

# 10.Responsive Web Design

2 décembre 2024

## Développement web il3

### Responsive Web Design

#### *Conception de sites web adaptatifs*

HE-Arc (DGR) 2024

### Site adaptatif ?

- Surfer depuis : PC, mobiles, tablettes, tv, ...
- UX : navigation avec un minimum de zoom, pan, scroll
- S'adapter aux spécificités des appareils
  - orientation
  - taille caractères
  - modes d'interaction (ex: tactile, hover, ...)
  - ...
- 1 seul site à gérer : m.cool.com ni de cool.com/mobile
- Le même contenu pour tous
- Souvent basé sur la largeur de l'écran
- CSS3
- *Responsive Web Design* (1), Exemple<sup>1</sup>

---

<sup>1</sup><https://alistapart.github.io/code-samples/responsive-web-design/ex/ex-site-FINAL.html>

## Techniques

- Media queries : Taille de l'écran (ou sortie)
- UNITES RELATIVES
- Fonts : Dimensions en em
- Fluid Grids : Disposition et taille des éléments en %
- Flexible images (and media) : Taille des médias en %
- Autres considérations
  - Adaptatif avec grilles fixes<sup>2</sup>
  - Performances<sup>3</sup> : tps chargement, requêtes inutiles, ...
  - Transitions CSS
  - ...
- Exemple<sup>4</sup>

## Media Queries<sup>5</sup>

- media type : all, screen, print, tv, braille, handheld, ...

```
<link rel="stylesheet" type="text/css" href="style.css" media="screen" />
<link rel="stylesheet" type="text/css" href="printfriendly.css" media="print" />
```

- media query

1. dans élément link

```
<link rel="stylesheet" type="text/css" media="screen and (max-device-width: 800px)"
href="style800.css" />
```

2. dans une feuille de style

```
#nav {float: right;}
#nav ul {list-style: none;}
@media screen and (min-width: 400px) and (orientation: portrait)
{
    #nav li {float: right; margin: 0 0 0 .5em; border:1px solid #000000;}
}
@media screen and (min-width: 800px)
```

---

<sup>2</sup><https://blog.atolcd.com/adaptive-design-versus-responsive-design/>

<sup>3</sup><https://www.netguru.com/blog/improve-web-app-performace>

<sup>4</sup><https://webdesignerwall.com/tutorials/responsive-design-in-3-steps>

<sup>5</sup>[https://developer.mozilla.org/fr/docs/CSS/Media\\_queries](https://developer.mozilla.org/fr/docs/CSS/Media_queries)

```

{
    #nav { width: 200px; }
    #nav li {float: left; margin: 0 0 0 .5em; border: none;}
}

```

### 3. Règle CSS import

```
@import url(style600min.css) screen and (min-width: 600px);
```

## Media Queries<sup>6</sup>

width, height, device-width, device-height, orientation, aspect-ratio, device-aspect-ratio, color, color-index, monochrome, resolution, scan, grid

- Règles CSS selon medium (souvent min-, max-width)
- Opérateurs : only, not, and
- Au moins 3 layouts : mobile, tablet, desktop
- Resolution breakpoints : 320, 480, 600, 768, 1024, 1200px
- Souvent ces règles sont utilisées pour :
  - agrandir la taille du texte
  - agrandir la taille des zones cliquables (utilisation au doigt)
  - faire passer le contenu sur une seule colonne
  - masquer ou afficher des éléments spécifiques
  - ajuster les dimensions et marges
- Attention à l'ordre de chargement

## Meta Tag viewport<sup>7</sup>

- Introduit pour iPhone<sup>8</sup>, puis standard de fait
  - Par défaut, l'affichage est réduit (980px affichés sur écran 320px)
  - Meta tag viewport permet de changer ce ratio
- viewport : display area ≠ rendering surface
- Mobiles : viewport > écran (iphone 3 : v:980 / é:320)

<sup>6</sup>[https://developer.mozilla.org/fr/docs/CSS/Media\\_queries](https://developer.mozilla.org/fr/docs/CSS/Media_queries)

<sup>7</sup><https://lukeplant.me.uk/blog/posts/you-can-stop-using-user-scalable-no-and-maximum-scale-1-in-viewport-meta-tags-now/>

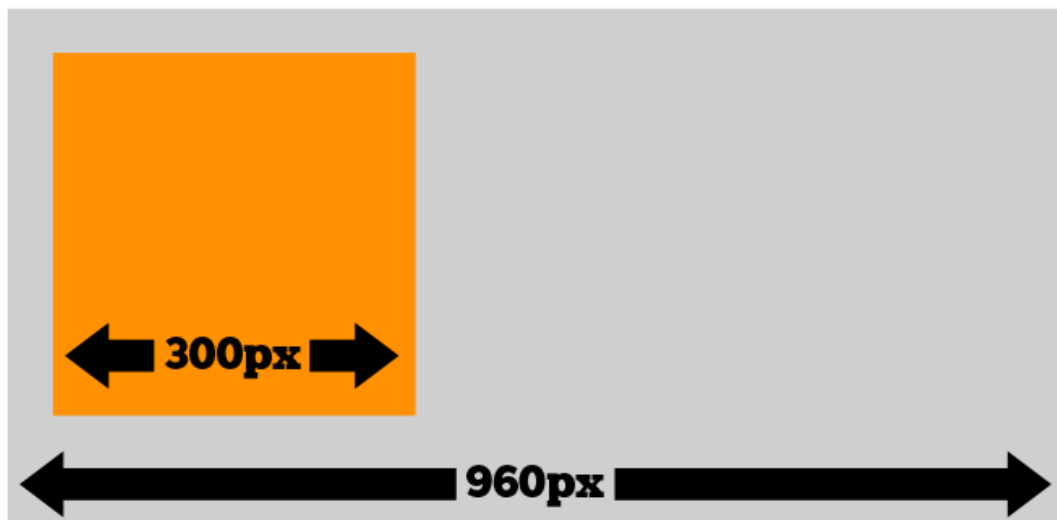
<sup>8</sup><https://developer.apple.com/library/content/documentation/AppleApplications/Reference/SafariWebContent/UsingtheViewport/UsingtheViewport.html>

- Media queries comparent au viewport
- Mise à jour du viewport à la taille de l'écran nécessaire :

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

**Résultat = Cible / Contexte**

**target / context = result**



**300px / 960px = 31.25%**

Figure 1: Target / Context

## Texte

- Texte redimensionnable : em pour les polices
- 1rem : font-size:100% dans élément html (16px/défaut)
- 1em : font-size:100% dans élément courant
- Conversion px -> em : result = target/context

- ne pas arrondir
- laisser le rapport en commentaire

## Fluid Grids

- Layout basé sur une grille en pixel
- Conversion px -> % : **result = target/context**
  - ne pas arrondir
  - laisser le rapport en commentaire
- Appliqué au style des divs :

width, margin, padding, background-position, ...

## Responsive Images

- Nouveautés de HTML 5
  - Eléments <picture>, <source>
  - Attributs srcset et sizes
- Besoins<sup>9</sup>
  - Écrans haute densité : srcset
  - Taille variable : srcset et sizes
  - Substitution<sup>10</sup> et modification layout : <picture>, <source>
  - Choix formats de fichiers <picture>
- Différences<sup>11</sup> entre <picture> et srcset
- Exemple<sup>12</sup> en français

## Flexible images

- Eviter qu'une image ne déborde de son conteneur
  - La réduire

```
img, embed, object, video{ max-width: 100%; }
```

<sup>9</sup><https://www.smashingmagazine.com/2014/05/14/responsive-images-done-right-guide-picture-srcset/>

<sup>10</sup><https://ericportis.com/etc/smashing-mag-picture-examples/art-direction.html>

<sup>11</sup><https://css-tricks.com/responsive-images-youre-just-changing-resolutions-use-srcset/>

<sup>12</sup><https://www.hteumeuleu.fr/attribut-srcset-images-responsive/>

- La découper

```
.feature { overflow: hidden; }
.feature img { display: block; max-width: auto; }
```

- Pas de standard pour servir différentes tailles de fichier
- Quelques idées recensées par *Smashing Magazine* (2)

## Outils

- Tester
  - Simulateur mobile des devtools, largeur browser
  - bookmarklet<sup>13</sup> pour afficher les media queries
  - mais surtout tester sur mobile
- Et Après ? MOBILE FIRST<sup>14</sup>, OFFLINE FIRST<sup>15</sup>, PWA<sup>16</sup>
- framework ou from scratch ?

## Références

- Exemples
  - Livre<sup>17</sup> d'Ethan Marcotte
  - thenextweb<sup>18</sup>
  - designshack<sup>19</sup>
- Plus loin...
  - Généralités<sup>20</sup>
  - viewport et media queries<sup>21</sup>
  - D'autres techniques, liste de Smashing magazine (2)
  - Améliorer la performance<sup>22</sup>, Awesome WPO<sup>23</sup>
  - Making sites more responsive, responsibly<sup>24</sup>

<sup>13</sup><https://sparkbox.github.io/mediaQueryBookmarklet/>

<sup>14</sup>[https://www.lukew.com/resources/mobile\\_first.asp](https://www.lukew.com/resources/mobile_first.asp)

<sup>15</sup><https://offlinefirst.org/>

<sup>16</sup><https://developers.google.com/web/progressive-web-apps/>

<sup>17</sup><https://ethanmarcotte.com/books/responsive-web-design/full/>

<sup>18</sup><https://thenextweb.com/dd/2013/01/13/30-new-inspiring-responsive-design-websites/>

<sup>19</sup><https://designshack.net/articles/css/20-amazing-examples-of-using-media-queries-for-responsive-web-design/>

<sup>20</sup><https://johnpolacek.github.io/scrolldeck.js/decks/responsive/>

<sup>21</sup>[https://www.quirksmode.org/blog/archives/2010/09/combining\\_meta.html](https://www.quirksmode.org/blog/archives/2010/09/combining_meta.html)

<sup>22</sup><https://csswizardry.com/2013/01/front-end-performance-for-web-designers-and-front-end-developers/>

<sup>23</sup><https://github.com/davidsonfelli/awesome-wpo>

<sup>24</sup><https://24ways.org/2014/making-sites-more-responsive-responsibly/>

## Pratique

- Tester les exemples sur un mobile
- Comprendre les sources
- Présentation adaptative de votre équipe de projet

## Sources

1. MARCOTTE, Ethan. Responsive Web Design. [en ligne]. 25 mai 2010. [Consulté le 6 novembre 2017]. Disponible à l'adresse : <https://alistapart.com/article/responsive-web-design>
2. THE SMASHING EDITORIAL. Responsive Web Design Techniques, Tools and Design Strategies. [en ligne]. 22 juillet 2011. [Consulté le 6 novembre 2017]. Disponible à l'adresse : <https://www.smashingmagazine.com/2011/07/responsive-web-design-techniques-tools-and-design-strategies/>