

Formation CESI : Création de site Web

Daniel VERMONDEN
Janvier / février 2020

Jour 3 : Mise en page
responsive

Compléments jours précédents

Ajouter lien site (github.io) sur readme.md du dépôt !

Les librairies d'icônes : l'exemple de Font Awesome

<https://fontawesome.com/>

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>title</title>
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.4/css/all.css" integrity="sha384-JdIuD5NtQ1y0UzSGmK1j+AMCJQ0B+PjEJLjZJGZGZqGJZ3HsWZJZk" crossorigin="anonymous">
  </head>
  <body>
    <h1>Test Font Awesome</h1>
    <a href="https://fontawesome.com">https://fontawesome.com</a>
    <i class="far fa-address-card"></i>
  </body>
</html>
```

Revue des projets

Les polices alternatives habituelles

<https://websitesetup.org/web-safe-fonts-html-css/>
https://www.w3schools.com/cssref/css_websafe_fonts.asp

Objectifs du cours

- Maîtriser les spécificités du "mobile"
- Comprendre la notion de RWD
- Maîtriser les outils du navigateur pour analyser le RWD
- Maîtriser les techniques de mise en pages CSS pour RWD
- Comprendre les problématiques spécifiques associées aux images dans le cadre du RWD

Déroulement du cours

- Réfléchir à la notion de "mobile" et aux spécificités du "mobile"
- La notion de RWD (origine, enjeux) et des notions associées
- Identifier les adaptations du RWD
- Mettre en oeuvre le RWD en CSS
- La problématique des images en RWD

Le mobile

La notion de "plateforme mobile"

Essayons de préciser ensemble cette notion : quelles caractéristiques ? quelles spécificités ?

Mobile ?



Les spécificités du mobile

Spécificités du mobile : taille de l'écran

Les sites et applications mobiles doivent fonctionner sur une grande diversité de types d'écran et de modes d'affichages.

Spécificités du mobile : taille de l'écran

Des tailles d'écran très différentes : globalement, en France, en France sur mobile

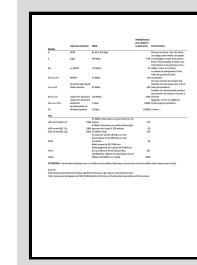
Spécificités du mobile : taille de l'écran

- Le format d'affichage peut varier sur le même appareil avec la rotation d'écran !

Spécificités du mobile : puissance et mémoire RAM

- Entrée de gamme actuel en France : 1 GHz, 1 Go de RAM
- Samsung GNote 2 : 1,6Ghz et 2 GB de RAM
- Macbook (Notebook) : 2,7GHz et 8 GB de RAM

Spécificités du mobile : réseau : Gamme de débits



Spécificités du mobile : réseau : la question du déplacement

La qualité du réseau n'est pas constante !

Spécificités du mobile : des API spécifiques

- Accéléromètre
- GPS
- Carnet de contacts
- Notifications
- etc.

Spécificités du mobile : UI/interactions : touch screen et clavier virtuel

Spécificités du mobile : gardez en tête ces éléments !

- Taille de l'écran très variable
- Puissance et mémoire RAM peuvent être limités (mais pas forcément)
- Réseau : rapidité pas assurée, déconnexions possibles (car mobilité)
- API spécifiques disponibles : accéléromètre, GPS, carnet de contacts, les notifications
- UI/interactions : touch screen et clavier virtuel

Historique du mobile et importance de l'utilisation aujourd'hui

Historique : 2007...



Historique : année 2007

- La révolution du smartphone avec l'iPhone
- Lancement d'Android par Google

Quelques chiffres du marché et de la consultation

desktop-mobile-tablet

Des explications concernant les sources

"When compiling statistics, larger sample sizes generally lead to more reliable results. Our stats are based on over 10 billion page views per month recorded across more than 2 million websites."

Définition du mobile et de la tablette

"MOBILE: How do you define a mobile device? We define a mobile device as a pocket-sized computing device, typically having a display screen with touch input or a miniature keyboard."

"TABLET: How do you define a tablet? We define a tablet as a portable computing device, larger than a mobile device, with a touchscreen interface."

Comment identifier que la page est vue sur un mobile ?

"With each page view a useragent string is sent which allows us to determine the browser and operating system used and also to establish if the page view came from a mobile device."

(Petit) challenge

Retrouvez cette information dans le navigateur ?

Exemple sur Chrome



The screenshot shows the Network tab in the Chrome DevTools. A single request is listed:

- Request URL: http://gs.statcounter.com/images/box_wide.png
- Request Method: GET
- Status Code: 200 OK
- Remote Address: 104.20.2.47:80
- Referer Policy: no-referrer-when-downgrade

Below the request, the Response Headers section is expanded, showing:

- Accept: image/webp,image/apng,image/*,*;q=0.8
- Accept-Encoding: gzip, deflate
- Accept-Language: fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4
- Cache-Control: no-cache
- Content-Type: image/png

Under the Response Headers, the Request Headers section is expanded, showing:

- Accept: image/webp,image/apng,image/*,*;q=0.8
- Accept-Encoding: gzip, deflate
- Accept-Language: fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4
- Cache-Control: no-cache
- Content-Type: image/png

Cookie information is present in both sections, including a long session ID and visitor unique identifiers.

Statistiques de consultation (web) mobile en France

Statistiques pour la France

Encore quelques chiffres

- Google, 2017 : les requêtes sur mobile dépassent celles sur ordinateur
- 3 secondes.
- 8,8 secondes.

Un point sur les types de développements pour mobiles

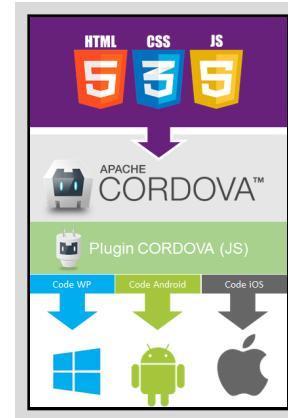
Quelles sont les options techniques pour le développement mobile (hors Web) ?

Applications natives

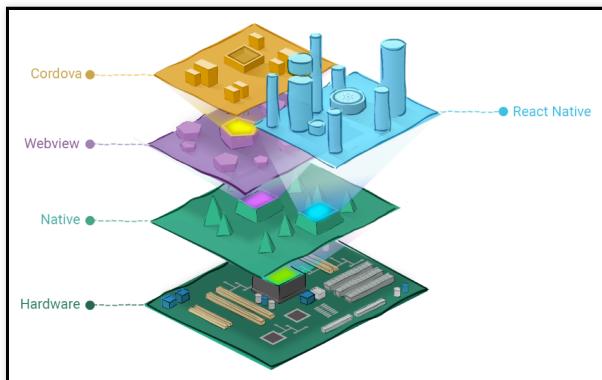
- Android : Java
- Apple : Objective C et Swift

Développements hybrides

- le Web avec les technologies HTML,CSS, et JS
- les hybrides (Cordova ou PhoneGap, ReactNative ou VueNative)



Cordova vs ReactNative



Sans hybride, le HTML5 permet de réaliser de plus en plus de choses !

Exemple de la l'API Push :
<https://gauntletface.github.io/simple-push-demo/>

Tableau de synthèse

| | Native | Web | Hybride (Cordova, Phonegap) |
|-----------------------------|--|--|--|
| Langage(s) de développement | Android Java Objetif C, Swift C/C++ | C# Visual Basic, HTML, CSS, JS | HTML, CSS, JS |
| Type de projet | Jeu, application gourmande en ressources, toute fonction de devoir doit respecter au plus près les prescriptions de la plateforme | Application de contenu, accès universel (mobile et desktop) | Pas trop gourmand en ressources, mais accès aux API (GPS, etc.) nécessaire |
| Atouts | <ul style="list-style-type: none"> - Accès aux "outils" spécifiques : GPS, SMS, etc. - Interface correspondant parfaitement aux standards de la plateforme | <ul style="list-style-type: none"> - Accès simple via une URL. Pas de nécessité de passer par les "stores". - Accès universel (mobile et desktop) à tous les appareils, mobiles ou pas (desktop, etc.) - Partage facile : un simple lien HTML | <ul style="list-style-type: none"> - Développement grâce aux langages du web. - Accès aux API spécifiques des mobiles : GPS, etc. (via des plugins) |
| Limites | <ul style="list-style-type: none"> - Téléchargement nécessaire - Mise à jour complète de l'application - Hors de toute plateforme - Distribution via les stores d'applications | <ul style="list-style-type: none"> - Pas d'accès à tous les outils : GPS, SMS, etc. - Utilisation dans la fenêtre du navigateur | <ul style="list-style-type: none"> - Pas aussi performant que le développement native - Pas de travail sur un développement Web car un peu d'intégration par plateforme est nécessaire à minima - Accès via les stores uniquement |

Quelles caractéristiques ?

Explorez l'exemple de lemonde.fr

Une (vraie) alternative Web avec les Progressive Web App (PWA)

- Les Web fournit aujourd'hui tous les outils pour développer des applications offrant une expérience utilisateur d'un niveau au moins égal aux applications natives ou hybrides tout en conservant les avantages spécifiques du Web (détails plus loin).
- Ma conviction : les PWA sont l'avenir du développement mobile.

Les caractéristiques d'une PWA

- Progressive
- Sécurisée
- Engageante
- Installable
- Rapidité
- Optimisation pour le référencement
- Indépendante de la connexion

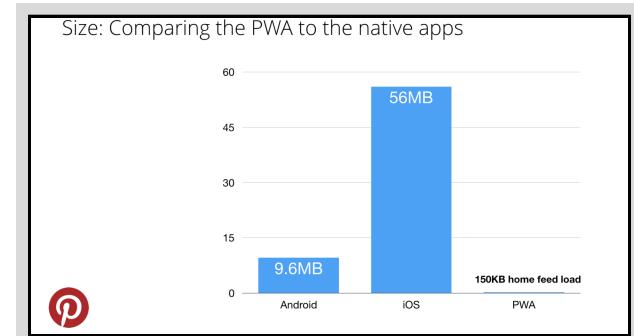
PWA, l'avenir du développement mobile ?

- Et néanmoins, comme les applications natives
 - consultable en plein écran
 - icône présente sur l'écran d'accueil (pas différentes des applications natives)
 - consultables hors ligne
 - accès aux principales API du mobile (géolocalisation, notifications, médias, etc.)

PWA vs applications natives ou hybrides

- PWA :
 - plus légères initialement
 - distribution par une simple URL (pas besoin de passer par les stores)
 - facilite la découverte (une simple URL à consulter)
 - mises à jour simplifiées (limitée aux fichiers modifiés)
 - "universelles" (consultables sur ordinateur aussi, sans développement spécifique)

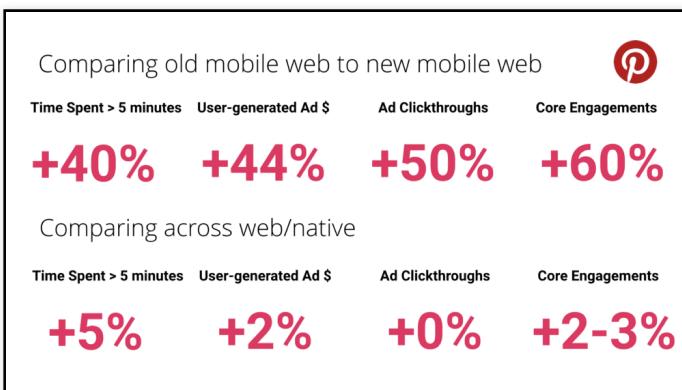
PWA vs app native : taille



Le cas de Pinterest

- En 2017, l'équipe Pinterest réalise qu'un grand nombre d'internautes ne convertissent pas sur smartphone. Ils ne se connectent pas, ne créent pas de compte et ne téléchargent pas l'application.

- Résultat : un an après le développement de la progressive web app, le nombre d'utilisateurs actifs hebdomadaires avaient augmenté de 103 %. Le temps d'utilisation de l'application a connu une hausse. Le nombre d'utilisateurs en Inde et au Brésil a également fortement augmenté.



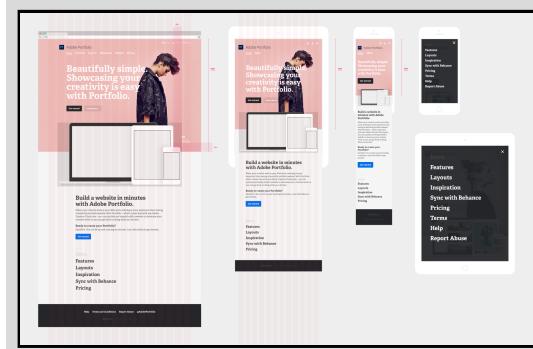
Le cas de Twitter Lite

« L'application web progressive de Twitter, Twitter Lite, pèse moins de 1 Mo, contre plus de 100 Mo pour son application native iOS et 23 Mo pour son application Android native », pointe Addy Osmani (responsable de l'ingénierie de l'équipe Chrome de Google)

L'application a pu bénéficier de 65 % d'augmentation de pages par session, 75 % d'augmentation des tweets envoyés et 20 % de baisse du taux de rebond comparé au site mobile avant PWA.

PWA et RWD

Bien sûr, sans le Responsive Web Design, pas de PWA !



La notion de Responsive Web Design (RWD) et les notions associées

Définition du RWD

Le Responsive Web design (RWD) est une approche de conception Web qui vise à l'élaboration de sites offrant une expérience de lecture et de navigation optimales pour l'utilisateur quelle que soit sa gamme d'appareil (téléphones mobiles, tablettes, liseuses, moniteurs d'ordinateur de bureau).

Concrètement, quelles sont les caractéristiques d'un site Web responsive ?

Exemples de sites non-responsive

- alistapart
- france2

Origine du concept RWD : Ethan Marcotte en 2010

- Combinaison de
 - grilles fluides
 - images flexibles, et
 - media queries
- <https://alistapart.com/article/responsive-web-design>

Des alternatives au RWD ?

Oui, le design adaptatif (Adaptive Web Design, AWD)

- Tout d'abord, il faut remarquer que le terme est source d'ambiguité, particulièrement en français, où il peut être la traduction de RWD :-(
- Exemple :

<https://developers.google.com/web/fundamentals/design-and-ux/responsive/?hl=fr>

Identifiez les caractéristiques du Design Adaptatif

Explorez le site de france.tv

Le concept d'AWD

- Accent sur l'adaptation des contenus, via la détection du device par le serveur.
- Donc, dans cette approche, des appareils différents obtiennent des versions de site différentes.
- L'URL peut être unique, ou bien multiple. Si l'URL pour la version mobile est spécifique, on parle aussi de stratégie de "site mobile dédié".

RWD vs AWD : synthèse en image

| | Responsive | Adaptive | Mobile Dédié |
|--------------------------|-----------------------------------|--|---|
| On conserve la même url | Oui monsite.com monsite.com | Oui monsite.com monsite.com | Non monsite.com m.monsite.com |
| On conserve le même HTML | Oui | Pas toujours Injection dynamique de certains composants HTML (carte, slider, etc.) dans un template responsive | Non Templates HTML créés sur mesure pour une taille d'écran (mobile, tablette, etc.) |
| On conserve le contenu | Oui | Pas toujours Injection dynamique de certains contenus (images ou blocs supplémentaires) dans un template responsive | Non Certains contenus peuvent ne pas être présentés sur mobile |

Au-delà du RWD et AWD

- La tendance actuelle est clairement de remplacer les sites adaptatifs par des PWAs.
- On l'a vu : c'est notamment le choix que [lemonde.fr](#) a fait.
- Objectifs : rapidité, légerté, adaptabilité !

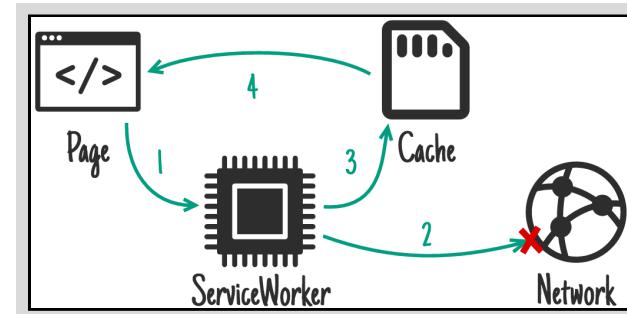
Au-delà du RWD et AWD : Explorez le cas de [lemonde.fr](#)

Quelles spécificités techniques ?

Au-delà du RWD et AWD : Réponse

- service worker => offline possible
- raccourci sur écran principal
- lancement fullscreen
- optimisée => rapide

Au-delà du RWD et AWD : Service Worker



La mise en page responsive en CSS

Premier élément de la mise en page responsive : la balise viewport

- Une balise mise au point par Apple lors de la sortie du premier iPhone.
- A placer dans le <head> du document (comme toutes les balises <meta>...)

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

Exercice responsive 1

Consultez cette URL sur votre mobile et sur votre ordinateur :

<http://vermonden.free.fr/rwd/ethan/index.html>

Quel est le problème ?

- Avec cette balise meta viewport, le navigateur des appareils mobiles comprendra qu'il ne doit pas simuler de largeur d'écran plus grande et le site s'affiche correctement.

Ressources

- MDN Mozilla
- W3schools

Second élément de la mise en page responsive : des grilles flexibles

Rappel CSS : Mettre en forme avec le CSS

- CSS est LE langage de mise au point des interfaces graphiques pour le web
- 3 manières d'ajouter du CSS
 - dans le code HTML
 - dans le head
 - dans une feuille externe
- le ciblage des éléments à mettre en forme :
https://www.w3schools.com/css/css_syntax.asp

Exercice de pré-requis

Créer un fichier HTML contenant 3 boîtes de 300 px de large sur 400 px de haut, avec une couleur différente pour chaque boîte. Les boîtes se trouvent les unes à côté des autres.

[./exercices/css-cols-1.html](#)

Exercice responsive

Comment transformer ces 3 colonnes pour qu'elles restent toujours sur une seule ligne quelle que soit la largeur de la fenêtre ?

[./exercices/css-cols-1-2.html](#)

Pour des grilles fluides, utilisez des unités relatives !

Les unités en CSS

- Pas besoin d'unité pour zéro
- unités absolues :
 - pt : point
 - in : pouce
 - px : pixel
 - mm, cm
- unités relatives :
 - em et rem
 - %
 - vw et vh
- https://www.w3schools.com/cssref/css_units.asp

em vs rem

- em : relatif à son parent
- rem : relatif au font-size de base (html, idéalement défini en %)
- Illustration : <https://engageinteractive.co.uk/blog/em-vs-rem-vs-px>

em vs rem

- Donc, REM comme compromis entre relatif et absolu
- Utilisez des REMs pour les tailles de police et les espacements de lignes, et recommandé aussi pour les marges.

font-size de base

- On évite d'utiliser un font-size à la racine en px.
- On utilisera plutôt : `html { font-size: 100% }` * *This means 16px by default**
- On respecte ainsi le choix par défaut de l'utilisateur.

% vs vw/vh

- vh et vw : relatif au viewport
- % : relatif à la taille du contenant
- par exemple, 100 vh pas possible en pourcentage au sens strict
- Exemple % vs vh
`./exercices/css-vh-vw-vs-pourcentage.html`
- Exercice

Réalisez ceci : `./exercices/css-vh-fullscreen-background.html`

Un exemple de grille fluide

<https://alistapart.github.io/code-samples/fluidgrids/examples/grid/final.html>

Analysons la ensemble.

Bilan de l'analyse

- Des largeurs exprimées en % ou em
- Des marges éventuellement en px, mais on pourrait se passer aussi du px.

Utiliser les bonnes unités

- px: do not scale. Use for borders only. That's all.
- em, rem (> IE8): scale with the font-size. 1.5em is 150% of the font size of the current element. 0.75rem is 75% of the font size of the html element. Use rem for typography and everything vertical like margins and paddings. Use em wisely for elements relative to the font-size (icons as a font for instance) and use it for media query breakpoints.
- %, vh, vw (> IE8): scale with the resolution. vh and vw are percentages of the viewport height and width.

Le troisième élément de la mise en page responsive en CSS : les media queries (@media)

- @media

La règle @media dans une feuille CSS permet de prévoir des règles de mise en page en fonction de caractéristiques spécifiques de l'appareil ou de la fenêtre d'affichage.

```
@media (min-width: 30em) { ... }
```

- - C'est cette règle @media qui permet notamment d'afficher ou de ne pas afficher des éléments en fonction de la taille de la fenêtre.

Charger des feuilles CSS de manière conditionnelle avec les media queries

```
<link rel="stylesheet" media="screen and (max-width: 1280px)" href="petiteresolution.css" />
```

Exemples simples

- ./exercices/media-query-demo.html
- ./exercices/css-3col.html

La notion de points de rupture (breakpoints)

La notion de point de rupture est centrale dans l'approche RWD. Il s'agit de décider des frontières (en largeur d'écran le plus généralement) en fonction desquelles on va réaménager le design.

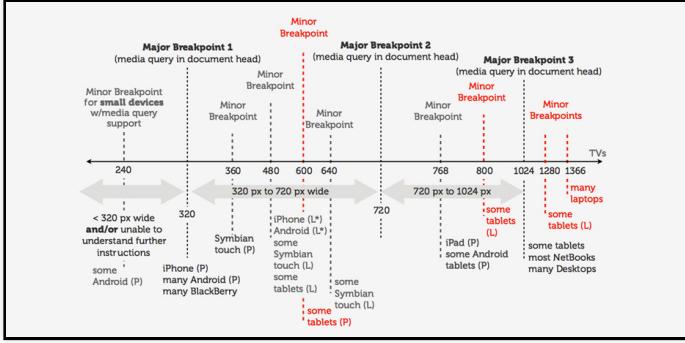
Et encore plus complet

./exercices/rwd-samplefiles-ethan-marcotte-book/index.html

Analysons cet exemple ensemble : quelles sont les adaptations réalisées ? Et comment sont-elles réalisées en CSS ?

Exemple de points de rupture standard

```
/* Extra small devices (phones, up to 480px) */
@media screen and (max-width: 767px) {
}
/* Small devices (tablets, 768px and up) */
@media (min-width: 768px) and (max-width: 991px) {
}
/* tablets/desktops and up ----- */
@media (min-width: 992px) and (max-width: 1199px) {
}
/* large desktops and up ----- */
@media screen and (min-width: 1200px) {
}
```



Les options de @media (suite) : screen et print

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

Les options de @media

- color : gestion de la couleur (en bits / pixel).
- height : hauteur de la zone d'affichage (fenêtre).
- width : largeur de la zone d'affichage (fenêtre).
- aspect-ratio : @media (aspect-ratio: 1 / 1)
- device-height (déprécié) : hauteur du périphérique.
- device-width (déprécié) : largeur du périphérique.
- orientation : orientation du périphérique (portrait ou paysage).

Et bien d'autres options

- En plus de "screen" et print, il existe d'autres options pour l'écran de sortie, mais la plupart ont été récemment dépréciés :
 - (déprécié) handheld : périphérique mobile ;
 - print : impression ;
 - (déprécié) tv : télévision ;
 - (déprécié) projection : projecteur ;
 - speech : correspond aux outils de synthèse vocale ;
 - all : tous les types d'écran.

En pratique, on ne va quasiment utiliser que

- max-width et
- min-width !

Ressources media queries

- https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries
- Spécification mediaqueries :
<https://www.w3.org/TR/css3-mediaqueries/>

Quelles unités utiliser dans les @media (em ou px) ?

- Idéalement, pas de px. Utilisez des em. Mais il faut veiller dans ce cas à définir la font-size de base (html) en pourcentage.
- Dans la pratique, encore pas mal de @media utilisent les px (et notamment le framework Bootstrap !). Le débat ici : <https://github.com/twbs/bootstrap/pull/17403>

Liste des aménagements

- Les options @media les plus utilisées concernent la largeur de l'écran.
- Les types d'aménagements les plus courants avec les media queries sont les suivants :
 - largeur des colonnes
 - taille de la police et des marges
 - déplacement d'items (logo, navigation, menu, etc.)
 - visibilité ou non d'items
 - adaptation des médias (photos, vidéos)

Une série de modèles d'adaptations

<https://developers.google.com/web/fundamentals/design-and-ux/responsive/patterns>

Autres ressources sur les "responsive patterns"

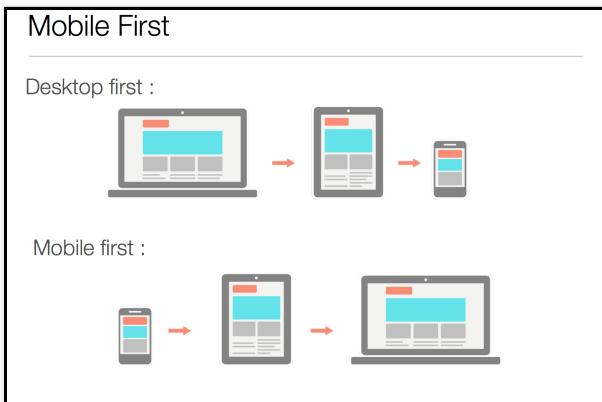
- des modèles généraux : <https://bradfrost.github.io/this-is-responsive/patterns.html>
- des modèles spécifiques pour la navigation :
<http://bradfrost.com/blog/post/responsive-navigation-patterns/>

Question : quelles améliorations possibles du design
d'Ethan Marcotte ?

Réponse

- Adopter une approche Mobile First
- Utiliser flex et/ou grid

L'approche Mobile First



Mobile First = penser d'abord aux supports mobiles, et adapter ensuite pour les autres supports.

Voici la version Mobile First de la version des points de rupture présentée précédemment :

```
CSS CODE FOR MOBILE HERE
/*
Small devices (tablets, 768px and up)
*/
@media screen and (min-width: 768px) {
}

/*
tablets/desktops and up -----
*/
@media screen and (min-width: 992px) {
}

/*
large desktops and up -----
*/
@media screen and (min-width: 1200px) {
```

Du responsive avec flex et grid

- L'exemple du Holy Grail Layout
 - Requirements :
 - Have a fluid center with fixed-width sidebars.
 - The center column (main content) should appear first in the HTML source.
 - All columns should be the same height, regardless of which column is actually the tallest.
 - They should require minimal markup.
 - The footer should “stick” to the bottom of the page when content is sparse.
 - ./exercices/float-flex-grid

Option : une grille responsive sans @media !

./exercices/rwd-with-grid-no-mq

Rappel : mise en page CSS Responsive à ce stade

- la balise <meta name="viewport"> dans le head
- des références relatives (em, rem, %, vh / vw)
- les media queries : @media (min-width: 640px) { ... }, avec principalement l'option min-width
- de nouvelles options display en CSS qui facilitent la mise en page : flex et grid

Option : la propriété calc()

- Permet d'utiliser le résultat d'un calcul comme valeur d'une propriété
- Permet de mélanger les unités
- Peut être utile pour s'en sortir quand certaines parties sont de taille fixe et d'autres de taille variable
- Utilisable partout sauf avec Opera sur mobile

Exemple

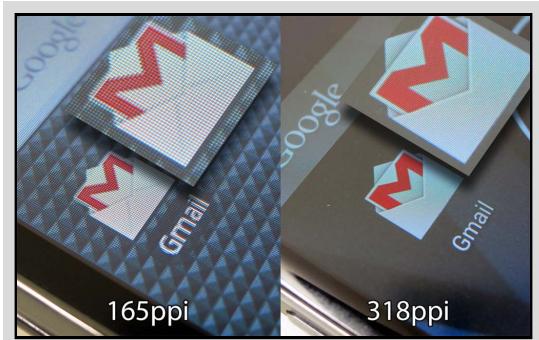
```
.bidule {  
    width: calc(100%/3 - 200px - 2*10em);  
}
```

RWD : La problématique des images sur mobile

3 problématiques concernant les images

- l'adaptation de l'image à la taille de l'écran ou de son conteneur,
- la densité des écrans (en PPP), et
- le poids des fichiers.

Densité des écrans



Exercice

Créer une page web reprenant un fichier d'image suivant vermonden.free.fr/montagne.jpg mais présentés avec des tailles différentes à l'écran.

Voyez-vous les 2 problèmes qui se présentent ?

Hum, mais si on réduit, l'image ne s'adapte pas à l'écran...

Et regardons le poids des fichiers chargés...

Premier problème : l'adaptation de l'image à l'écran
(responsive)

./exercices/exercice-img/img-responsive.html

Comment a-t-on fait ?

Code CSS

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

Second problème

- Adapter l'image à la résolution de l'écran (écrans Retina notamment)
- Ce faisant, on gère évidemment aussi le troisième problème puisque l'on évite d'envoyer la plus grosse image à tout le monde !

Solution avec CSS :

En CSS, proposer des images à la résolution 2 fois plus élevées pour les écrans Retina. Par exemple :

```
/* for high resolution display */
@media only screen and (min-device-pixel-ratio: 2) {
    .image {
        background: url(/path/to/my/highreslogo.png) no-repeat;
        background-size: 200px 400px;
        /* rest of your styles... */
    }
}
```

Pré-requis

On ne met pas en ligne des images brutes, directement issues de l'appareil photo ou du portable !

On prépare ses images (ou un script les prépare).

Solution alternative directement en HTML : srcset

On propose plusieurs images alternatives, avec des largeurs associés, ainsi que les tailles d'affichage de l'image.

```

```

Le navigateur va choisir l'image la plus adaptée.

Un exemple simple

```
<picture>  
  <source media="(orientation: landscape)" srcset="landscape.jpg">  
    
</picture>
```

Une seconde solution en HTML : picture

- A utiliser quand il s'agit aussi d'adapter l'image envoyée de manière conditionnelle (zoom par exemple).
- Cela s'appelle l'"art direction"

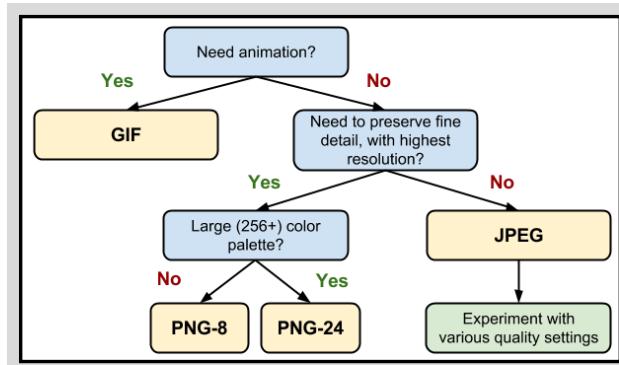
Et un exemple (un peu) plus complexe

```
<picture>  
  <!-- 16:9 crop -->  
  <source media="(min-width: 36em)" srcset="quilt_2/detail/large.jpg 1240w,  
          quilt_2/detail/medium.jpg 620w,  
          quilt_2/detail/small.jpg 310w">  
  <!-- square crop -->  
  <source srcset="quilt_2/square/large.jpg 822w,  
          quilt_2/square/medium.jpg 640w,  
          quilt_2/square/small.jpg 320w">  
    
</picture>
```

Démo

<https://responsiveimages.org/demos/variable-width/index.html>

Choisir le bon format d'image : les formats classiques



Et une version encore plus complète, exploitant les "types"

```
<picture>
  <!-- 16:9 crop -->
  <source
    type="image/webp"
    media="(min-width: 36em)"
    srcset="quilt_2/detail/large.webp 1920w,
            quilt_2/detail/medium.webp 960w,
            quilt_2/detail/small.webp 480w" />
  <source
    media="(min-width: 36em)"
    srcset="quilt_2/detail/large.jpg 1920w,
            quilt_2/detail/medium.jpg 960w,
            quilt_2/detail/small.jpg 480w" />
  <!-- square crop -->
  <source
    type="image/webp"
    srcset="quilt_2/square/large.webp 822w,
            quilt_2/square/medium.webp 648w,
            quilt_2/square/small.webp 320w" />
  <source
    srcset="quilt_2/square/large.jpg 822w,
            quilt_2/square/medium.jpg 640w,
            quilt_2/square/small.jpg 320w" />
  
</picture>
```

Choisir le bon format d'image : comparatif



De nouveaux formats

- WebP, développé par Google et
- svg, pour les images vectorielles (pour les icônes par exemple)

Stratégies complémentaires

- Utiliser le lazy-loading : on ne charge pas toutes les images d'un page immédiatement mais on les charge en fonction des besoins (déplacement sur la page). Exemple : Medium.
- Stratégie adaptative : on détecte la qualité de la connection côté serveur et on envoie des images adaptées en fonction de ce critère. Exemple : <http://adaptive-images.com>.

- On charge d'abord des images de petite taille, et on charge ensuite, en arrière plan, des images plus détaillées. Une fois ce chargement fini, on remplace les premières par les deuxièmes.
- Cache navigateur.

Conclusion

- Utilisez des tailles relatives pour les images afin d'éviter tout dépassement accidentel de la capacité du conteneur.
- Utilisez srcset pour proposer des images adaptées au besoin de l'usager, et éventuellement l'élément picture pour spécifier des images différentes en fonction des caractéristiques de l'appareil.

Option : logos responsive

<http://www.responsivelogos.co.uk/>

Adaptations complémentaires et spécifiques pour mobiles

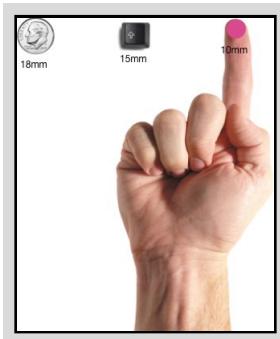
Touch screen

Quelles conséquences pour le design ?

Touch screen = pas de survol !

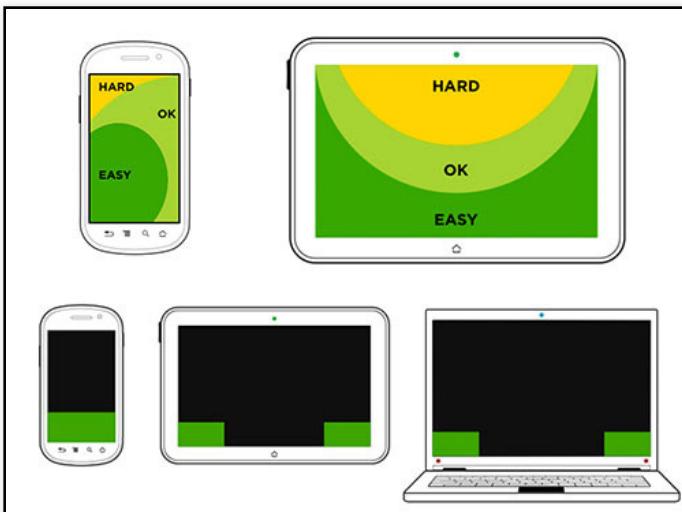
Je répète : pas de survol !

Touch screen = navigation avec les doigts

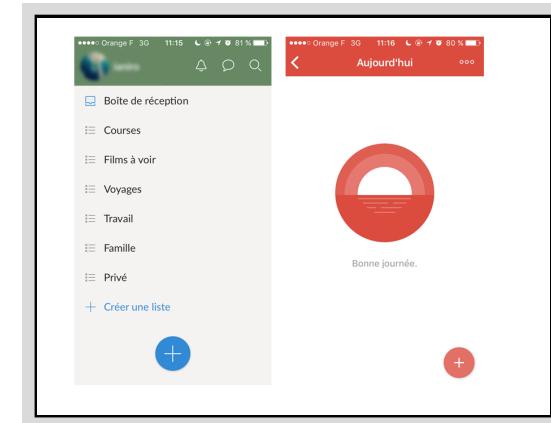


- Pas de souris, donc moins de précision.
- Des boutons de minimum 5 mm

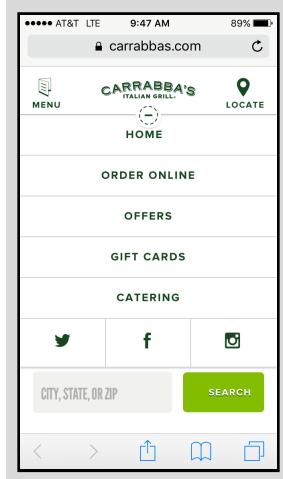
Navigation en bas plus accessible



Est-ce que le plus important est accessible avec mon pouce ?



Des menus simples



Ressource sur la navigation mobile

<https://www.lukew.com/ff/entry.asp?1649>

Le clavier virtuel

Conclusions

Rappel des contraintes du mobile

Identification des principales spécificités du mobile :

- une taille d'écran très variable
- une puissance qui peut être limitée
- un accès au réseau fluctuant (situation de mobilité)
- des API spécifiques (GPS, etc.)
- des interactions spécifiques : touch screen, clavier virtuel

Questions à se poser pour mobile

Répondez à ces questions avant de commencer un projet mobile :

- Dans quel contexte, les visiteurs consultent le site ?
- Y a-t-il des différences dépendantes du périphérique dans l'utilisation du site ?
- Comment le site est utilisé sur le desktop et quelles sont les tâches accomplies ?
- Quelles sont les exigences avec la page Web si la page est consulté avec un smartphone ?
- Quelles sont les ressources disponibles pour le développement et l'entretien du site ?
- Comment sont gérées et modulées les activités marketing pour ce type de périphérique ?

Mise en page

Avec grid, potentiel pour abandonner les frameworks

Et pour terminer

Expérience redesign du Guardian : "Performance is key" !

Référence : <https://lozworld.com/lozworts/notes-on-a-responsive-guardian-redesign>

Exercice : burger menu

Exemple 2
voir exercices / menu-burger-css.html

Exemple 1

Expliquez cet exemple :
[https://www.w3schools.com/howto/tryit.asp?
filename=tryhow_js_topnav](https://www.w3schools.com/howto/tryit.asp?filename=tryhow_js_topnav)

Projet

- Appliquez la démarche Mobile First à votre projet
- Veillez à ce que votre projet soit responsive