

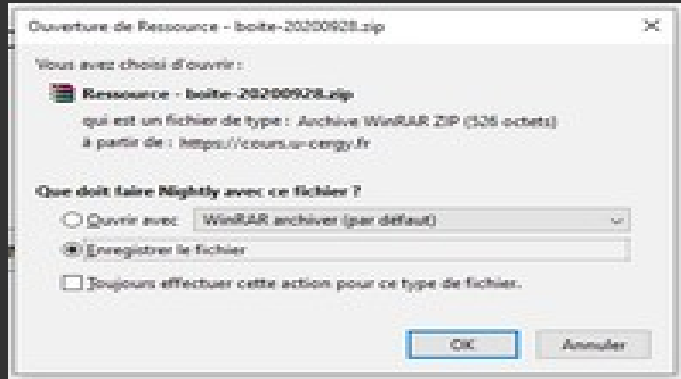
# Intégration Web

TP#6 - S1

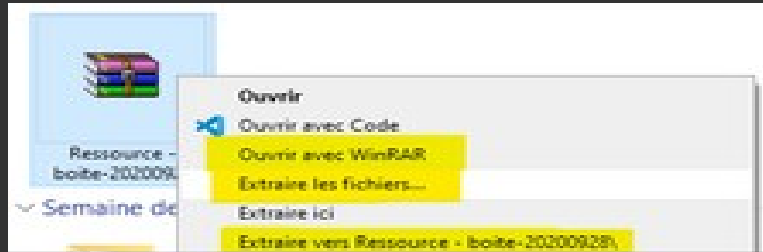


Danielo **JEAN-LOUIS**  
Michele **LINARDI**

# Petit point sur les ressources de cours

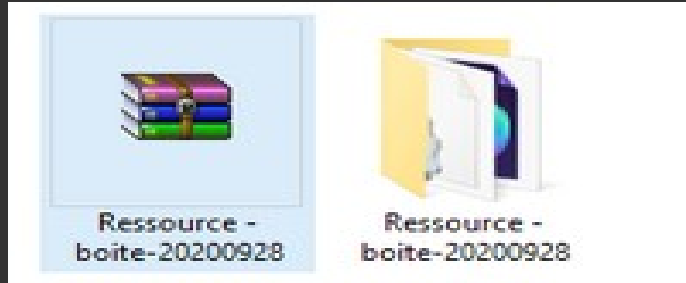


- Les ressources se présentent sous la forme d'une archive une fois téléchargées
- L'archive doit être impérativement extraite sinon vous ne serez pas capables de réaliser correctement les tp



Menu contextuel sur l'archive (clic droit). Choisir un des choix surligné.

# Petit point sur les ressources de cours



- Une fois extrait, vous devez obtenir un dossier avec le contenu de l'archive
- Libre à vous de travailler dans le dossier OU en copier le contenu pour le coller ailleurs
- **En aucun vous devez travailler dans l'archive**

**2007**

# iPhone



iPhone 11 Pro Max

- 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

- Points à prendre en compte sur ces appareils :
  - Représentent la majorité de l'audience web
  - Écrans plus petits
  - Interface tactile → nouvelles interactions :
    - zoom / double toucher / *swipe*
  - Pas de clic droit (enfin à l'époque)
  - La mobilité (l'utilisateur n'est pas tout le temps en wi-fi)
    - Mode hors-ligne

## Source :

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

# Smartphone / tablette

- Orientation de l'écran
- Caméra disponible
- Définition élevée de nos jours
- Notifications et Push
- [...]

## Source :

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



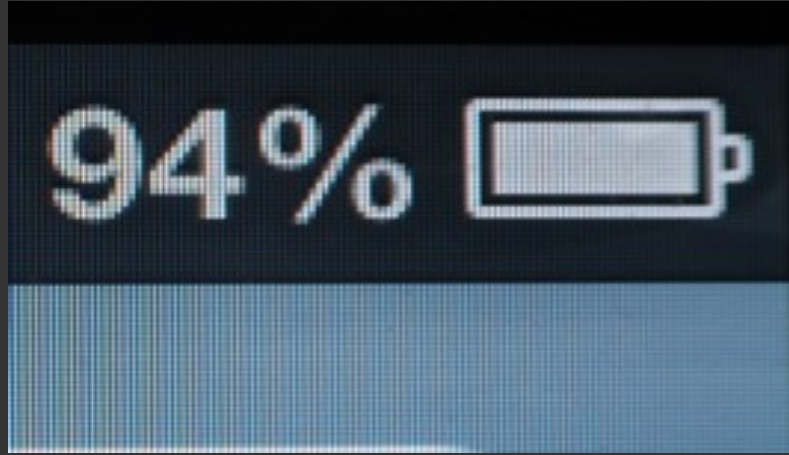
# Point technique – Définition / résolution

- La définition représente le nombre de pixels qu'un écran peut afficher
  - Exemple : une image 3000 x 2000 a une définition 6 millions de pixels
- La résolution représente la densité de pixels. Autrement dit le nombre pixels par pouce (1 pouce  $\approx$  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)

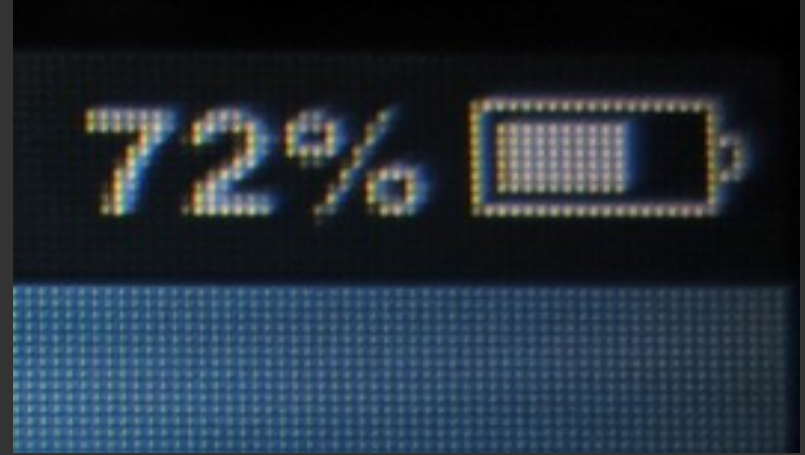
## Sources :

- [https://fr.wikipedia.org/wiki/D%C3%A9finition\\_d%27%C3%A9cran](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.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)

## Sources :

- [https://fr.wikipedia.org/wiki/D%C3%A9finition\\_d%27%C3%A9cran](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.frandroid.com/comment-faire/comment-fonctionne-la-technologie/342762_difference-entre-definition-resolution)
- [https://fr.wikipedia.org/wiki/%C3%89cran\\_Retina](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**

## Sources :

- <https://alistapart.com/article/responsive-web-design/> - en
- <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](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 Web Design  
est le mot magique**

# Avant le Responsive design – Version mobile

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

## Sources :

- <https://alistapart.com/article/responsive-web-design/> - en
- <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](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 à Internet. 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

## Sources :

- <https://jakearchibald.com/2020/avif-has-landed/> - en
- <https://fr.wikipedia.org/wiki/WebP>
- <https://squoosh.app/> - pour compresser vos images
- [https://developer.mozilla.org/fr/docs/Web/Media/Formats/Image\\_types](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

## Sources :

- <https://jakearchibald.com/2020/avif-has-landed/> - en
- <https://fr.wikipedia.org/wiki/WebP>
- <https://squoosh.app/> - pour compresser vos images
- [https://developer.mozilla.org/fr/docs/Web/Media/Formats/Image\\_types](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

## Sources :

- <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](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](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](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/> - en
- <https://developers.google.com/web/tools/chrome-devtools/remote-debugging> - en

# 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 responsive (Partie 1)

Pré-requis :

- Avoir la ressource `ressources/images-responsives/srcset.html`
- A télécharger ici :  
<https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-6/ressources>



**Notre image est responsive mais...**

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

# Attributs srcset / sizes et balise <picture>

- Permettent de gérer des images en fonction de l'appareil
  - Taille d'écran
  - Résolution
  - Orientation (uniquement la balise picture)
  - Formats fichiers gérés (uniquement la balise picture)
- Chargent uniquement l'image pertinente
  - Économie de bande passante
- Gérés par les navigateurs modernes

## Sources :

- <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](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Comment/Ajouter_des_images_adaptatives_%C3%A0_une_page_web)

# Attributs srcset / sizes et balise <picture>

- Leur rôle n'est pas interchangeable
- 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 <img />

## Sources :

- <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](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Comment/Ajouter_des_images_adaptatives_%C3%A0_une_page_web)

# Attribut srcset

- Attribut "srcset" liste les sources des images, le navigateur choisira la plus pertinente tout seul
- Valeurs séparées par une virgule
- Image peut être associée à une résolution ou une taille

**srcset="mon-imageHD.jpg 2x"**

descripteur  
(ici de densité)

source de  
mon image

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-srcset>
- <https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>

# Attribut srcset – descripteurs

- Descripteur "x"
  - Désigne la densité de pixels de l'écran
  - 1x représente un écran normal / 2x un écran HD...
  - Accepte un nombre décimal. Exemple "3.5x"
- Descripteur "w"
  - Désigne la largeur de l'image (pas de l'écran)
  - Accepte un entier positif. Exemple "320w"

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-srcset>
- <https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>

# Attribut srcset – descripteurs

- **Attention** : Descripteur de densité les images doivent être proportionnelles
  - Exemple :
    - @3x (écran de densité de pixels de 3) → 3000 de largeur
    - @2x (écran de densité de pixels de 2) → 2000 de largeur (66,67%)
    - @1x (écran de densité de pixels de 1) → 1000 de largeur (33,33%)
  - Astuce : partir de l'image la plus grande puis la rétrécir pour avoir les autres images
  - Astuce 2 : Avoir une image @4x pour trouver plus facilement les coefficients de rétrécissement (100 → 75 → 50 → 25)
- Pour le descripteur de densité, l'image occupera la largeur de l'image définie dans l'attribut "src"

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-srcset>
- <https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>

# Attribut srcset – En application

```

```

Le code ci-dessus dit au navigateur "j'ai une image de 320 px de largeur et une autre de 640 px de largeur, prend la plus adaptée. Si tu ne connais pas l'attribut "srcset" ou qu'aucune valeur de l'attribut "srcset" convient affiche "mon-image.jpg"

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-srcset>
- <https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>



# Attribut srcset

- Équivalent pour la propriété "background-image" avec la fonction "image-set()"
  - **Pas géré correctement par Safari (ne sait pas gérer l'unité DPI)**
  - Ne permet de gérer que la densité de l'image
  - Syntaxe :

```
.test-image-set {  
    background-image: image-set(  
        "chat.png" 1x,  
        "chat-2x.png" 2x,  
        "chat-print.png" 600dpi  
    );  
}
```

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/image-set>

# Attribut sizes

- Lié à la balise <img />
- Fonctionne de pair avec l'attribut "srcset"
  - Uniquement s'il y a des descripteurs de largeur "w"
- Éléments séparés par une virgule

**sizes="**(max-width : 320px) 280px, 900px"  
condition sur le média  
(media query)

espace occupé par  
l'image pour la condition

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-srcset>
- <https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-sizes>

# Attribut sizes

- Important : En absence de l'attribut "sizes", le navigateur estime que votre image doit s'afficher à 100vw soit 100 % de la largeur de l'écran
- Ne fonctionne qu'avec le descripteur de largeur

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-srcset>
- <https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>

# Pratiquons ! - Images responsive (Partie 2 et 3)

Pré-requis :

- Avoir la ressource `ressources/images-responsives/srcset.html`
- A télécharger ici :  
<https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-6/ressources>

**Notre image change maintenant en  
fonction de l'écran et possède une  
taille en fonction de l'écran**

**Voyons maintenant  
la balise <picture>**

# Balise <picture>

- Utilise la balise <img /> comme valeur par défaut
  - C'est elle qui définit le style des images même si elle n'est pas affichée
  - Il ne faut pas mettre de CSS sur la balise <picture> mais <img>
- Fonctionne de pair avec la balise <source/>
  - Fonctionne avec les attributs srcset et media/type

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/picture>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source>

# Balise picture

- A utiliser si l'image doit changer (rognage par exemple) en fonction du périphérique
  - A utiliser pour la "direction artistique"
- Permet de gérer différents formats d'image quand certains ne sont pas gérés par le navigateur
  - Ex : le format svg n'est pas géré alors le navigateur va afficher un autre format

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/picture>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source>



# Balise picture – balise source

## Dans le cas de la balise <picture>

- Fonctionne avec les attributs "srcset" et "media/type"
  - srcset : définit un lien vers une image
  - media : définit la condition sur le media (media queries)
  - type : définit le format de l'image (au format MIME)
  - Attributs "media" et/ou "type" nécessaires avec l'attribut "srcset"
- srcset fonctionne comme vu précédemment. On peut donc définir plusieurs images pour le même media/type dépendamment de la définition de l'écran

Une fois tout ça défini, le navigateur s'adapte

### Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/picture>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source#attr-media>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source#attr-srcset>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source#attr-type>
- <https://alligator.io/html/picture-element/> - anglais

# Balise <picture> – En application

<picture>

<source srcset="image-medium.jpg" media="(max-width: 399px)" />

<source srcset="image-mini.jpg" media="(min-width: 400px) and (max-width: 768px)"/>



</picture>

Le code ci-dessus dit au navigateur "j'ai une image à afficher pour un appareil dont la largeur fait au maximum 399px, si l'appareil ne correspond pas, j'ai une image à afficher pour appareil dont la largeur est comprise entre 400 et 768px. Si tu ne trouves pas ou que tu ne connais pas la balise <source> j'ai une image par défaut à afficher"

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/picture>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source#attr-media>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source#attr-srcset>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source#attr-type>
- <https://alligator.io/html/picture-element/> - anglais

**L'ordre des valeurs de l'attribut  
"media" est important**

# Pratiquons ! - Balise <picture>

Pré-requis :

- Avoir la ressource `ressources/images-responsives/picture.html`
- A télécharger ici :  
<https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-6/ressources>

# Balise picture – Combiner le tout

```
<picture>
  <source
    media="(max-width: 1113px)"
    srcset="image-pc@2x.png 2x, image-pc@2x.png 1x"
  />
  <source
    media="(max-width: 767px)"
    srcset="image-mobile@3x.png 3x, image-mobile@2x.png 2x, image-mobile@1x.png 1x"
  />
  
</picture>
```

Notre code (un peu verbeux) affiche une image pour mobile et une pour PC, mais gère également la densité de pixels des écrans.

(La second <source> ne sera jamais appliquée)

## Sources :

- <https://webdesign.tutsplus.com/tutorials/quick-tip-how-to-use-html5-picture-for-responsive-images—cms-21015> - anglais

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

# Media queries

- Introduit avec CSS3
- Permet de définir un comportement 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"

## 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/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 → Adaptive design
- Peut être géré directement dans le css ou le html via la balise `<link />` ou `<style>`

## 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/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
  - En ligne sur grand écran | En colonne sur petits écrans
- Utilisation d'unités relatives (% et rem principalement)
  - Le contenu s'adaptera aux écrans plus facilement

## 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/Requ%C3%Aates_m%C3%A9dia/Utiliser_les_Media_queries)
- <https://www.alsacreations.com/article/lire/930-css3-media-queries.html>

# Media queries – Dans le css

- Nécessite la règle @ (at) @media et ses conditions

```
@media screen and (max-width: 640px) {  
  .titre-principal {  
    font-size: 25px;  
  }  
}
```

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**

## 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/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/R%C3%A8gles_@)
- [https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%Aates\\_m%C3%A9dia/Utiliser\\_les\\_Media\\_queries#types](https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%Aates_m%C3%A9dia/Utiliser_les_Media_queries#types)

# 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/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/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/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/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://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-s1/travaux-pratiques/numero-6/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/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-s1/travaux-pratiques/numero-6/ressources>





**Questions ?**