Adapter son site aux écrans mobiles

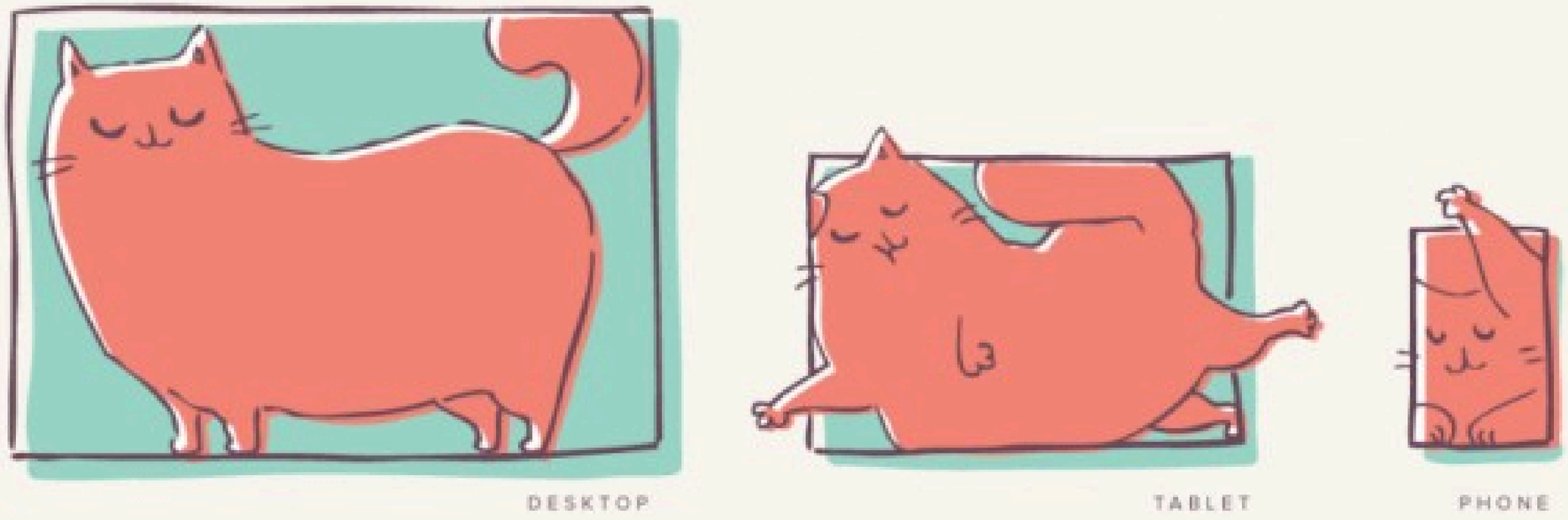
Objectif: découvrir les mediaqueries et leur fonctionnement

TOUCHET Laurent / Afec 2025

C

Premiers pas avec CSS

1. **Qu’est-ce que le responsive design ?**
2. **Approche Mobile-First vs. Desktop-First**
3. **Les media queries**
4. **La fonction clamp()**
5. **Bonnes pratiques en Responsive Design**
6. **Ressources**



RESPONSIVE DESIGN

Qu’est-ce que le responsive design ?

Le responsive design permet de créer des sites web qui s'adaptent automatiquement à toutes les tailles d'écrans (mobile, tablette, desktop). Plutôt que de concevoir plusieurs versions d’un site, on utilise des techniques CSS pour adapter l'affichage.

Pourquoi utiliser le responsive design ?

- Une **meilleure expérience utilisateur** sur tous les appareils.
- **Optimisation SEO** : Les moteurs de recherche privilégient les sites adaptés aux mobiles.
- **Économie de temps et de ressources** : Un seul site au lieu de plusieurs versions.

Approche Mobile-First vs. Desktop-First

Il existe deux approches principales pour concevoir un site responsive :

- **Mobile-First** (Recommandé ✅)
- **Desktop-First** (Moins recommandé ❌)

Mobile-First

- On commence par concevoir le site pour les **mobiles** (petits écrans).
- Ensuite, on ajoute des styles pour les **écrans plus grands** (tablettes, ordinateurs).
- **Avantage** : Plus rapide et adapté aux usages actuels (majorité du trafic sur mobile).

```
body {
  font-size: 16px;
  background-color: lightgray;
}

@media (min-width: 768px) {
  body {
    background-color: white;
  }
}
```

Les media queries

Les media queries permettent d'appliquer des styles CSS uniquement lorsque certaines conditions sont remplies (largeur d'écran, résolution, orientation, etc.).

Syntaxe de Base

```
@media (condition) {
  /* Styles appliqués si la condition est remplie */
}
```

Adapter selon la largeur d'écran

```
@media (max-width: 600px) {
  body {
    background-color: blue;
  }
}
```

Si l'écran fait 600px ou moins, le fond devient bleu.

```
@media (max-width: 480px) { /* Smartphones */ }
@media (max-width: 768px) { /* Tablettes */ }
@media (max-width: 1024px) { /* Petits ordinateurs */ }
@media (min-width: 1025px) { /* Grands écrans */ }
```

https://www.afecdax.ovh/ressources/dev/css/css_demo_responsive.html

Les media queries

Les media queries permettent d'appliquer des styles CSS uniquement lorsque certaines conditions sont remplies (largeur d'écran, résolution, orientation, etc.).

Adapter selon l'orientation de l'écran

```
@media (orientation: portrait) {
  body {
    background-color: yellow;
  }
}
```

Si l'appareil est en mode portrait, le fond devient jaune.

La fonction clamp()

clamp() est une **fonction CSS** qui va vous permettre d'écrêter une valeur entre deux bornes inférieure et supérieure.

Imaginons que vous souhaitiez avoir **un titre <h1>** responsive, vous voulez qu'il soit **petit sur les petites résolutions** et **grand sur les grandes résolutions**.

clamp(valeur_min**,** valeur_a_tester**,** valeur_max**);**

- une valeur minimale
- la valeur à tester, aussi appelée "valeur préférée"
- et une valeur maximale.

https://codepen.io/RatonCodeur/pen/MWXGOKw

https://www.swebdev.fr/blog/la-fonction-css-clamp

Les bonnes pratiques en responsive design

✅ Utiliser **meta viewport.** Ajoutez cette ligne dans <head> pour que le site s'adapte correctement aux écrans :

- <meta name="viewport" content="width=device-width, initial-scale=1.0">

✅ **Éviter les tailles fixes** : utilisez **%**, **vw** (viewport width) ou **em** au lieu de **px** pour des mises en page flexibles.

✅ Utiliser **flexbox** et **grid** pour des mises en page adaptatives

✅ **Tester son site sur plusieurs appareils**

- Outils Chrome/Firefox : F12 (Mode appareil)
- Sites en ligne : BrowserStack, Responsinator
- Test réel : Smartphones, tablettes

Exercice

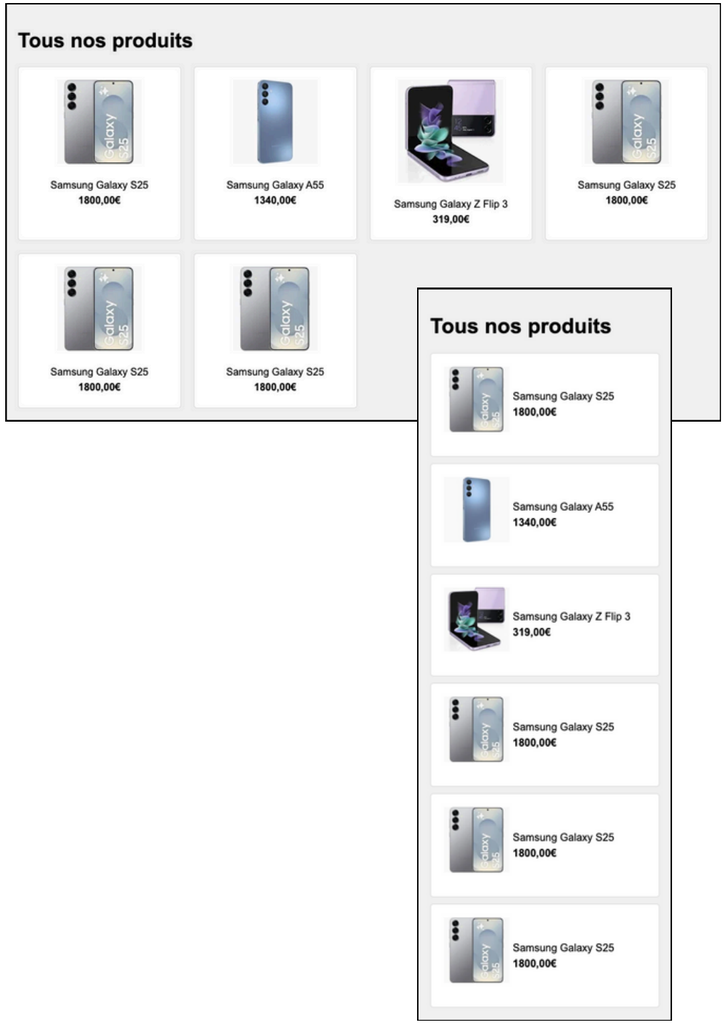
Consignes :

- Récupérer le fichier HTML avec toutes les variables et la structure.
- Reproduire en responsive design sur le comportement des captures 5_1 et 5_2.
- Créer une grille fluide en CSS avec flexbox & grid
- Utiliser toutes les variables présentes dans le fichier HTML.
- Utiliser le point d'arrêt "480px" @media (min-width: 480px) {}

https://www.afecdax.ovh/ressources/dev/css/exercices/exercice_5.html

https://www.afecdax.ovh/ressources/dev/css/exercices/exercice_5_1.png

https://www.afecdax.ovh/ressources/dev/css/exercices/exercice_5_2.png



Ressources

Demo Responsive
https://www.afecdax.ovh/ressources/dev/css/css_demo_responsive.html

Doc Media Queries
https://developer.mozilla.org/fr/docs/Web/CSS/CSS_media_queries/Using_media_queries

Containers Queries
https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_containment/Container_queries

Exercice
https://www.afecdax.ovh/ressources/dev/css/exercices/exercice_5.html
https://www.afecdax.ovh/ressources/dev/css/exercices/exercice_5_1.png
https://www.afecdax.ovh/ressources/dev/css/exercices/exercice_5_2.png