

# Intégration Web

MMI - Travaux Pratiques #4 - S1

Danielo **JEAN-LOUIS**  
Développeur front-end

# But du cours

- Voir les bases de la programmation d'un site web
  - Langage HTML
  - Langage CSS
- Sensibilisation au web : design et programmation
  - Accessibilité
  - Bonnes pratiques
- Avoir les connaissances pour développer un site web

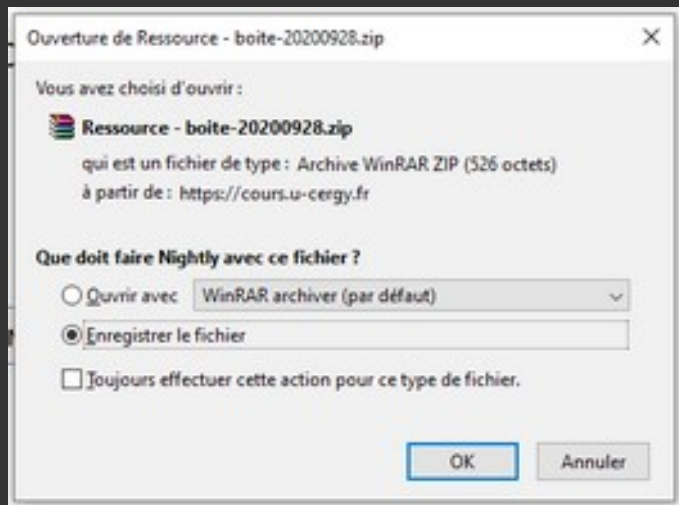
# Récapitulatif du tp précédent

- Les pseudo-formats et pseudo-éléments
- Les transitions css
- La propriété css "transform" et ses fonctions

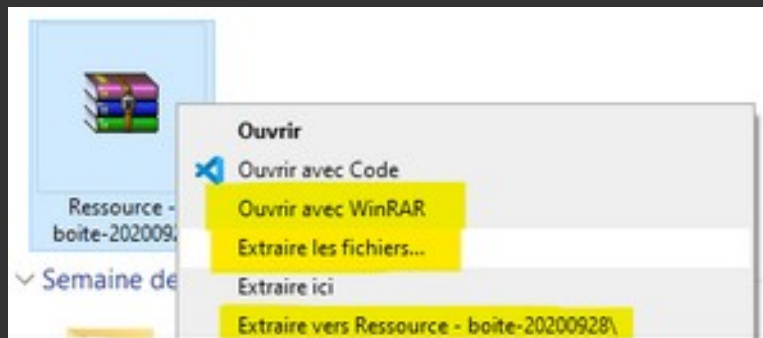
**Source :**

- <https://github.com/DanYellow/presentations/blob/master/mmi/html/travaux-pratiques/numero-3/presentation.pdf>

# Petit point sur les ressources de cours

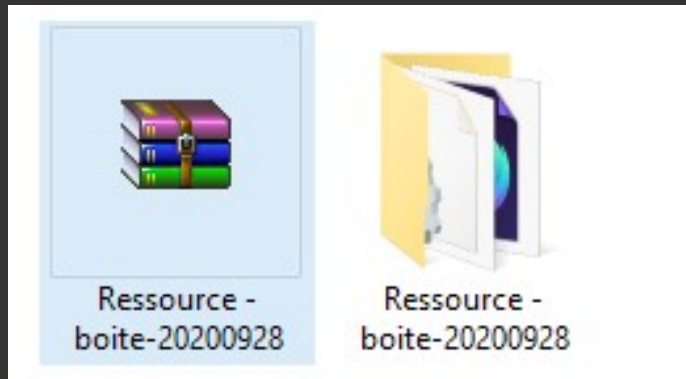


- Les ressources se présentent sous la forme d'une archive une fois téléchargées
- L'archive doit être impérativement extraite sinon vous ne serez pas capables de réaliser correctement les tp



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

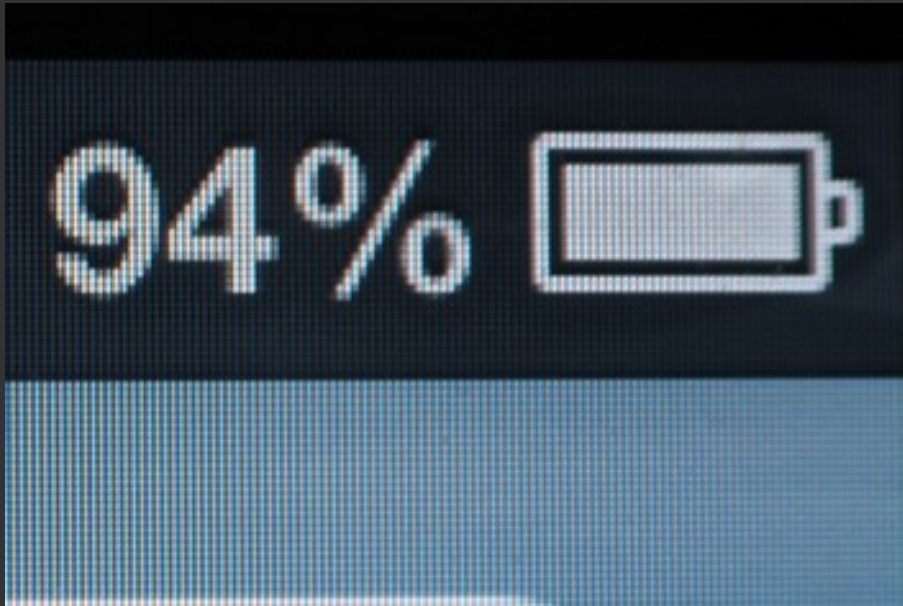
# Point technique – Définition / résolution

- 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)

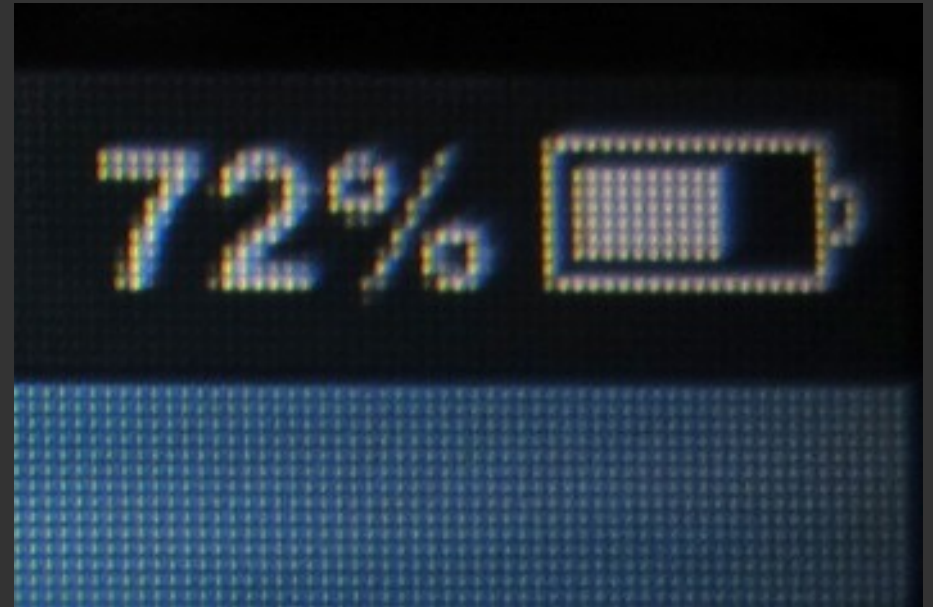
## Sources :

- [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>

# Responsive design

- Terme inventé par Ethan Marcotte en 2010
- Désigne 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 pas lui faire confiance
- Utilisation du javascript
  - Lourdeur potentielle des scripts

## 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

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

## Sources :

- <https://jakearchibald.com/2020/avif-has-landed/> - en
- <https://fr.wikipedia.org/wiki/WebP>
- <https://squoosh.app/> - pour compresser vos images

# 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
  - Pourcentage pour les longueurs
    - exemple : "width: 50%" ou "padding: 3%"
  - em/rem/pourcentage pour les tailles de police
    - exemple : "font-size: 1.5rem"
- 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
- 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

# Mode mobile sur PC - Firefox



# Mode mobile sur PC - Chrome

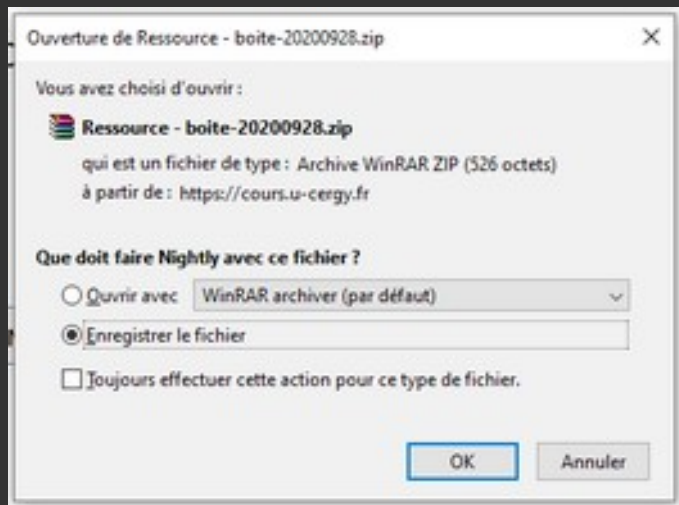




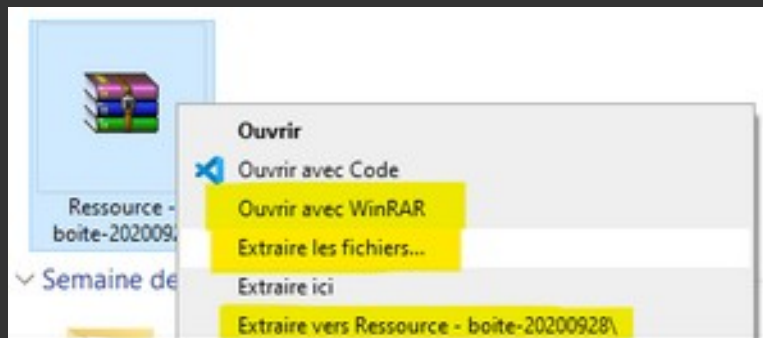
# Ressource du cours sur ENT

## Travaux Pratiques #4 > "Ressource – Images responsives"

# Petit point sur les ressources de cours

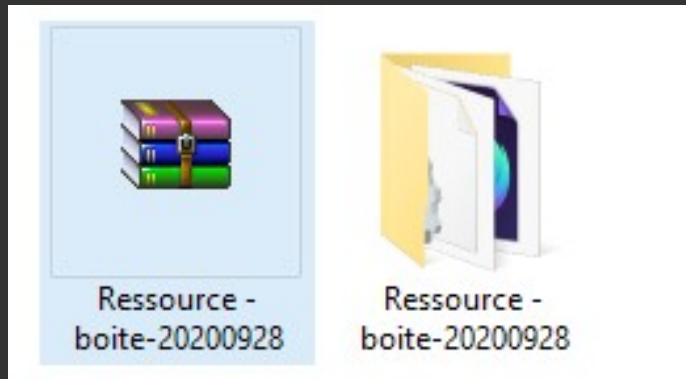


- Les ressources se présentent sous la forme d'une archive une fois téléchargées
- L'archive doit être impérativement extraite sinon vous ne serez pas capables de réaliser correctement les tp



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

# Petit point sur les ressources de cours



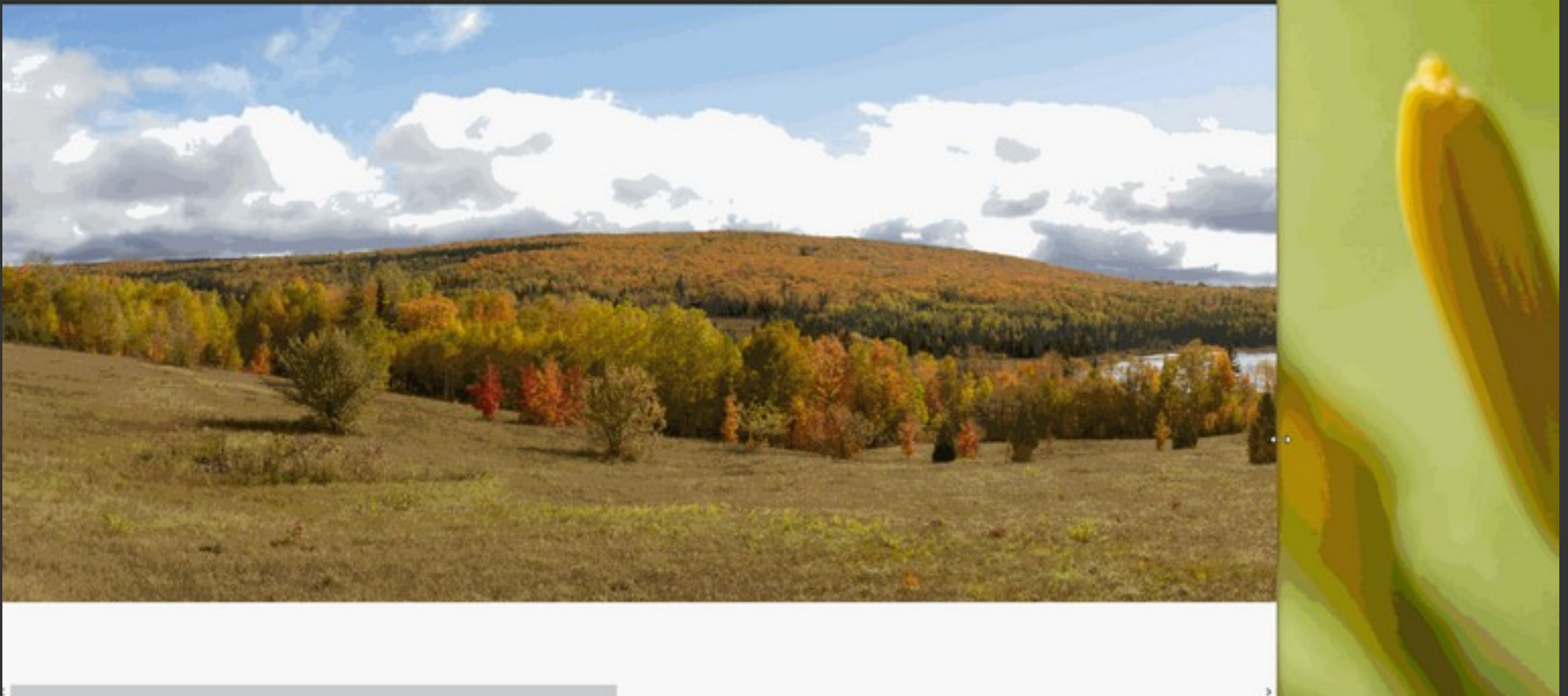
- Une fois extrait, vous devez obtenir un dossier avec le contenu de l'archive
- Libre à vous de travailler dans le dossier OU en copier le contenu pour le coller ailleurs
- **En aucun vous devez travailler dans l'archive**
- Une fois fait, glissez le dossier dans VS Code

# Pratiquons ! - Notre image responsive

Pré-requis :

- Avoir la ressource "images-responsives"
- Ouvrir le fichier "srcset.html" dans votre éditeur de code
- Ajouter une balise `<img >` avec l'image "panorama.jpg" présente dans le dossier "assets/images"

# Pratiquons ! - Notre image responsive



Notre image s'affiche, mais on a un défilement horizontal

# Pratiquons ! - Notre image responsive

Pré-requis :

- Avoir la ressource "images-responsives" et réalisé la pratique précédente
- Ajouter une classe "img-responsive" dans un fichier css avec les déclarations suivantes :

```
max-width: 100 %;  
height: auto;
```

- Ajouter la classe à l'image
- Redimensionnez votre fenêtre, votre image s'adapte maintenant à votre fenêtre

**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 drapsienne 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 par tous les navigateurs présentement**
- 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"**

taille de l'image pour la condition donnée

condition sur le média (media query)

## 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.

Si et seulement si le descripteur est 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 ! - Attributs srcset / sizes

Pré-requis :

- Avoir la ressource "images-responsives"
- Les consignes sont dans la ressource, dans la page srcset.html

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

# 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
- 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 → On parlera de "direction artistique"
- Permet de gérer différents formats d'image quand certains ne sont pas gérés par le navigateur
  - Ex : le format png n'est pas géré alors le navigateur va gérer 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

**Si vous ne mettez que des bornes minimum, elles doivent être dans l'ordre décroissant**

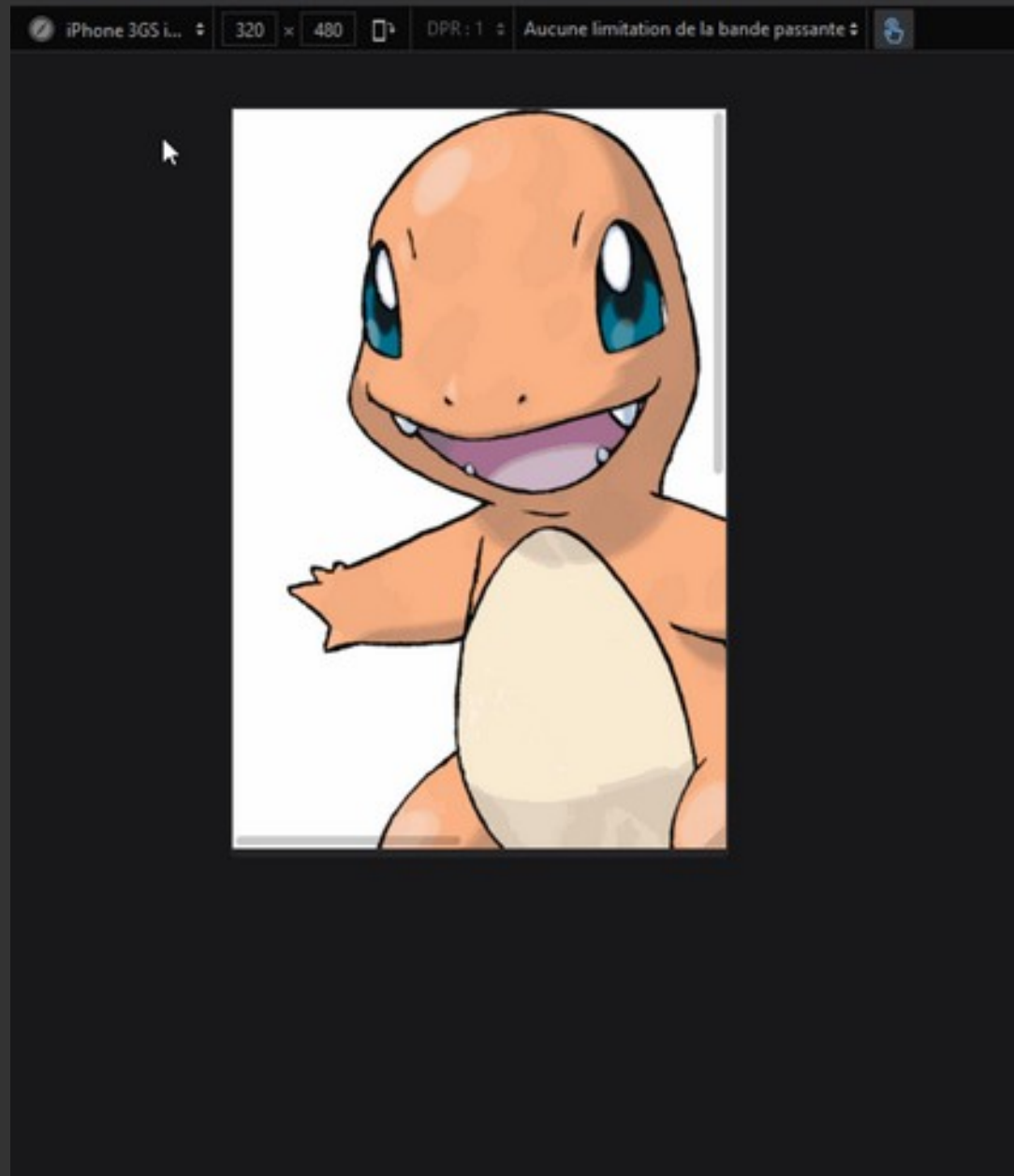


# Pratiquons ! - Balise <picture>

Pré-requis :

- Avoir la ressource "images-responsives"
- Compléter la structure <picture> présente dans la ressource et gérer trois largeurs d'écran dans le fichier picture.html
  - Largeur écran  $\leq 767\text{px}$  (smartphones)
  - $768\text{px} \leq \text{largeur écran} \leq 1024\text{px}$
  - $1025\text{px} \leq \text{largeur écran} \leq 1099\text{px}$
- Utiliser les images contenues dans le dossier "assets/images/balise-picture/"
- Plus de détails dans le fichier "picture.html" de la pratique

# Pratiquons ! - Balise <picture>



# 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 de pourcentage
  - 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'applique 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-principale {  
~     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_@)

# Pratiquons ! - Media queries css

Pré-requis :

- Avoir la ressource "media-queries"
- Éditer la règle @media de façon à ce que la taille du titre "Ma galerie d'images" change sur un appareil dont la largeur d'écran est inférieure ou égale à 640px
- Ajouter une règle @media pour gérer un écran dont la taille est supérieure ou égale à 400px et appliquer des déclarations sur n'importe quel sélecteur CSS
  - Ex : ajouter des marges latérales
- Ajouter une règle @media pour gérer un écran ayant l'orientation paysage (landscape en anglais)

# 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 dans le html

Pré-requis :

- Avoir la ressource "media-queries"
- Avoir réalisé la pratique "media queries css"
- Lister le nombre de règles @media contenues dans le fichier style.css
- Créer autant de fichiers css qu'il y a de règles @media
- Lier le fichier index.html aux fichiers css (ne pas oublier les attributs media)
- Placer le contenu des règles @media dans les bons fichiers

# Quiz

**"Test – Responsive et mobile" sur ENT**

Conditions du test :

- 10 minutes
- 1 tentative



**Questions ?**



# Intégration Web

MMI - Travaux Pratiques #4 - S1





Danielo **JEAN-LOUIS**  
Développeur front-end

## **But du cours**

- Voir les bases de la programmation d'un site web
  - Langage HTML
  - Langage CSS
- Sensibilisation au web : design et programmation
  - Accessibilité
  - Bonnes pratiques
- Avoir les connaissances pour développer un site web

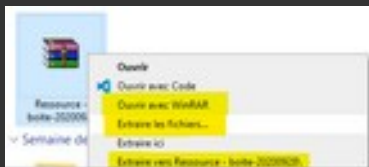
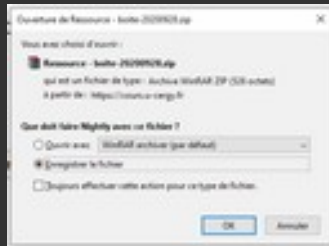
## Récapitulatif du tp précédent

- Les pseudo-formats et pseudo-éléments
- Les transitions css
- La propriété css "transform" et ses fonctions

**Source :**

- <https://github.com/DanYellow/presentations/blob/master/mmi/html/travaux-pratiques/numero-3/presentation.pdf>

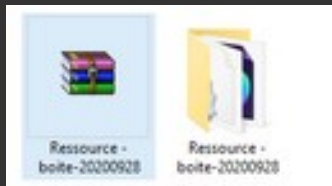
## Petit point sur les ressources de cours



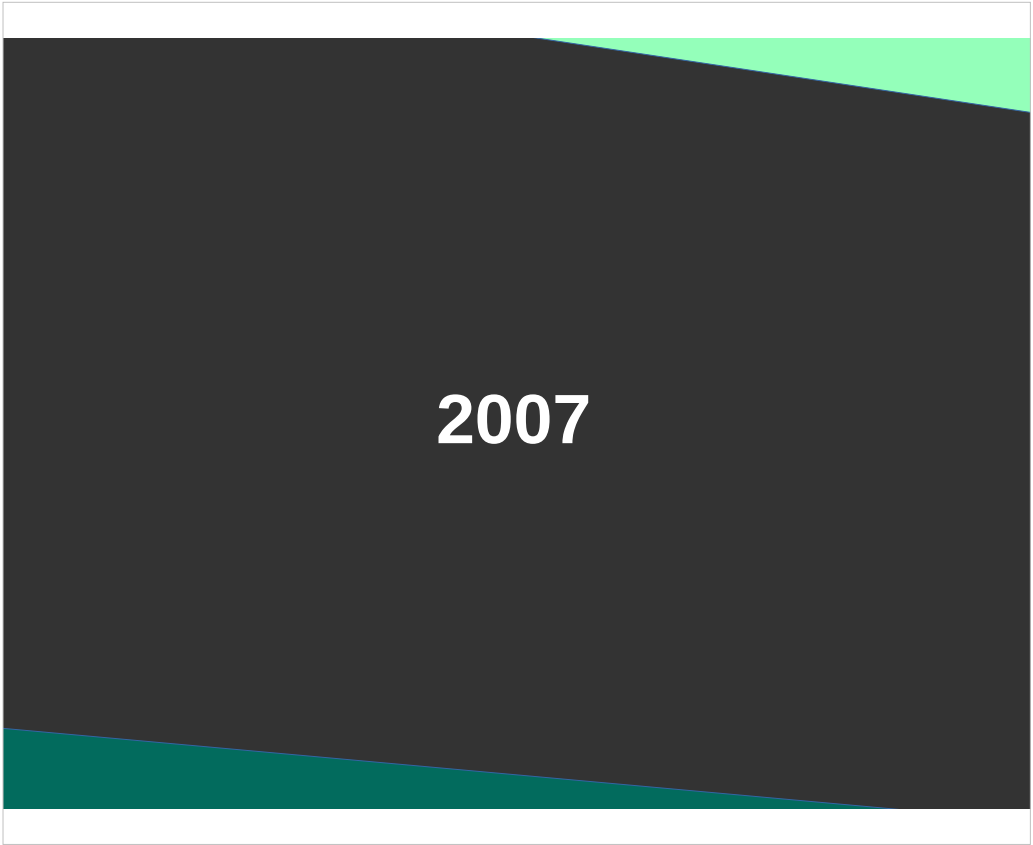
Menu contextuel sur l'archive (clic droit). Choisir un des choix surligné.

- 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

## 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**





- Apple étant développeur du moteur html "webkit", ceci lui a permis d'apporter beaucoup de fonctionnalités à html et css pour le 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

- Il est possible d'orienter l'écran sur PC de bureau, toutefois on ne le fait pas tout le temps et l'écran reste relativement grand, et ce changement n'est pas pris en compte
- Les utilisateurs mobiles peuvent accéder à Internet sur le réseau cellulaire, il faut donc limiter le plus possible les ressources lourdes pour la data et la batterie
- Les smartphones ont par défaut un appareil photo et un microphone.
- Attention sous Safari iOS, la permission n'est pas gérée par site mais l'application totale. Si un utilisateur refuse, ça sera le cas pour tous les sites dont le vôtre.

## 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

- Il est possible d'orienter l'écran sur PC de bureau, toutefois on ne le fait pas tout le temps et l'écran reste relativement grand, et ce changement n'est pas pris en compte
- Les utilisateurs mobiles peuvent accéder à Internet sur le réseau cellulaire, il faut donc limiter le plus possible les ressources lourdes pour la data et la batterie
- Les smartphones ont par défaut un appareil photo et un microphone.
- Attention sous Safari iOS, la permission n'est pas gérée par site mais l'application totale. Si un utilisateur refuse, ça sera le cas pour tous les sites dont le vôtre.

## Point technique – Définition / résolution

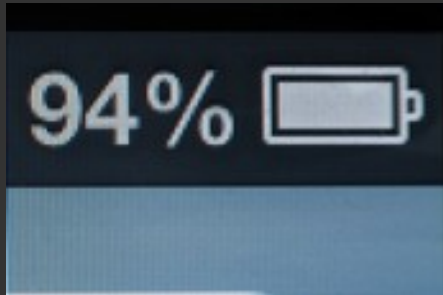
- La définition représente le nombre de pixels qu'un écran peut afficher
  - Exemple : une image 3000 x 2000 a une définition 6 millions de pixels
- La résolution représente la densité de pixels. Autrement dit le nombre pixels par pouce (1 pouce  $\approx$  2,54cm)
  - Plus la densité est élevée, plus les traits sont détaillés. Retina est le terme utilisé par Apple pour désigner les écrans haute résolution (apparition avec l'iPhone 4)

### Sources :

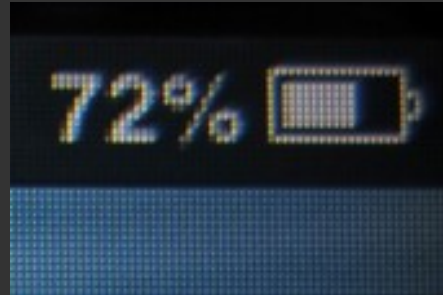
- [https://fr.wikipedia.org/wiki/D%C3%A9finition\\_d%27%C3%A9cran](https://fr.wikipedia.org/wiki/D%C3%A9finition_d%27%C3%A9cran)
- [https://www.frandroid.com/comment-faire/comment-fonctionne-la-technologie/342762\\_diff%C3%A9rence-entre-d%C3%A9finition-et-r%C3%A9solution](https://www.frandroid.com/comment-faire/comment-fonctionne-la-technologie/342762_diff%C3%A9rence-entre-d%C3%A9finition-et-r%C3%A9solution)
- <https://www.paintcodeapp.com/news/iphone-6-screens-demystified>

Une haute résolution d'écran octroie un confort de lecture notamment sur les petits écrans

## 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\\_diff%C3%A9rence-entre-d%C3%A9finition-et-r%C3%A9solution](https://www.frandroid.com/comment-faire/comment-fonctionne-la-technologie/342762_diff%C3%A9rence-entre-d%C3%A9finition-et-r%C3%A9solution)
- [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>

## Responsive design

- Terme inventé par Ethan Marcotte en 2010
- Désigne 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.alsacreation.com/article/lire/1615-cest-quoi-le-responsive-web-design.html>
- <https://responsivedesign.is/patterns/> - exemples basiques de mise en page responsive

- Par "chaque écran", il faut comprendre que les tailles les plus utilisées. D'ailleurs pour tablette prenez plus les iPad comme repère, elles sont plus populaires et les tailles sont plus standardisées.
- Pour l'adaptive design, la gestion des points de rupture (ou breakpoints) peuvent également être gérés côtés mobile, mais c'est une mauvaise idée



## **Responsive Web Design est le mot magique**

- Le terme Responsive (web) Design est le plus utilisé surtout de nos jours, les autres formes sont soit du passé soit peu utilisé niveau vocabulaire. Dites vous qu'on oppose le Responsive Web Design (RWD) au "Fixed" design

## 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 pas lui faire confiance
- Utilisation du javascript
  - Lourdeur potentielle des scripts

### 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

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

### Sources :

- <https://jakearchibald.com/2020/avif-has-landed/> - en
- <https://fr.wikipedia.org/wiki/WebP>
- <https://squoosh.app/> - pour compresser vos images

- Format vectoriel : Image composée de formules mathématique, l'image n'est jamais floue même zoomée à fond. S'oppose aux format matricielle.
- A noter que si le format webp n'a pas été très suivi, le format webm est très utilisé pour afficher des vidéos

## 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/>

- Format vectoriel : Image composée de formules mathématique, l'image n'est jamais floue même zoomée au maximum. S'oppose aux formats matriciels.
- A noter que si le format webp n'a pas été très suivi, le format webm est très utilisé pour afficher des vidéos

## 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
  - Pourcentage pour les longueurs
    - exemple : "width: 50%" ou "padding: 3%"
  - em/rem/pourcentage pour les tailles de police
    - exemple : "font-size: 1.5rem"
- 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)

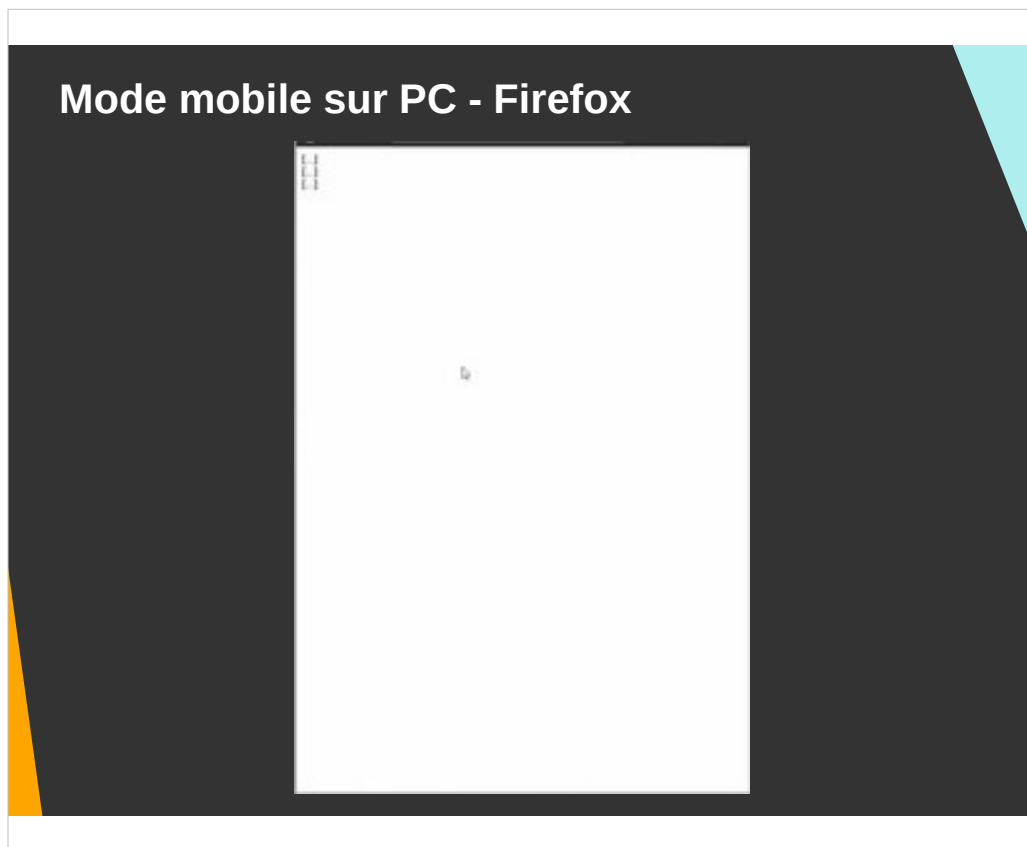
## Tester son site mobile sur PC

- 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
- 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

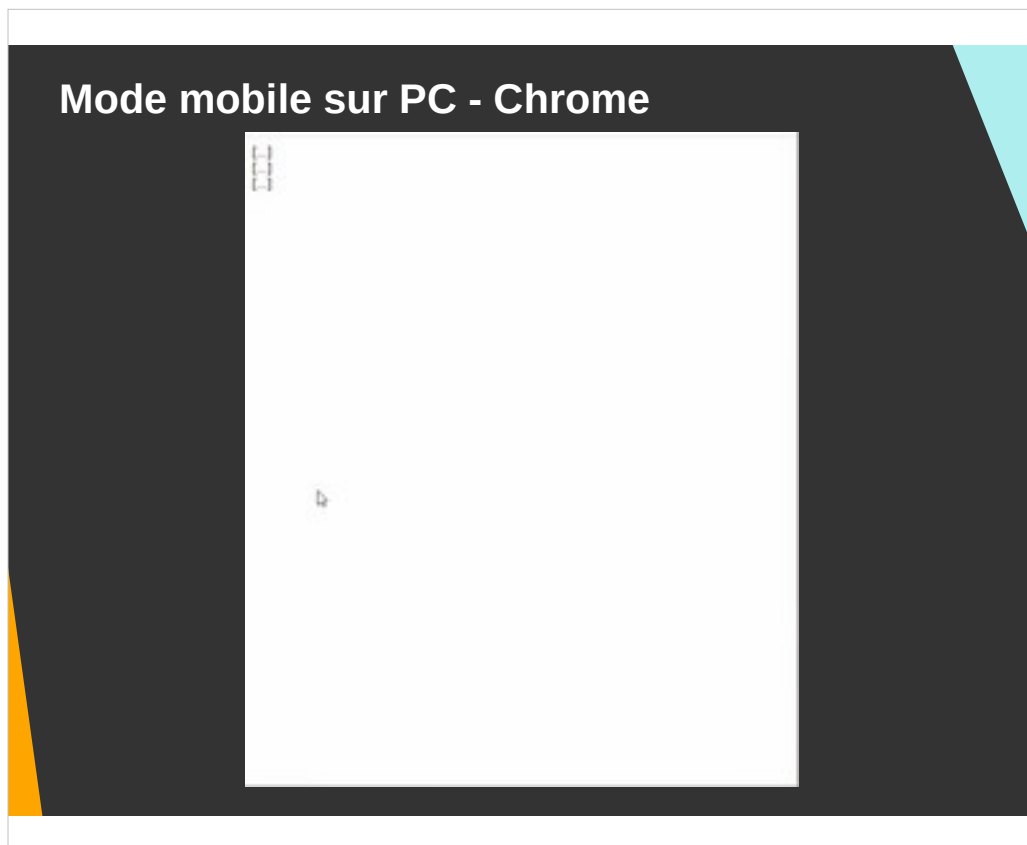
### Sources :

- <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

- Pour inspecter son site web sous Safari, il activer le mode développeur sous iPhone, iPad et MacOS. Les tout doit être connecté par câble USB.



- Pour inspecter son site web sous Safari, il activer le mode développeur sous iPhone, iPad et MacOS. Les tout doit être connecté par câble USB.



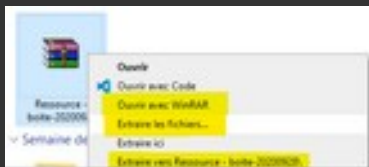
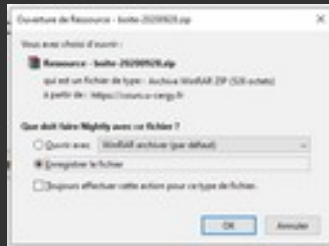
- Pour inspecter son site web sous Safari, il activer le mode développeur sous iPhone, iPad et MacOS. Les tout doit être connecté par câble USB.
- Pour android :  
<https://developers.google.com/web/tools/chrome-devtools/remote-debugging>



**Ressource du cours sur ENT**

**Travaux Pratiques #4 > "Ressource – Images  
responsives"**

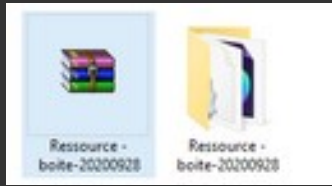
## Petit point sur les ressources de cours



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

- 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

## Petit point sur les ressources de cours



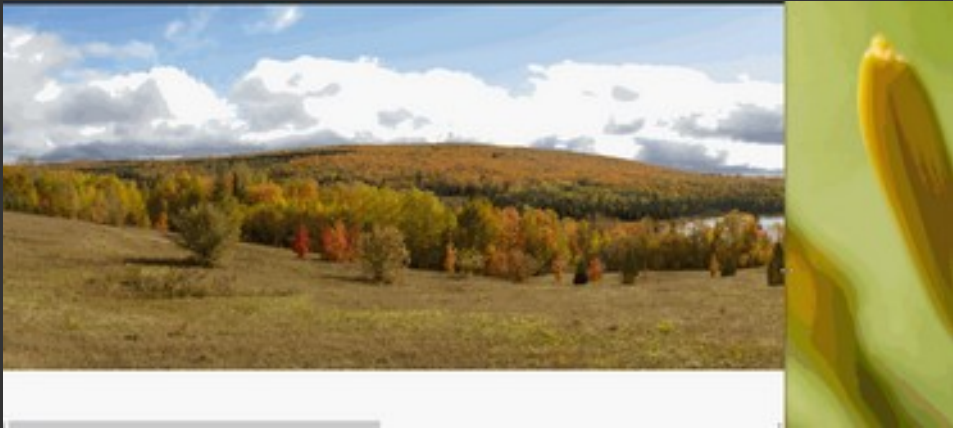
- Une fois extrait, vous devez obtenir un dossier avec le contenu de l'archive
- Libre à vous de travailler dans le dossier OU en copier le contenu pour le coller ailleurs
- **En aucun vous devez travailler dans l'archive**
- Une fois fait, glissez le dossier dans VS Code

## Pratiquons ! - Notre image responsive

Pré-requis :

- Avoir la ressource "images-responsives"
- Ouvrir le fichier "srcset.html" dans votre éditeur de code
- Ajouter une balise `<img >` avec l'image "panorama.jpg" présente dans le dossier "assets/images"

## Pratiquons ! - Notre image responsive



Notre image s'affiche, mais on a un défilement horizontal

## Pratiquons ! - Notre image responsive

Pré-requis :

- Avoir la ressource "images-responsives" et réalisé la pratique précédente
- Ajouter une classe "img-responsive" dans un fichier css avec les déclarations suivantes :

**max-width: 100 %;**  
**height: auto;**

- Ajouter la classe à l'image
- Redimensionnez votre fenêtre, votre image s'adapte maintenant à votre fenêtre



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



Afficher image de 2000px, même correctement, sur un écran de 320 de largeur, n'a absolument aucun sens, alors le W3C a apporté une solution. Les smartphones les plus grands ont une largeur de 450px.



## 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.alsacreation.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 drapsienne 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.alsacreation.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.alsacreation.com/article/lire/1621-responsive-images-srcset.html>

- La taille de l'image pour la condition donnée ne peut être qu'exprimée qu'en pixels ou vw/vh
- La dernière valeur de l'attribut ne doit pas avoir de condition sur le média, c'est la valeur par défaut

## 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.alsacreation.com/article/lire/1621-responsive-images-srcset.html>

- Vous n'avez pas besoin de savoir exactement la densité de pixels des écrans, le navigateur là encore va trancher et prendre la valeur du descripteur la plus proche

## 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.alsacreation.com/article/lire/1621-responsive-images-srcset.html>

- Vous n'avez pas besoin de savoir exactement la densité de pixels des écrans, le navigateur là encore va trancher et prendre la valeur du descripteur la plus proche

## 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 par tous les navigateurs présentement**
- 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"**

taille de l'image pour la condition donnée

condition sur le média (media query)

### Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-srcset>
- <https://www.alsacreation.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.  
Si et seulement si le descripteur est de largeur

### Sources :

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


## Pratiquons ! - Attributs srcset / sizes

Pré-requis :

- Avoir la ressource "images-responsives"
- Les consignes sont dans la ressource, dans la page srcset.html

```

```



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

## 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
- 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>

- Même si votre image de rechange (fallback) n'est pas utilisée, elle est obligatoire. Car sans elle vous ne pouvez pas appliquer de style sur votre image
- Il est souvent dit que la balise `<picture>` doit être utilisée pour des raisons artistiques. Exemple mon image est un grand plan, si le sujet apparaît très bien sur desktop, sur mobile c'est plus compliqué

## Balise picture

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

### Sources :

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

- Même si votre image de rechange (fallback) n'est pas utilisée, elle est obligatoire. Car sans elle vous ne pouvez pas appliquer de style sur votre image
- Il est souvent dit que la balise `<picture>` doit être utilisée pour des raisons artistiques. Exemple mon image est un grand plan, si le sujet apparaît très bien sur desktop, sur mobile c'est plus compliqué

## 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

- Il est précisé "dans le cas de la balise <picture>" car la balise <source> est aussi utilisée pour les vidéos et l'audio. Toutefois les propriétés ne sont pas les mêmes.

## Balise <picture> – En application

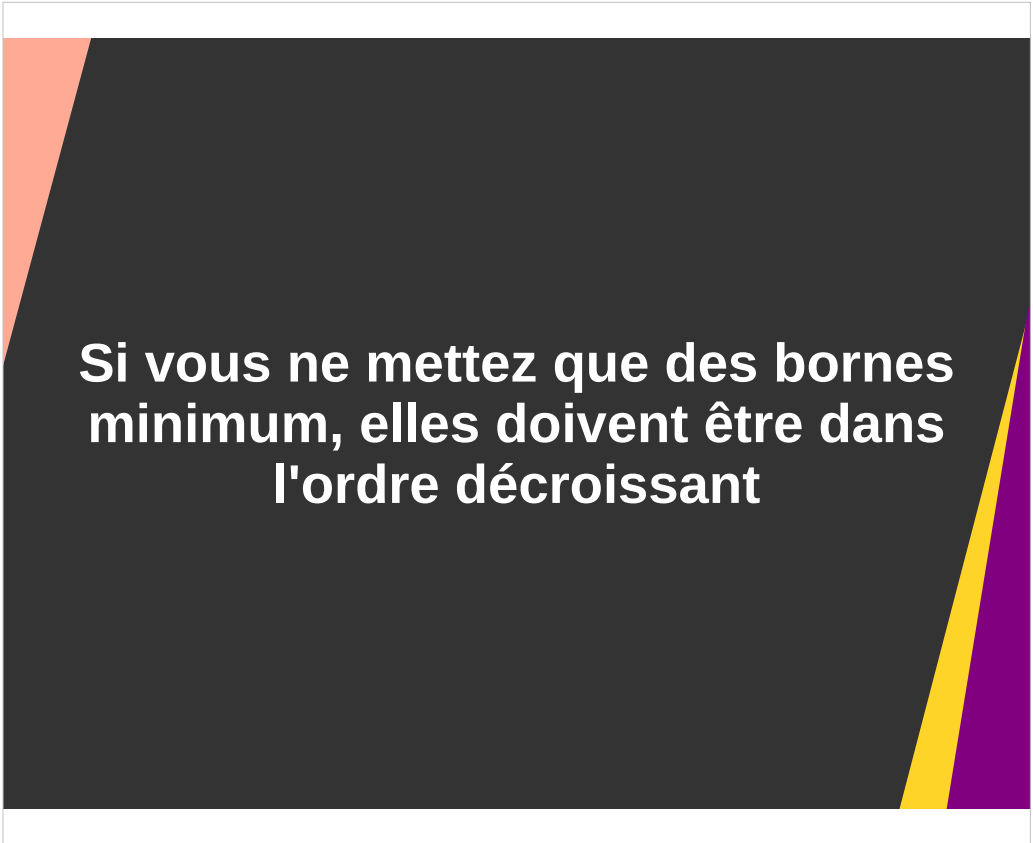
```
<picture>
  <source srcset="image-medium.jpg" media="(max-
width: 399px)" />
  <source srcset="image-mini.jpg" media="(min-width:
400px) and (max-width: 768px)"/>
  
</picture>
```

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

```
<picture>
  <source srcset="image-mini.jpg"
media="(min-width: 320px)" />
  <source srcset="image-large.jpg"
media="(min-width: 600px)"/>
  
</picture>
```



**Si vous ne mettez que des bornes  
minimum, elles doivent être dans  
l'ordre décroissant**



## Pratiquons ! - Balise <picture>

Pré-requis :

- Avoir la ressource "images-responsives"
- Compléter la structure <picture> présente dans la ressource et gérer trois largeurs d'écran dans le fichier picture.html
  - Largeur écran  $\leq 767$ px (smartphones)
  - $768$ px  $\leq$  largeur écran  $\leq 1024$ px
  - $1025$ px  $\leq$  largeur écran  $\leq 1099$ px
- Utiliser les images contenues dans le dossier "assets/images/balise-picture/"
- Plus de détails dans le fichier "picture.html" de la pratique

```

```

## Pratiquons ! - Balise <picture>



La syntaxe (min-width: 600px) est souvent utilisée dans une autre fonctionnalité du html 5 : les media queries

## Balise picture – Combiner le tout

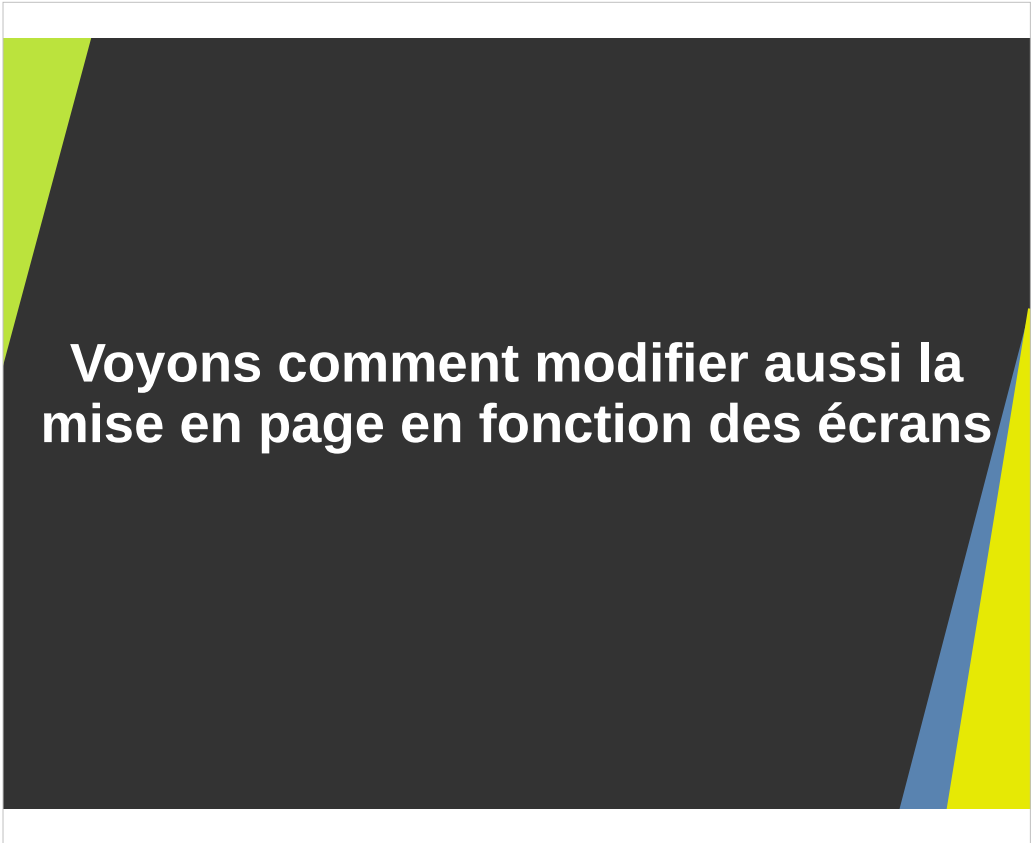
```
<picture>
  <source
    media="(max-width: 1113px)"
    srcset="image-pc@2x.png 2x, image-pc@2x.png 1x"
  />
  <source
    media="(max-width: 767px)"
    srcset="image-mobile@3x.png 3x, image-mobile@2x.png 2x, image-mobile@1x.png 1x"
  />
  
</picture>
```

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

- Il est précisé "dans le cas de la balise <picture>" car la balise <source> est aussi utilisée pour les vidéos et l'audio. Toutefois les propriétés ne sont pas les mêmes.



**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.alsacreation.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 de pourcentage
  - 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.alsacreation.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'applique 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)

On précise bien écran, car il est possible de remplacer "screen" par "print" pour désigner les imprimantes. Retirer "screen" implique que la media query s'applique aussi bien sur les écrans que les imprimantes



## 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_@)

- Ici on définit que ces règles sur tous les médias ayant une hauteur de 440px ou orientés en portrait
- A noter qu'en CSS4, la virgule pourra être remplacée par le mot-clé "or", plus explicite

## 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_@)

- Ici on définit que ces règles sur tous les médias ayant une hauteur de 440px ou orientés en portrait
- A noter qu'en CSS4, la virgule pourra être remplacée par le mot-clé "or", plus explicite

## Pratiquons ! - Media queries css

Pré-requis :

- Avoir la ressource "media-queries"
- Éditer la règle @media de façon à ce que la taille du titre "Ma galerie d'images" change sur un appareil dont la largeur d'écran est inférieure ou égale à 640px
- Ajouter une règle @media pour gérer un écran dont la taille est supérieure ou égale à 400px et appliquer des déclarations sur n'importe quel sélecteur CSS
  - Ex : ajouter des marges latérales
- Ajouter une règle @media pour gérer un écran ayant l'orientation paysage (landscape en anglais)

```

```

## 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>

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

## Pratiquons ! - Media queries dans le html

Pré-requis :

- Avoir la ressource "media-queries"
- Avoir réalisé la pratique "media queries css"
- Lister le nombre de règles @media contenues dans le fichier style.css
- Créer autant de fichiers css qu'il y a de règles @media
- Lier le fichier index.html aux fichiers css (ne pas oublier les attributs media)
- Placer le contenu des règles @media dans les bons fichiers

```

```

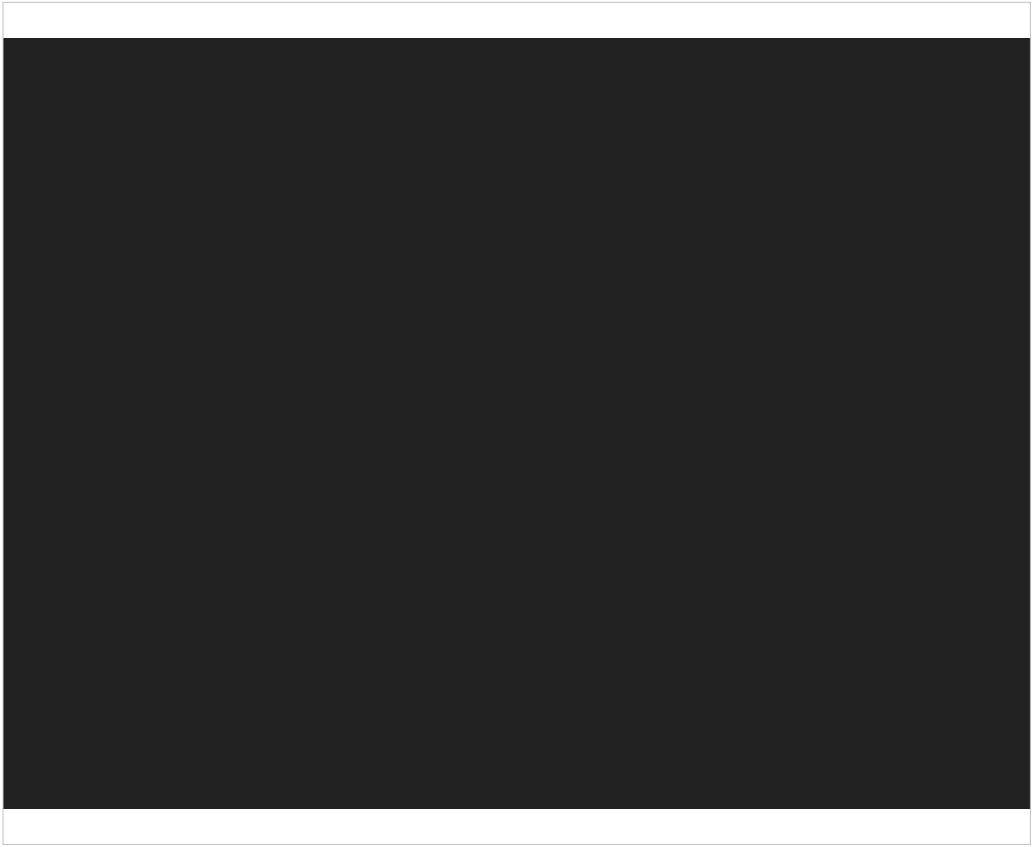
A dark gray rectangular slide with a white border. On the left side, there are two overlapping triangles: a cyan one in front of a green one. On the right side, there is a yellow triangle. The text is centered in white.

# Quiz

**"Test – Responsive et mobile" sur ENT**

Conditions du test :

- 10 minutes
- 1 tentative



**Questions ?**