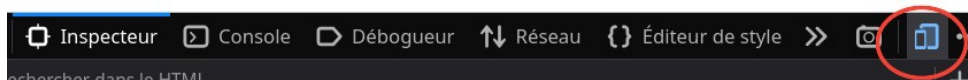


# CSS

## Reponsive design

### Utiliser les outils de développement

Simuler les différentes tailles d'écran en utilisant les outils de développement



### Définir les points de rupture

Les points de rupture sont des largeurs à partir desquelles l'affichage change.

#### Exemple



Affichage du menu pour les écrans  $\geq 1024\text{px}$ . Affichage d'un menu burger pour les écrans  $< 1024\text{px}$ .

#### Points de ruptures fréquents

Les points de rupture sont généralement entre 360px et 1920px.

### Media Query

Ce sont des règles css appliquée en fonction de certaines conditions, en général, la largeur de l'écran.

Exemples

```
@media (min-width: 1024px) {
  h1 {
    font-size: 6rem;
  }
  nav#site-navigation .menu {
    display: block;
    margin: 1rem;
  }
}
```

Les instructions « @media » doivent être positionnées après les instructions css « générales »

### Balise méta viewport

Elle indique au navigateur comment contrôler les dimensions et la mise à l'échelle de la page . Elle garanti que l'affichage sur l'appareil sera bien conforme au code produit.

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

### Approche mobile first

Cette approche consiste à concevoir un site web en priorité pour les appareils mobiles. Il s'agit simplement d'écrire le code css pour les smartphone, puis de créer les médias queries pour les résolutions plus larges

```
/* MOBILE (+ TABLET PORTRAIT) */
#demo {
  width: 100%;
}
/* TABLET LANDSCAPE */
@media (min-width: 768px) {
  #demo {
    width: 80%;
  }
}
/* LAPTOP */
@media (min-width: 1024px) {
  #demo {
    width: 50%;
  }
}
/* DESKTOP */
@media (min-width: 1280px) {
```

```
#demo {
    width: 25%;
}
```

### Pour aller plus loin

anatomy des media query : types de média, opérateur

<https://www.alsacreations.com/article/lire/930-css3-media-queries.html>

conteneur query

<https://www.alsacreations.com/article/lire/1915-Les-Container-Queries-en-CSS.html>

prefers-color-scheme

La caractéristique média prefers-color-scheme permet de détecter les préférences exprimées par l'utilisateur quant au thème à utiliser (sombre ou clair).

<https://developer.mozilla.org/fr/docs/Web/CSS/@media/prefers-color-scheme>

## Police responsive

Utiliser des tailles de police en unité « rem ».

Les liseuses, les personnes âgées, les personnes ayant des problèmes de vue ont souvent l'habitude d'augmenter la taille des caractères dans le browser en faisant: Paramètres --> Apparence --> Taille des polices (Moyenne par défaut).

Pour donner à l'utilisateur la possibilité de faire un choix de taille (plus grande ou moins grande) il faut déclarer la taille de la police en pourcents au niveau de l'élément html.

Comme les navigateurs ont une taille de caractère par défaut de 16px, il est commode d'appliquer à l'ensemble de la page une taille de 62,5 %, soit 10px. Ainsi, lorsqu'on dimensionne un texte en unité « rem », la conversion est très facile. Exemple : 2.2rem équivaut à 22px.

```
html{
    font-size:62.5%; /* correspond à 10px/16px*/
}
```

[https://raphaelgoetter.wordpress.com/2016/07/05/lastuce-du-font size-62-5-oui-non-ou-ca-depend/](https://raphaelgoetter.wordpress.com/2016/07/05/lastuce-du-font-size-62-5-oui-non-ou-ca-depend/)

## Image responsive

### Fixer l'image à 100 %

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

### Charger des images différentes en fonction des tailles d'écran

attributs HTML srcset et sizes

<https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>

<https://www.pierre-giraud.com/html-css-apprendre-coder-cours/images-responsive/>

### Plus de conseils

<https://web.dev/learn/design/responsive-images?hl=fr>

## Vidéo responsive

<https://dessinemoiunsite.com/comment-transformer-vos-videos-en-responsive/>

<https://fr.eitca.org/web-development/eitc-wd-hcf-html-and-css-fundamentals/multimedia-in-html-and-css/inserting-html5-videos-and-embedding-external-videos/examination-review-inserting-html5-videos-and-embedding-external-videos/how-can-we-make-a-video-responsive-in-html-and-css/>

## Astuces

### Moins de média query avec Grid

<https://www.gerits-aurelien.be/fr/pages/28-creer-des-vignettes-carrees-avec-css-grid/>

### Positionnement absolu et responsive

Anchor positioning (ne fonctionne pas encore avec tous les navigateur)

<https://ishadeed.com/article/anchor-positioning/#intro>

<https://caniuse.com/?search=position-anchor>

## Ressources

[https://developer.mozilla.org/fr/docs/Learn\\_web\\_development/Core/CSS\\_layout/Responsive\\_Design](https://developer.mozilla.org/fr/docs/Learn_web_development/Core/CSS_layout/Responsive_Design)

<https://css-tricks.com/a-complete-guide-to-css-media-queries/>

## Exercices

**Récupérer les dossiers zippés. Ils contiennent des bases de code html-css.**

**Chaque dossier d'exercice contient un fichier readme.md qui contient les instructions à suivre.**

- newzletter.zip
- takeoff.zip
- École-de-musique.zip
- archeologie.zip
- login.zip