

CSS: Display Flex & Grid

Introduzione

- **CSS Flexbox** e **CSS Grid** sono due potenti strumenti per la gestione del layout in CSS.
- **Flexbox** (Flexible Box) è ideale per allineamenti monodimensionali.
- **Grid** è perfetto per layout bidimensionali.

Flexbox: Concetti Base

- Modello di layout unidimensionale.
- Gestisce la disposizione degli elementi lungo un asse principale.
- Proprietà chiave:
 - `display: flex;`
 - `flex-direction`
 - `justify-content`
 - `align-items`
 - `flex-wrap`

Esempio Base di Flexbox

```
.container {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
  align-items: center;  
  height: 200px;  
  background: lightgray;  
}  
.item {  
  width: 50px;  
  height: 50px;  
  background: steelblue;  
  margin: 5px;  
}
```

```
<div class="container">  
  <div class="item"></div>  
  <div class="item"></div>  
  <div class="item"></div>  
</div>
```

Flexbox: Proprietà Principali

- `flex-direction` : determina la direzione dell'asse principale.
- `justify-content` : allinea gli elementi lungo l'asse principale.
- `align-items` : allinea gli elementi lungo l'asse trasversale.
- `flex-wrap` : specifica se gli elementi devono andare a capo.

CSS Grid: Concetti Base

- Modello di layout bidimensionale.
- Gestisce righe e colonne in modo simultaneo.
- Proprietà chiave:
 - `display: grid;`
 - `grid-template-rows` e `grid-template-columns`
 - `gap`
 - `grid-auto-flow`

Esempio Base di Grid

```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-gap: 10px;  
  background: lightgray;  
  padding: 10px;  
}  
.item {  
  background: steelblue;  
  padding: 20px;  
  text-align: center;  
  color: white;  
}
```

```
<div class="container">  
  <div class="item">1</div>  
  <div class="item">2</div>  
  <div class="item">3</div>  
  <div class="item">4</div>  
  <div class="item">5</div>  
  <div class="item">6</div>  
</div>
```

Grid: Proprietà Principali

- `grid-template-columns` : definisce il numero e la dimensione delle colonne.
- `grid-template-rows` : definisce il numero e la dimensione delle righe.
- `grid-gap` : definisce lo spazio tra gli elementi.
- `justify-items` e `align-items` : allineano gli elementi nelle celle.

Confronto tra Flexbox e Grid

Caratteristica	Flexbox	Grid
Layout	Monodimensionale	Bidimensionale
Direzione	Orizzontale/Verticale	Righe e colonne
Posizionamento	Più flessibile	Strutturato
Facilità d'uso	Intuitivo per layout semplici	Ideale per griglie complesse

Conclusione

- **Flexbox** è perfetto per layout lineari e allineamenti dinamici.
- **Grid** è la scelta migliore per strutture complesse.
- Spesso è utile combinare entrambi per ottenere il miglior risultato.

Esercizio: Prova a creare un layout combinato con Flexbox e Grid!