

Intégration Web

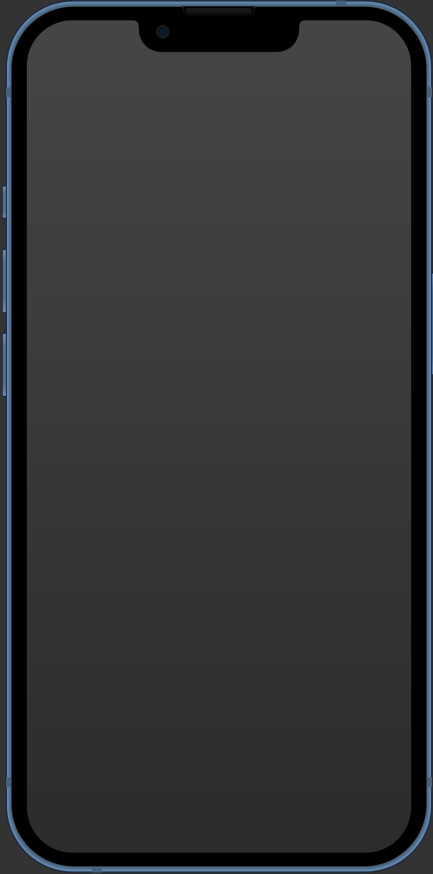
MMI – TP#4 S3



Danielo **JEAN-LOUIS**
Michele **LINARDI**

2007

iPhone



iPhone 13

- Annoncé en janvier 2007
- Sort en juin de la même année
- Premier vrai smartphone
- Premier vrai navigateur mobile
 - Safari mobile
 - Vrais débuts du web mobile

Smartphone / tablette

- Représentent la majorité de l'audience web
- Peuvent perdre leur connexion
- Souvent utilisés en déplacement
 - Utilisation de la data
- Terminaux différents d'un PC

Source(s) :

- <https://developer.apple.com/design/human-interface-guidelines/ios/user-interaction/gestures/> - anglais

Smartphone / tablette

- Présentent des caractéristiques qui leurs sont propres :
 - Écrans plus petits
 - Interface tactile → nouvelles interactions :
 - *zoom / swipe...*
 - Pas de clic droit (enfin au début)

Source(s) :

- <https://developer.apple.com/design/human-interface-guidelines/ios/user-interaction/gestures/> - anglais

Smartphone / tablette

- Orientation de l'écran
- Caméras
- Définition élevée d'écran (de nos jours)
- Notifications et Push
- [...]

Source(s) :

- <https://developer.apple.com/design/human-interface-guidelines/ios/user-interaction/gestures/> - anglais

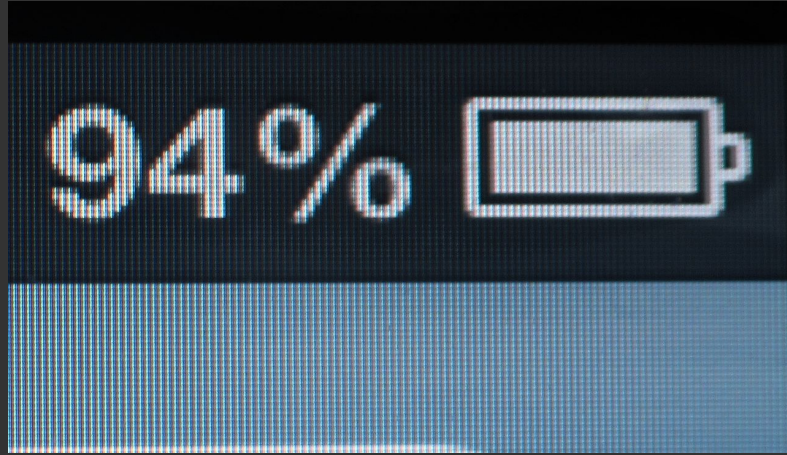
Point technique – Définition / résolution

- La définition représente le nombre de pixels total d'un média
 - Exemple : une image 3 000 x 2 000 a une définition 6 millions de pixels
- La résolution représente la densité de pixels. Autrement dit le nombre pixels par pouce (1 pouce \simeq 2,54cm)
 - Plus la densité est élevée, plus les traits sont détaillés. Retina est le terme utilisé par Apple pour désigner les écrans haute résolution (apparition avec l'iPhone 4 – 2010)

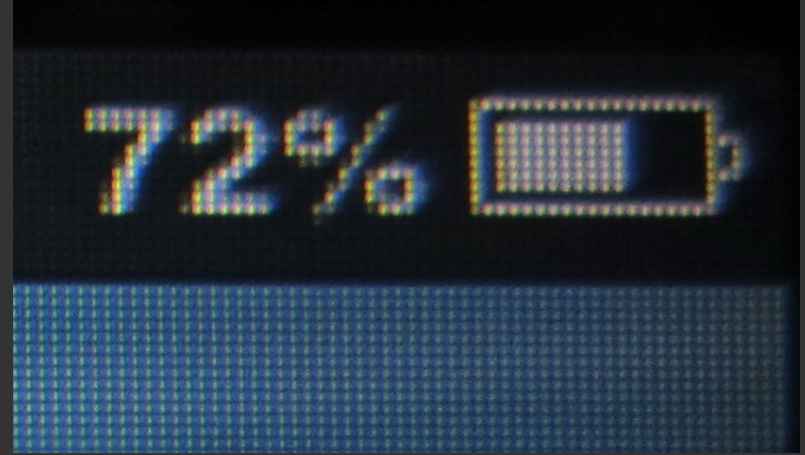
Source(s) :

- https://fr.wikipedia.org/wiki/D%C3%A9finition_d%27%C3%A9cran
- https://www.frandroid.com/comment-faire/comment-fonctionne-la-technologie/342762_difference-entre-definition-resolution
- <https://www.paintcodeapp.com/news/iphone-6-screens-demystified>

Petit point technique – Définition / résolution



Affichage Retina
(source wikipedia)



Affichage non Retina
(source wikipedia)

Source(s) :

- https://fr.wikipedia.org/wiki/D%C3%A9finition_d%27%C3%A9cran
- https://www.frandroid.com/comment-faire/comment-fonctionne-la-technologie/342762_difference-entre-definition-resolution
- https://fr.wikipedia.org/wiki/%C3%89cran_Retina
- <https://www.paintcodeapp.com/news/iphone-6-screens-demystified> - anglais

Responsive design

- Terme inventé par Ethan Marcotte en 2010
- Désigne un concept, **une approche du web** :
 - Site fluide (le site se comporte comme un liquide), voir sources
 - Images fluides / adaptatives
 - Utilisation des media queries
 - (Un seul code html pour tous les appareils)
 - **Ce n'est pas une technologie mais un ensemble**

Source(s) :

- <https://alistapart.com/article/responsive-web-design/> - anglais
- <https://mdn.github.io/css-examples/learn/rwd/liquid-width.html> – Site fluide
- https://developer.mozilla.org/fr/docs/Apprendre/CSS/CSS_layout/Responsive_Design
- <http://lehollandaisvolant.net/tuto/responsive-css>
- <https://www.alsacreations.com/article/lire/1615-cest-quoi-le-responsive-web-design.html>

Un peu de vocabulaire

- Responsive design : vu précédemment
- Fixed design : le web "ancien", le site (et ses éléments) ont une taille définie et si ça ne rentre pas... tant pis
- Adaptive design : forme évoluée du fixed design, les designers proposent des designs propres à "chaque écran" et les développeurs s'arrangent
- Fluid design : Le site et ses éléments ont des tailles en pourcentage

Sources :

- <https://medium.com/@popart.studio/fluid-vs-adaptive-vs-responsive-design-62de51e036bd> – en
- <https://www.alsacreations.com/article/lire/1615-cest-quoi-le-responsive-web-design.html>
- <https://responsivedesign.is/patterns/> - exemples basiques de mise en page responsive

Responsive design

- Nouvelle approche côté design
 - Les designers doivent également penser leur design en version mobile
 - Adapter la mise en page en fonction de l'écran

Sources :

- <https://medium.com/@popart.studio/fluid-vs-adaptive-vs-responsive-design-62de51e036bd> – en
- <https://www.alsacreations.com/article/lire/1615-cest-quoi-le-responsive-web-design.html>
- <https://responsivedesign.is/patterns/> - exemples basiques de mise en page responsive

Responsive design



Sources :

- <https://kinsta.com/fr/blog/design-web-respo>

Avant le Responsive design – Version mobile

- Nécessite un serveur pour la gestion
- Double charge de travail :
 - Version PC et mobile à gérer

Source(s) :

- <https://alistapart.com/article/responsive-web-design/> – anglais
- <https://mdn.github.io/css-examples/learn/rwd/liquid-width.html> – Site fluide
- https://developer.mozilla.org/fr/docs/Apprendre/CSS/CSS_layout/Responsive_Design
- <http://lehollandaisvolant.net/tuto/responsive-css>
- <https://www.alsacreations.com/article/lire/1615-cest-quoi-le-responsive-web-design.html>

Avant le Responsive design – Version mobile

- Techniques de détection de mobile pas fiable
 - User agent → Ne jamais lui faire confiance
- Utilisation du javascript
 - Lourdeur potentielle des scripts → lenteur de chargement de site

Source(s) :

- <https://alistapart.com/article/responsive-web-design/> – anglais
- <https://mdn.github.io/css-examples/learn/rwd/liquid-width.html> – Site fluide
- https://developer.mozilla.org/fr/docs/Apprendre/CSS/CSS_layout/Responsive_Design
- <http://lehollandaisvolant.net/tuto/responsive-css>
- <https://www.alsacreations.com/article/lire/1615-cest-quoi-le-responsive-web-design.html>

Images fluides oui, optimisation avant tout

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

Ces deux lignes rendent vos images responsives, elles gardent également leurs proportions

- Ces deux déclarations rendent nos images "fluides" tout en gardant leur proportion
- Elles ne dépasseront jamais la largeur de l'écran (et conteneur)
- Pour autant, ces images ne sont pas forcément optimisées

Sources :

- <http://lehollandaisvolant.net/tuto/responsive-css/#astuce-image>

Point accessibilité – Images

- Rappel : on n'oublie pas l'attribut "alt" même s'il doit être vide
- Les mobiles utilisent souvent la data pour accéder au web. Mettre des images de 15 Mo (même très belles) consomme beaucoup de données
 - Limiter la taille des images. Préférer des images qui font 3 Mo ou moins
 - Privilégier le format .jpg, plus léger que le .png (quand c'est possible)

Source(s) :

- <https://jakearchibald.com/2020/avif-has-landed/> - anglais
- <https://fr.wikipedia.org/wiki/WebP>
- <https://squoosh.app/> - pour compresser vos images
- https://developer.mozilla.org/fr/docs/Web/Media/Formats/Image_types

Point accessibilité – Images

- Formats à utiliser :
 - jp(e)g / png / .gif
 - webp : format avec compression sans perte par Google (pas géré par Safari)
 - AVIF : format prometteur avec compression sans perte
 - Non géré par tous les navigateurs à l'heure actuelle (10/2020)
 - svg : pour les logos / icônes – format vectoriel

Source(s) :

- <https://jakearchibald.com/2020/avif-has-landed/> - anglais
- <https://fr.wikipedia.org/wiki/WebP>
- <https://squoosh.app/> - pour compresser vos images
- https://developer.mozilla.org/fr/docs/Web/Media/Formats/Image_types

Point accessibilité – Images

- jpeg : format d'image avec **compression avec** perte
 - Extension : .jpg ou .jp(e)g
- png : format d'image avec **compression sans** perte
 - Extension : .png
 - Gère la transparence
- gif : format d'image avec **compression avec** perte
 - Extension : .gif
 - Gère les animations
 - Limité à 256 couleurs → Inadapté pour vos photos de vacances

Source(s) :

- <https://graphilink.fr/jpg-png-gif-differences/>

La tête dans le responsive

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

- Balise meta indispensable pour le responsive design
- Notre meta en détails :
 - width=device-width : Indique au navigateur que le site doit avoir une largeur égale à celle de l'écran
 - initial-scale=1 : Définit le zoom par défaut du site
- Accepte la valeur "user-scalable=no". **Ne jamais l'utiliser**, elle empêche l'utilisateur de zoomer

Sources :

- https://developer.mozilla.org/fr/docs/Apprendre/CSS/CSS_layout/Responsive_Design

La tête dans le responsive - Astuces

- Utilisation indispensable d'unités relatives
 - rem/pourcentage pour les longueurs
 - exemple : "width: 16rem" ou "padding: 3%"
 - rem pour les tailles de police
 - exemple : "font-size: 1.5rem"

Sources :

- https://developer.mozilla.org/fr/docs/Web/CSS/length#Longueurs_li%C3%A9es_au_viewport

La tête dans le responsive - Astuces

- Utilisation de longueur liées au viewport (taille réelle de l'écran)
- Suppression des tailles fixes
 - ~~width: 500px~~ → "min/max-width: 500px"
 - ~~height: 500px~~ → "min/max-height: 500px"

Sources :

- https://developer.mozilla.org/fr/docs/Web/CSS/length#Longueurs_li%C3%A9es_au_viewport

Tester son site mobile sur PC

- Le mode responsive (ou vue adaptative) des navigateurs permet de tester son site à des résolutions différentes. **Ce mode garde le rendu du navigateur**
- Sur MacOS, il est possible de brancher son iPhone / iPad et tester avec l'inspecteur de Safari

Sources :

- <https://www.idownloadblog.com/2019/06/21/how-to-use-safari-web-inspector-ios-mac/> - anglais
- <https://developers.google.com/web/tools/chrome-devtools/remote-debugging> - anglais

Tester son site mobile sur PC

- Sur Windows / MacOS, il est possible de brancher son smartphone Android et de tester avec l'inspecteur de Chrome
- Raccourci clavier : ctrl + shift (maj) + m (Chrome / Brave / Firefox)
 - Préférer Firefox, il gère bien mieux la densité de pixels des écrans simulés

Sources :

- <https://www.idownloadblog.com/2019/06/21/how-to-use-safari-web-inspector-ios-mac/> - en
- <https://developers.google.com/web/tools/chrome-devtools/remote-debugging> - en

Pratiquons ! - Images responsives

Pré-requis :

- Avoir la ressource `ressources/images-responsives/exemple.html`
- A télécharger ici :
<https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s3/travaux-pratiques/numero-4/ressources>

Notre image est responsive mais...

**Elle est trop grande pour nos
terminaux mobiles**

Adaptation des images

- Deux outils proposés par HTML5 :
 - Balise <picture>
 - Attributs srcset / sizes
- **Leur rôle n'est pas interchangeable**

Source(s) :

- <https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>
- <https://www.smashingmagazine.com/2014/05/responsive-images-done-right-guide-picture-srcset/> - anglais
- https://developer.mozilla.org/fr/docs/Apprendre/HTML/Comment/Ajouter_des_images_adaptatives_%C3%A0_une_page_web

Attributs srcset / sizes et balise <picture>

- Si l'image doit changer son contenu en fonction de l'écran → balise <picture>
 - Par exemple : je veux afficher une draisienne sur smartphone, un tricycle sur tablette et un vélo sur un PC
- Si l'image doit changer de taille en fonction du périphérique (largeur d'écran, résolution / densité de pixels) → attribut srcset sur la balise

Source(s) :

- <https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>
- <https://www.smashingmagazine.com/2014/05/responsive-images-done-right-guide-picture-srcset/> - anglais
- https://developer.mozilla.org/fr/docs/Apprendre/HTML/Comment/Ajouter_des_images_adaptatives_%C3%A0_une_page_web

Attributs srcset / sizes et balise <picture>

- Par exemple, Instagram utilise l'attribut srcset pour afficher une image en fonction de la taille d'écran de l'appareil

Source(s) :

- <https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>
- <https://www.smashingmagazine.com/2014/05/responsive-images-done-right-guide-picture-srcset/> - anglais
- https://developer.mozilla.org/fr/docs/Apprendre/HTML/Comment/Ajouter_des_images_adaptatives_%C3%A0_une_page_web

**Voyons comment modifier plutôt la
mise en page en fonction des écrans**

Media queries

- Introduit avec CSS3
- Permet de changer la mise en page pour des conditions données :
 - Taille limite d'écran (hauteur ou largeur)
 - Gestion de certaines propriétés CSS
 - Orientation de l'écran
 - ...
- Une des trois composantes du "Responsive Web Design"

Source(s) :

- https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%Aates_m%C3%A9dia/Utiliser_les_Media_queries
- <https://www.alsacreations.com/article/lire/930-css3-media-queries.html>

Media queries

- Gestion d'opérateurs logiques
 - and / only / not
 - Plusieurs conditions peuvent être appliquées en même temps
- Au service des terminaux mobiles
 - Ajustement du design pour ces terminaux → Adaptative design
- Peut être géré directement dans le CSS ou le HTML via la balise `<link />` ou `<style>`

Source(s) :

- https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%Aates_m%C3%A9dia/Utiliser_les_Media_queries
- <https://www.alsacreations.com/article/lire/930-css3-media-queries.html>

Media queries

- Flexbox (et sa petite sœur : "grid") seront des alliés de taille pour la mise en page responsive
- Utilisation d'unités relatives (%) et rem principalement
 - Le contenu s'adaptera aux écrans plus facilement

Source(s) :

- https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%Aates_m%C3%A9dia/Utiliser_les_Media_queries
- <https://www.alsacreations.com/article/lire/930-css3-media-queries.html>

Media queries – Exemple

```
@media screen and (max-width: 640px) {  
  .title {  
    font-size: 1.2rem;  
  }  
}
```

Source(s) :

- https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%Aates_m%C3%A9dia/Utiliser_les_Media_queries
- https://developer.mozilla.org/fr/docs/Web/CSS/R%C3%A8gles_@
- https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%Aates_m%C3%A9dia/Utiliser_les_Media_queries#types

Media queries – Exemple

```
@media screen and (max-width: 640px) {
```

```
  .title {  
    font-size: 1.2rem;
```

```
  }
```

```
}
```

Sélecteurs CSS

Opérateur logique "et"

Condition sur le média

Ici on définit que ces règles ne s'appliquent que sur un écran **ET** ayant une largeur de 640px maximum inclus

Media queries – Dans le css

- Tout comme les sélecteurs CSS, il est possible de mettre plusieurs médias queries sur la même ligne, chaque media query devant être séparée par une virgule

```
✓ @media (max-height: 440px), orientation: portrait) {  
✓   .titre-principal {  
     color: pink  
   }  
}
```

Opérateur logique "ou"

Condition sur le média

Sources :

- https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%Aates_m%C3%A9dia/Utiliser_les_Media_queries
- https://developer.mozilla.org/fr/docs/Web/CSS/R%C3%A8gles_@

Media queries – Dans le css

```

@media (orientation: portrait) {
  .titre-principal {
    color: pink
  }
}

@media (max-height: 440px) {
  .titre-principal {
    color: pink
  }
}
```

Ceci équivaut au code de la slide précédente, en revanche, c'est plus verbeux et plus difficilement maintenable

Sources :

- https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%Aates_m%C3%A9dia/Utiliser_les_Media_queries
- https://developer.mozilla.org/fr/docs/Web/CSS/R%C3%A8gles_@

Media queries – mots-clés

- Gestion de taille de l'écran :
 - min/max-height | min/max-width
- Gestion de l'orientation de l'écran :
 - orientation : landscape (paysage) ou portrait
- Gestion du mode nuit du système d'exploitation :
 - prefers-color-schema : light ou dark
- et : and / ou : or / seulement : only
- [...]

Sources :

- https://developer.mozilla.org/en-US/docs/Web/CSS/@media#browser_compatibility
- <https://css-tricks.com/a-complete-guide-to-css-media-queries/> - anglais

Pratiquons ! - media queries (Partie 1)

Pré-requis :

- Avoir la ressource ressources/media-queries
- A télécharger ici :
<https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s3/travaux-pratiques/numero-4/ressources>

Media queries – Dans le html

- Utilisation de la balise <link> avec l'attribut "media"
- Plus besoin d'utiliser la règle @, l'attribut media remplissant le rôle

```
<link rel="stylesheet"  
      media="screen and (max-width: 640px)"  
      href="smartphone.css" type="text/css" />
```

Chemin vers le fichier css

Condition sur le média pour le fichier entier

Ici on définit que le contenu du fichier "smartphone.css" ne s'applique que sur un **écran ayant une largeur de 640px maximum inclus**

Sources :

- https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%Aates_m%C3%A9dia/Utiliser_les_Media_queries
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/link>

Pratiquons ! - media queries (Partie 2)

Pré-requis :

- Avoir la ressource ressources/media-queries
- A télécharger ici :
<https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s3/travaux-pratiques/numero-4/ressources>

Questions ?