




Positionnement avancé

Note importante : Ce document est conçu comme un guide de référence, mais n'est pas exhaustif. La capacité à rechercher des informations complémentaires et à résoudre des problèmes de manière autonome est une compétence essentielle pour tout développeur. Si vous rencontrez des difficultés ou si certaines étapes ne fonctionnent pas comme prévu, n'hésitez pas à consulter d'autres ressources (documentation officielle, tutoriels YouTube, forums spécialisés comme Stack Overflow). Apprendre à trouver l'information manquante et à déboguer par vous-même est

Grid Layout

CSS Grid est un système de mise en page bidimensionnel qui permet de créer des designs complexes.

W3Schools.com

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects like HTML, CSS, JavaScript, Python, SQL,  https://www.w3schools.com/css/css_grid.asp



Conteneur Grid

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr); /* 3 colonnes de taille égale */  
  grid-template-rows: 100px auto 100px; /* 3 lignes */  
  gap: 10px; /* espace entre les cellules */
```

```

/* Autres propriétés utiles */
grid-template-areas:
  "header header header"
  "sidebar content content"
  "footer footer footer";
justify-items: start; /* start, end, center, stretch */
align-items: center; /* start, end, center, stretch */
justify-content: space-between; /* start, end, center, stretch, space-around, space-between, space-evenly */
align-content: space-evenly; /* start, end, center, stretch, space-around, space-between, space-evenly */
}

```

Éléments Grid

```

.grid-item {
  grid-column: 1 / 3; /* de la ligne 1 à la ligne 3 */
  grid-row: 2 / 4; /* de la ligne 2 à la ligne 4 */

  /* Alternative */
  grid-column-start: 1;
  grid-column-end: 3;
  grid-row-start: 2;
  grid-row-end: 4;

  /* Raccourci avancé */
  grid-area: header; /* correspond à une zone nommée */
  /* ou */
  grid-area: 2 / 1 / 4 / 3; /* row-start / column-start / row-end / column-end */

  justify-self: center; /* start, end, center, stretch */
  align-self: end; /* start, end, center, stretch */
}

```

Layout classique avec Grid


```
.page-layout {
  display: grid;
  grid-template-columns: 250px 1fr;
  grid-template-rows: auto 1fr auto;
  grid-template-areas:
    "header header"
    "sidebar content"
    "footer footer";
  min-height: 100vh;
}
```

```
.header { grid-area: header; }
.sidebar { grid-area: sidebar; }
.content { grid-area: content; }
.footer { grid-area: footer; }
```

Responsive avec les Media Queries

Les Media Queries permettent d'appliquer des styles CSS en fonction des caractéristiques de l'appareil.

W3Schools.com

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects like HTML, CSS, JavaScript, Python, SQL,  https://www.w3schools.com/css/css3_mediaqueries.asp



Types de média

- `all` - Tous les appareils
- `print` - Mode d'impression
- `screen` - Écrans d'ordinateur, tablettes, smartphones, etc.


Propriétés fréquemment utilisées

- `width` / `min-width` / `max-width` - Largeur de la fenêtre
- `height` / `min-height` / `max-height` - Hauteur de la fenêtre

- **orientation** - Portrait ou paysage
- **aspect-ratio** - Rapport largeur/hauteur
- **resolution** - Résolution de l'appareil

Exemples pratiques

W3Schools.com

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects like HTML, CSS, JavaScript, Python, SQL,  https://www.w3schools.com/css/css_rwd_mediaqueries.asp



Adaptation en fonction de la largeur:

```
/* Mobile first */
.container {
  width: 100%;
  padding: 15px;
}

/* Tablettes (768px et plus) */
@media screen and (min-width: 768px) {
  .container {
    width: 750px;
    margin: 0 auto;
  }
}

/* Ordinateurs portables (992px et plus) */
@media screen and (min-width: 992px) {
  .container {
    width: 970px;
  }
}

/* Grands écrans (1200px et plus) */
@media screen and (min-width: 1200px) {
  .container {
```

```
width: 1170px;
}
}
```

Navigation responsive:

```
.nav {
  display: flex;
  justify-content: space-between;
}

/* Sur petit écran */
@media screen and (max-width: 768px) {
  .nav {
    flex-direction: column;
  }

  .nav-links {
    flex-direction: column;
    display: none; /* à activer par JavaScript */
  }

  .nav-links.active {
    display: flex;
  }

  .menu-toggle {
    display: block;
  }
}
```

Adaptation selon l'orientation:

```
.gallery {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
}
```

```
@media screen and (orientation: portrait) {  
  .gallery {  
    grid-template-columns: 1fr;  
  }  
}
```

Bonnes pratiques pour le responsive design

1. **Mobile-first** : Commencer par les styles pour petits écrans, puis ajouter des media queries pour les écrans plus grands.
2. **Points de rupture relatifs au contenu** : Choisir les points de rupture en fonction de votre contenu, pas des appareils spécifiques.
3. **Unités relatives** : Préférer les unités comme `em`, `rem`, `%` et `vh` / `vw` plutôt que des pixels.
4. **Images flexibles** : Utiliser `max-width: 100%` pour les images.
5. **Tester sur plusieurs appareils** : Vérifier le rendu sur différents écrans et orientations.

Protection intellectuelle : Ce cours, est une création originale de **MANJAL Younes** et est protégé par le droit d'auteur conformément au Code de la propriété intellectuelle français (article L.111-1). Toute reproduction, modification, diffusion ou utilisation, même partielle, sans autorisation écrite préalable est strictement interdite. L'intelligence artificielle a été utilisée uniquement comme outil d'assistance pour la correction orthographique et la reformulation de certaines phrases, sous ma supervision et validation. Le contenu pédagogique et intellectuel reste entièrement ma création personnelle.