Séance 4 : CSS Grid Layout

La Mise en Page Bidimensionnelle

Fossé Stéphane

30 Mai 2025

IUT de Bordeaux

Aperçu de la Séance 4

Objectifs de la Séance

- Maîtriser la mise en page bidimensionnelle avec CSS Grid.
- Comprendre les propriétés du conteneur et des éléments de grille.
- Introduire le Responsive Design et les Media Queries.

1. Organiser Votre Contenu avec une Grille

1.1 Introduction à CSS Grid Layout

- Système de mise en page bidimensionnel (lignes **et** colonnes).
- Idéal pour la structure globale d'une page (header, sidebar, main content, footer).

1.2 Conteneur de Grille et Éléments de Grille

- `grid-container` (parent) et `grid-item` (enfants directs).

```
<div class="grid-container"> <!-- Conteneur de Grille -->
    <div class="grid-item">Header</div> <!-- Élément de Grille -->
    <div class="grid-item">Sidebar</div>
    <div class="grid-item">Main Content</div>
    <div class="grid-item">Footer</div>
</div>
```

```
.grid-container {
    display: grid; /* Transforme l'élément en conteneur de grille */
    background-color: #F0F8FF; /* Pour visualiser le conteneur */
    padding: 15px;
    border: 1px solid #A5B4FC;
}
.grid-item {
```

```
    background-color: #BFDBFE; /* Pour visualiser les éléments */
    padding: 10px;
    border: 1px solid #60A5FA;
    margin: 5px;
}
```

1.3 Définir la Grille (Propriétés du Conteneur de Grille)

- `grid-template-columns`: Définit le nombre et la largeur des colonnes (`fr`, `repeat()`, `minmax()`).
- `grid-template-rows`: Définit le nombre et la hauteur des lignes.
- `gap`, `row-gap`, `column-gap`: Espacement entre les cellules.
- `justify-items`, `align-items`: Aligne le contenu **dans** les cellules.
- `justify-content`, `align-content`: Aligne la **grille entière**.
- `grid-template-areas`: Nommer des zones pour un placement lisible.

1.4 Placer les Éléments (Propriétés des Éléments de Grille)

- `grid-column-start`, `grid-column-end`, `grid-column`: Positionnement horizontal.
- `grid-row-start`, `grid-row-end`, `grid-row`: Positionnement vertical.
- `grid-area`: Placer un élément dans une zone nommée.
- `justify-self`, `align-self`: Aligne l'élément **lui-même** dans sa cellule.

1.5 Introduction au Responsive Design et Media Queries (Rappel)

1.5.1 Concept de Mobile-First

- Concevoir et développer d'abord pour les petits écrans.

1.5.2 Syntaxe de Base des Media Queries

- `@media screen and (min-width: 768px)`.
- `min-width`, `max-width`.

```
/* Styles par défaut (pour mobile d'abord) */
body {
    background-color: lightcoral;
}

/* Styles pour les écrans d'une largeur minimale de 768px (tablettes et plus)
*/
@media screen and (min-width: 768px) {
    body {
        background-color: lightblue; /* Le fond devient bleu sur les écrans
plus grands */
    }
}
```

1.5.2 Syntaxe de Base des Media Queries

```
/* Styles pour les écrans d'une largeur maximale de 600px (petits mobiles) */
@media screen and (max-width: 600px) {
    h1 {
        font-size: 1.5em; /* La taille du titre diminue sur les petits écrans
*/
    }
}
```

Fin de la Séance 4

Récapitulatif

- CSS Grid pour les layouts complexes et bidimensionnels.
- Propriétés de conteneur et d'éléments de grille pour un contrôle précis.
- Responsive Design avec Media Queries pour l'adaptation aux différentes tailles d'écran.

Prochaines Étapes

- Flexbox pour la mise en page unidimensionnelle et la distribution d'espace.