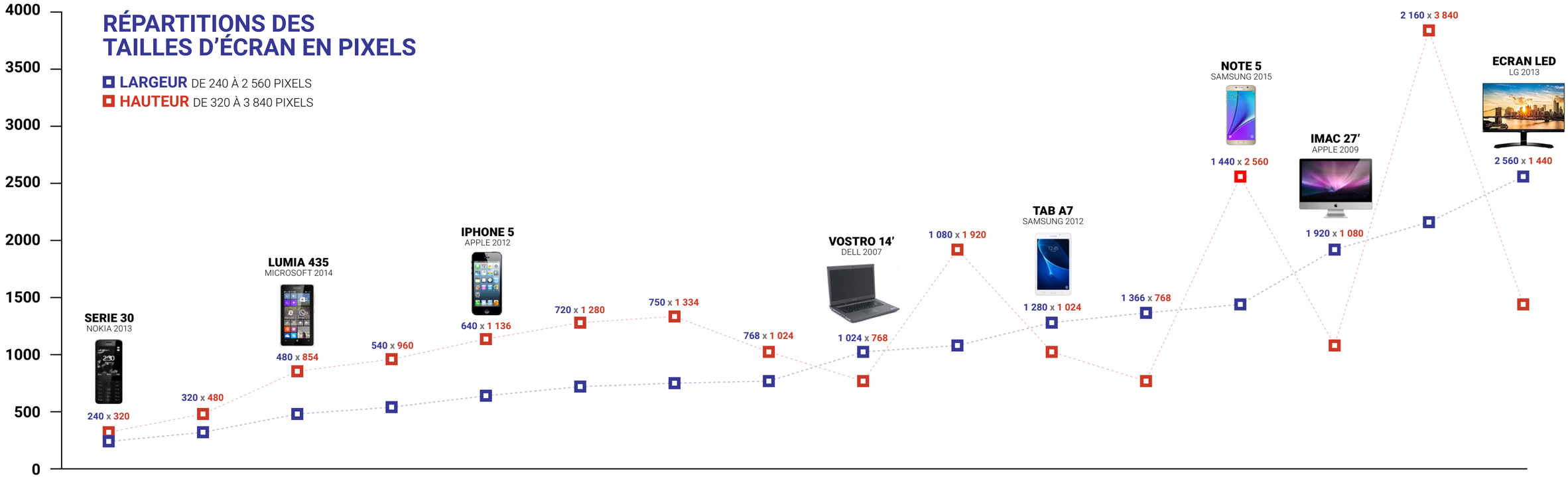
**Taille des balises **

L'unité em est proportionnelle à la balise parent, les balises font donc 11px (22 x 0,5 = 11)

```
h1 span{
    font-size: 0.5em;
}
```

DONNE MOI TON BREACKPOINT JE TE DIRAI QUI TU ES

Bien que le modèle ou la marque du support importent peu en reponsive design, leur largeur elle est primordiale pour savoir comment adapter un design. L'étude de DeviceAtlas de 2016 dresse néanmoins une liste des formats les plus utilisés que nous pouvons prendre en référence pour définir les breackpoints à intégrer dans une feuille de style CSS. Au delà de la simple adaptation du design dans les différentes tailles d'écran, il est primordial de concentrer sont expertise en responsive design sur l'analyse du contenu d'un document et la façon le plus efficace pour l'utilisateur de "consommer" l'information qu'il présente.



Les données présentées sont les caractèristiques constructeur mais pas les tailles physiques des supports

PENSER PETIT POUR VOIR GRAND

Si la course aux écrans les plus grands n'est pas prête de s'arréter, il est néanmoins possible de prévoir des designs qui s'adapterons progressivement.. Nous ne pouvons donc pas prévoir la largeur maximal d'affichage d'une page Web mais nous pouvons en rechanche décider de designer en priorité les formats les plus petits. Les avantages de cette méthode et d'apporter un contenu optimisé pour les supports mobiles qui deviennent aujourd'hui les supports les plus utilisés, de respecter les recommandation de Google qui indéxe en priorité les site en "Mobile First" mais également d'imposer une démarche de "User Centric Content Strategy". Bien que chaque contenu soit différent il est possible de définir des formats de base pour configurer les breackpoints d'un fichier CSS.

Mobile First

Le CSS initiale est prévu pour les écrans les plus petits

```css
main{ max-width: 35rem; margin: auto; }
```

Les principaux breackpoints

Au fil des années plusieurs tailles et techniques standards ont été défini pour optimiser l'affichage d'une page Web.

Le principe est de toujours commencer par la plus petite taille d'affichage et de modifier certaines propriété CSS des balises qu'il faut adapter sur les tailles les plus grandes.

Il n'y à aucune limite à l'utilisation des breackpoints, il est donc possible d'apater un design à tous les supports possibles et imaginables.

```css
@media screen and (min-width: 480px) {
  main{ max-width: 73rem; }
}

@media screen and (min-width: 736px) {
  main{ max-width: 95rem; }
}

@media screen and (min-width: 980px) {
  main{ max-width: 120rem; }
}
```

Encore plus petit, toujours plus grand

De la smartwatch à l'écran 4K, un breackpoint peremttra d'adapter n'importe quel design.

```css
@media screen and (max-width: 240px) {...}
@media screen and (min-width: 2560px) {...}
```