

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

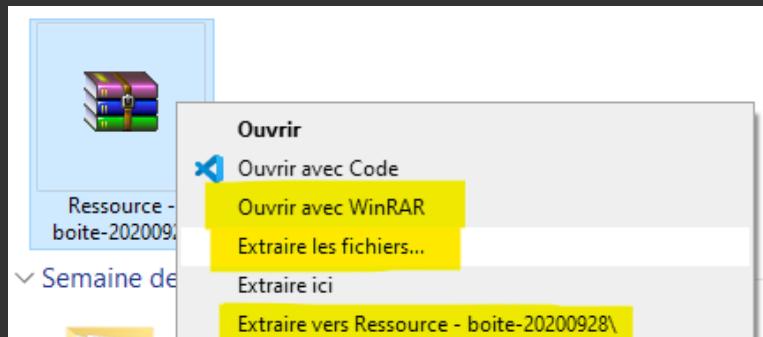
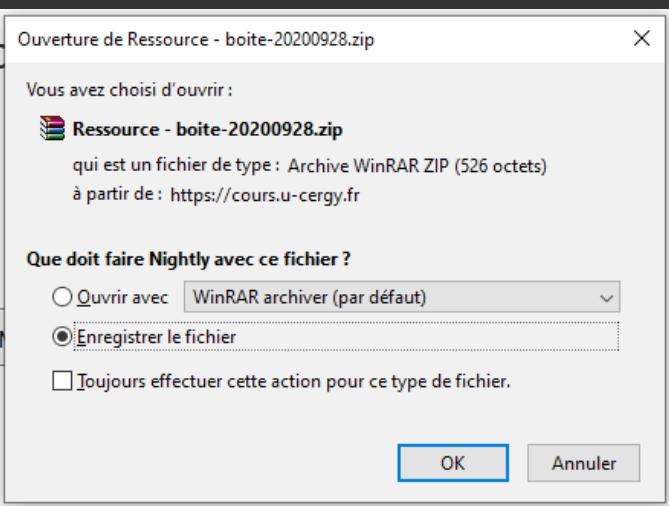
Récapitulatif du tp précédent

- 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>

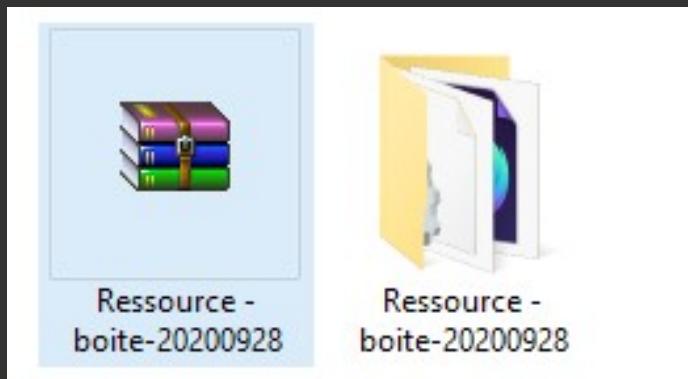
Petit point sur les ressources de cours



- 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**

Ressource du cours

**Travaux Pratiques #4 > "Ressource - Responsive" sur
ENT**

2007

iPhone



- 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

iPhone 11 Pro Max

Smartphone / tablette

- Points à prendre en compte sur ces appareils :
 - Représente la majorité de l'audience web
 - Écran 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
- Représente la majorité de la fréquentation web
- Définition élevée de nos jours
- Notifications et Push
- [...]

Source :

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

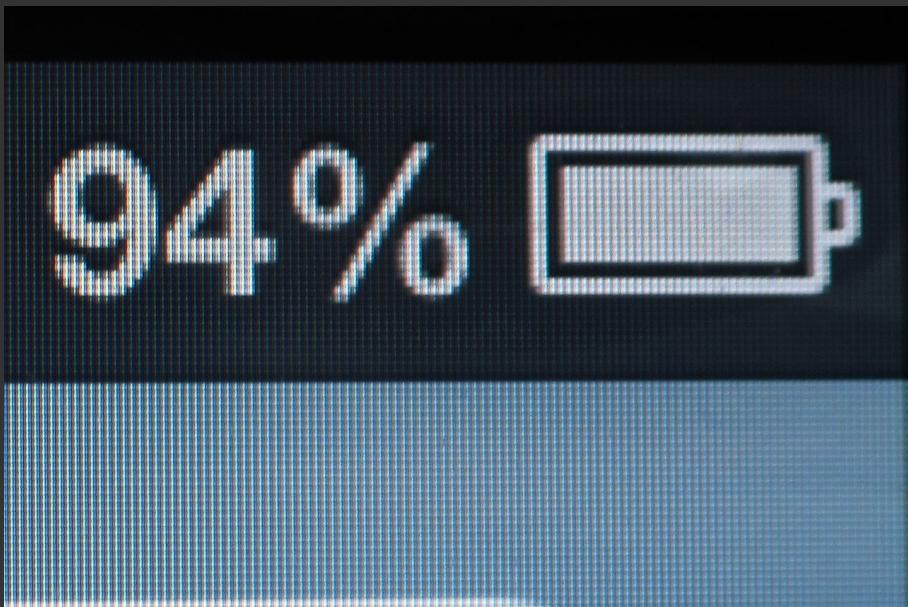
Point technique – Définition / résolution

- La définition représente le nombre de pixels (ou points) 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 = 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)

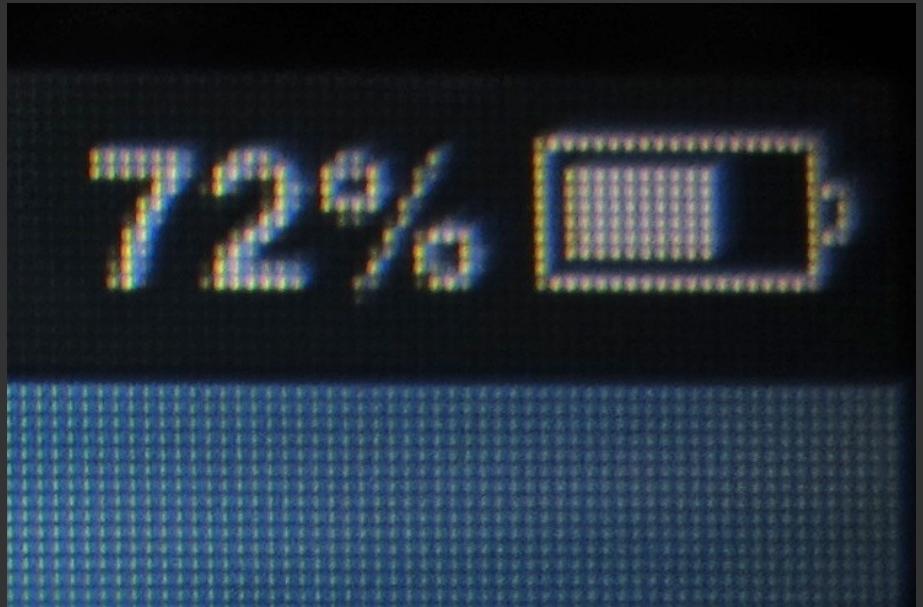
Sources :

- https://fr.wikipedia.org/wiki/D%C3%A9finition_d%C2%7C%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)

Sources :

- https://fr.wikipedia.org/wiki/D%C3%A9finition_d%C2%A7%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>

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
 - Utilisation des media queries
 - **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
- <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>



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
- 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
- <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
- 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 belle) 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

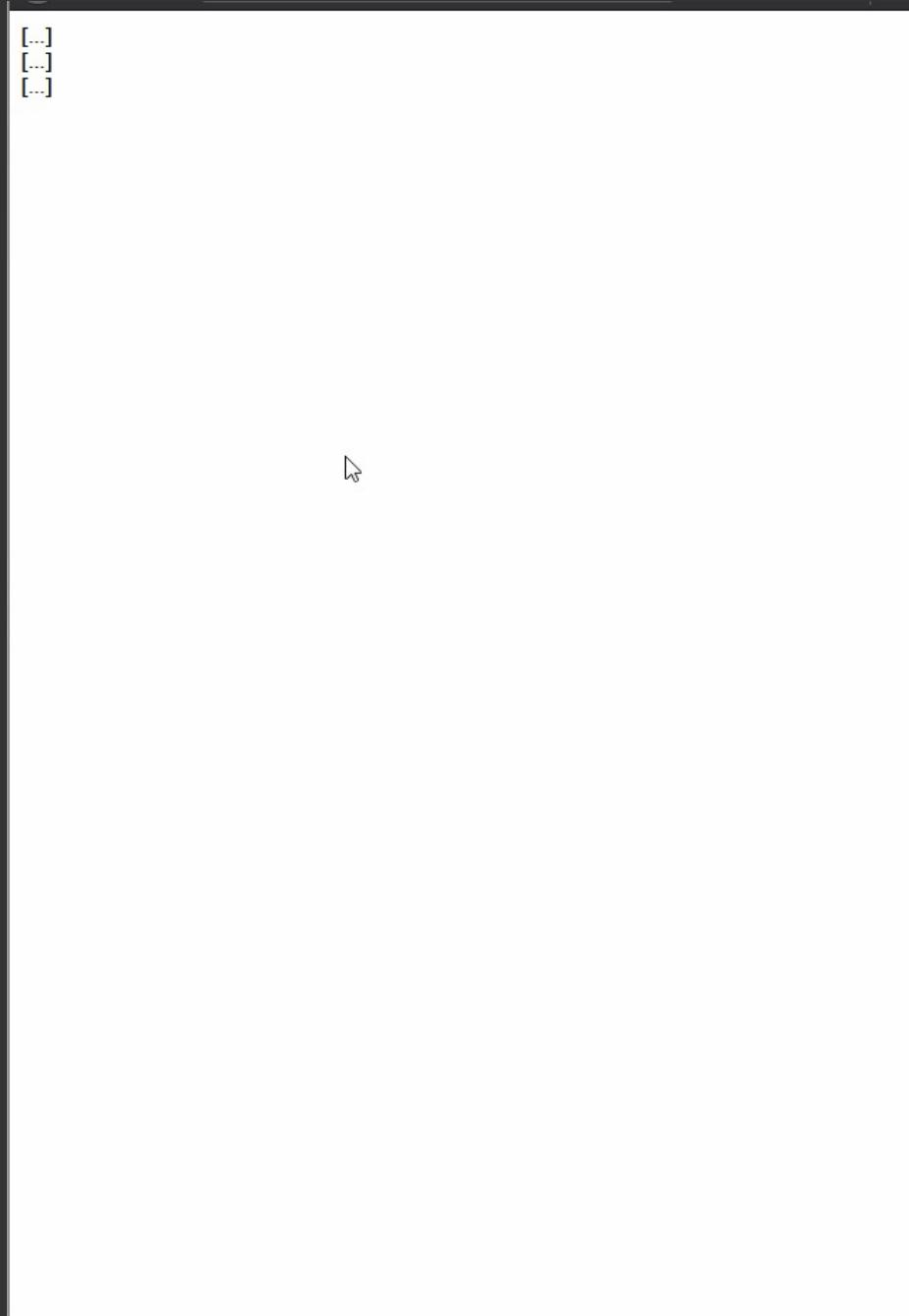
Tester son site mobile sur PC

- 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 l'inspecteur de Safari
- Raccourci clavier : ctrl + shift (maj) + m (Chrome / Brave / Firefox)

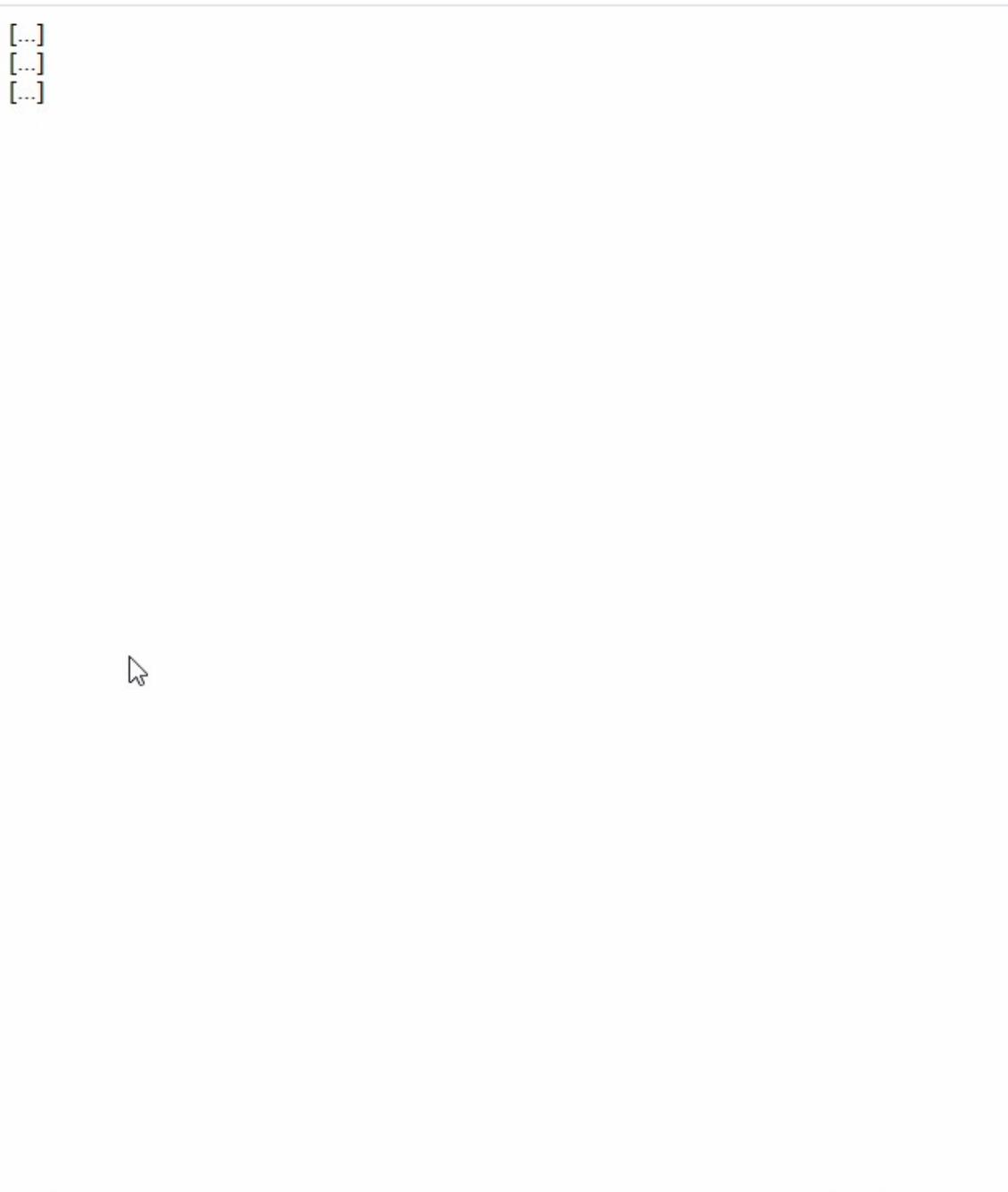
Sources :

- <https://www.idownloadblog.com/2019/06/21/how-to-use-safari-web-inspector-ios-mac/> - en

Mode mobile sur PC - Firefox



Mode mobile sur PC - Chrome

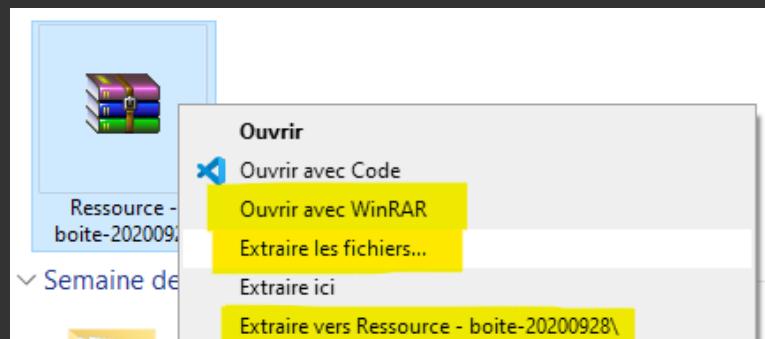
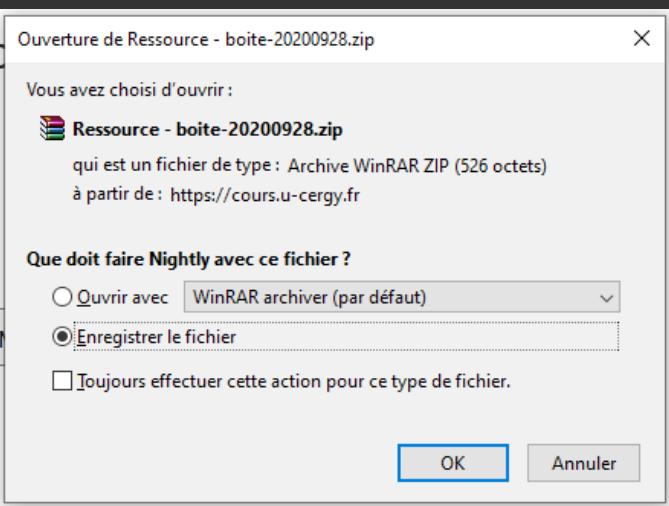


[...]
[...]
[...]

Ressource du cours sur ENT

Travaux Pratiques #4 > "Ressource – Images responsives"

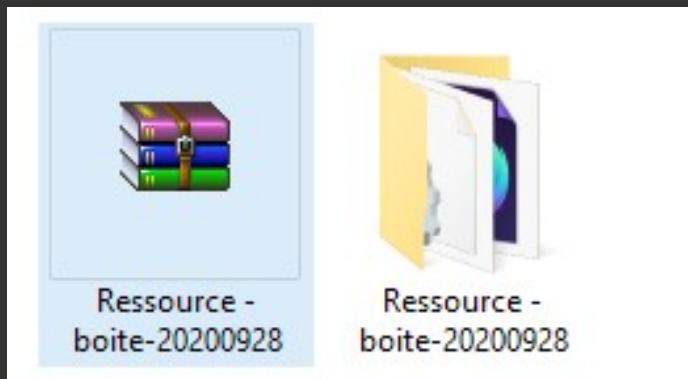
Petit point sur les ressources de cours



- 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



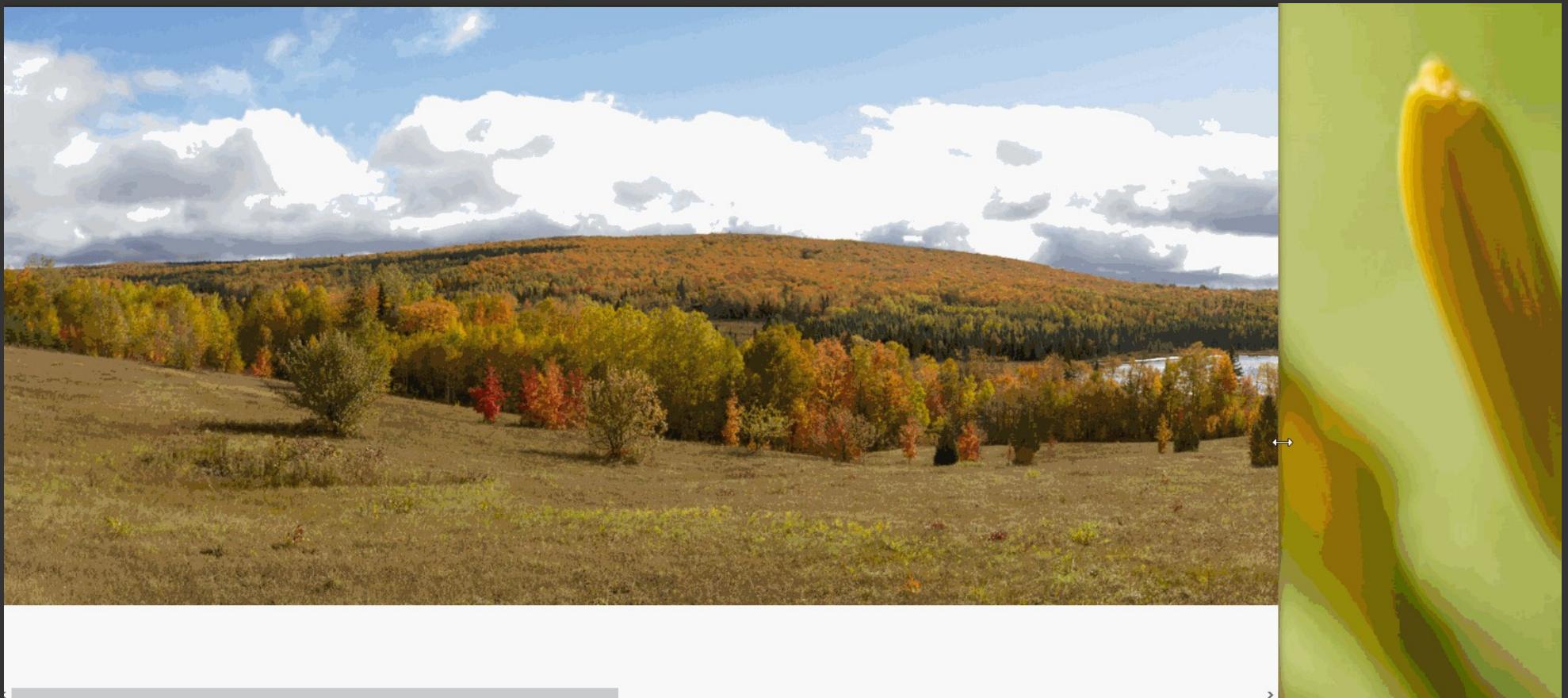
- Une fois extrait, vous devez obtenir un dossier avec le contenu de l'archive
- Libre à vous de travailler dans le dossier OU en copier le contenu pour le coller ailleurs
- **En aucun cas vous devez travailler dans l'archive**
- Une fois fait, glissez le dossier dans VS Code

Pratiquons ! - Notre image responsive

Pré-requis :

- Avoir la ressource "images-responsives"
- Ajouter une balise avec une image relativement large (+2000px)
- Redimensionnez votre fenêtre, votre image s'adapte maintenant à votre fenêtre

Pratiquons ! - Notre image responsive



Notre image s'affiche, mais on n'a un vilain scroll horizontal

Pratiquons ! - Notre image responsive

Pré-requis :

- Avoir la ressource "images-responsives"
- Avoir une balise avec une image relativement large
- 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

Attributs srcset / sizes et balise <picture>

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

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

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

The diagram illustrates the structure of the `srcset` attribute. It shows a green `srcset` attribute with a pink box around the value `"mon-imageHD.jpg 2x"`. This value is divided into two parts: `mon-imageHD.jpg` and `2x`. A blue bracket labeled "descripteur (ici de densité)" spans both parts. A pink bracket labeled "source de mon image" encloses the first part, `mon-imageHD.jpg`.

```
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://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-sizes>

Attribut srcset – descripteur

- Descripteur "x"
 - Désigne la densité de pixel de l'écran
 - 1x représente un écran normal / 2x un écran HD...
 - Accepte un nombre décimal. Exemple "3x"
- 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>

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 et si tu ne connais pas l'attribut srcset affiche "mon-image.jpg"

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-srcset>

Attribut sizes

- Lié à la balise
- 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)

taille de l'image pour la
condition donnée

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-srcset>

Pratiquons ! - Attributs srcset / sizes

Pré-requis :

- Avoir la ressource "images-responsives"
- Ajouter une balise `` avec les attributs `srcset` et `src`
- Mettre comme valeur pour l'attribut `src`, une image du dossier "assets/images/attribut-srcset/"
- Définir la valeur de l'attribut `srcset` avec autant de valeurs qu'il y a d'images dans le dossier "assets/images/attribut-srcset/" avec le descripteur de densité (x) jusqu'à 3x
- Tester avec le mode responsive du navigateur (ctrl + maj + m sur firefox)

Notre image change maintenant en fonction de l'écran

Pratiquons ! - Attributs srcset / sizes

Pré-requis :

- Avoir la ressource "images-responsives"
- Avoir une balise avec les attributs srcset et src
- Remplacer les descripteurs de densité par des descripteurs de largeur avec comme valeur le nombre contenu dans le nom de l'image
 - Ex : image-750.png → 750w
- Ajouter l'attribut sizes avec des media queries et une taille
 - Ex : (max-width : 480px) 100vw
 - Ne pas oublier la valeur par défaut
- Tester avec le mode responsive du navigateur (ctrl + maj + m sur firefox)

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

Balise picture

- Utilise la balise comme valeur par défaut
- Fonction de pair avec la balise <source/>
 - Fonctionne avec les attributs srcset et media
- A utiliser si l'image doit changer (rognage par exemple) en fonction du périphérique
- Permet de gérer différent format 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
 - type : définit le format de l'image (au format MIME)
 - Il faut au moins l'attribut "media" ou "type" avec l'attribut "srcset"
- Une balise <source> par image

Une fois tout ça définit, le navigateur décide tout seul

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>

Balise <picture> – En application

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

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 et si aucune condition est remplie ou que tu ne connais pas la balise source charge la "mon-image.jpg"

Sources :

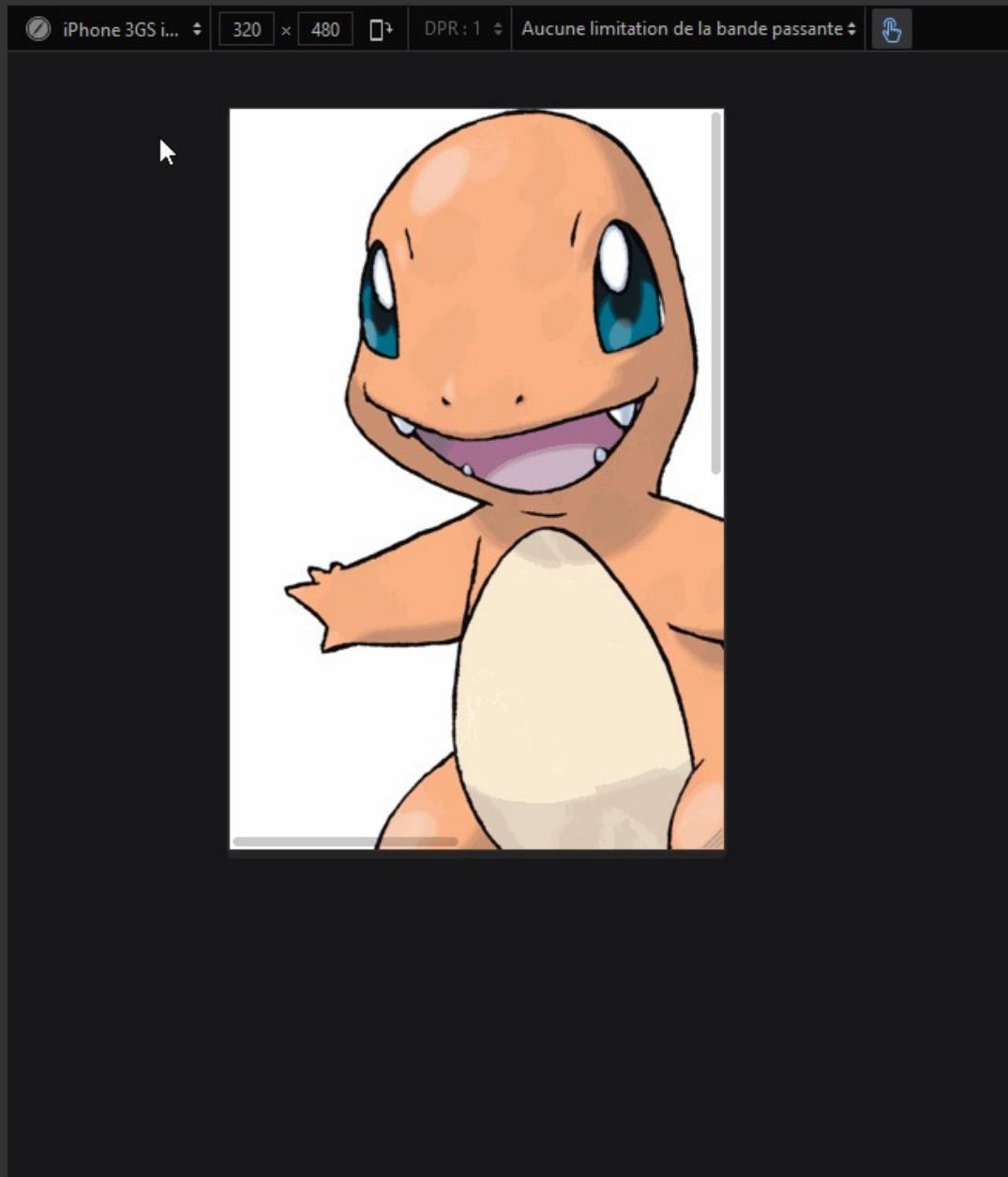
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/picture>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source#attr-media>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source#attr-srcset>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/source#attr-type>

Pratiquons ! - Balise <picture>

Pré-requis :

- Avoir la ressource "images-responsives"
- Ajouter la structure de code pour la structure <picture> (il y a un début de structure dans les notes de la slide précédente)
 - Gérer 3 résolutions
 - 320px (iPhone 5 / SE)
 - 640px (Samsung GS 10 / iPhone X)
 - 768px (iPad en mode portrait)
- Utiliser les images contenues dans le dossier "assets/images/balise-picture/"

Pratiquons ! - Balise <picture>



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"
- Gestion d'opérateurs logiques
 - and / only / not
 - Plusieurs conditions peuvent être appliquées
- Au service des terminaux mobiles
 - Ajustement du design pour ces terminaux
- Peut être géré directement dans le css ou le html via la balise <link />

Sources :

- 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/R%C3%A8gles_@
- https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%A9tes_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 séparé par une virgule

```
v @media (max-height: 440px), (orientation: portrait) {  
v   .titre-principal {  
v     color: pink  
v   }  
}  
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%A9gles_@

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%A9gles_@

Pratiquons ! - Media queries css

Pré-requis :

- Avoir la ressource "media-queries"
- Éditer la règle @media de façon à ce que le texte s'affiche plus gros sur un appareil dont la largeur d'écran est inférieure ou égale à 320px
- Ajouter une règle @media pour gérer un écran dont la taille est supérieure ou égale à 400px et appliquer des déclarations
 - Ex : ajouter des marges latérales

Pratiquons ! - Media queries css

- Ajouter une règle @media pour gérer un écran ayant l'orientation paysage (landscape en anglais)
- Éditer / ajouter des règles @media pour changer la mise en page de la page en fonction de vos envies

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 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 :

- 7 minutes
- 3 tentatives
- Meilleure des notes prise en compte

Questions ?