

Frontend Training - Day 3: CSS Combinators, Media Queries, Flexbox, and Grid

1. CSS Combinators

Combinators define relationships between selectors. Four main types:

1. Descendant selector (space): targets elements inside another element.
2. Child selector (>): targets only direct children.
3. Adjacent sibling selector (+): targets next immediate sibling.
4. General sibling selector (~): targets all following siblings.

Example:

```
.container p { color: red; }      /* descendant */  
.container > p { font-weight: bold; } /* child */  
p + span { color: green; }        /* adjacent sibling */  
p ~ p { font-style: italic; }     /* general sibling */
```

2. Media Queries

Media queries make websites responsive based on screen size.

Syntax:

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

Breakpoints:

- max-width: 576px - small devices
- max-width: 768px - medium devices
- min-width: 1200px - large devices

Example:

```
<style>
```

```
.box {  
  width: 100px;  
  height: 100px;  
  background: tomato;  
}
```

```
@media (max-width: 600px) {  
  .box { background: skyblue; }  
}
```

```
</style>
```

3. CSS Flexbox - In Depth

Used for one-dimensional layout (row/column).

Main container properties:

- display: flex;
- flex-direction: row | column | row-reverse | column-reverse
- justify-content: flex-start | center | flex-end | space-between | space-around | space-evenly

- align-items: stretch | flex-start | center | flex-end | baseline
- flex-wrap: wrap | nowrap
- gap: space between flex items

Child properties:

- flex-grow: defines how much an item will grow
- flex-shrink: defines how much an item will shrink
- flex-basis: default size before growing/shrinking
- flex: shorthand for grow, shrink, and basis

Example:

```
.flex-container {  
  display: flex;  
  justify-content: space-around;  
  align-items: center;  
}  
  
.box { flex: 1; }  
  
.box:nth-child(2) { flex: 2; }
```

4. CSS Grid

Two-dimensional layout system (rows and columns).

Key properties:

- display: grid;
- grid-template-columns: repeat(3, 1fr);

- grid-template-rows
- gap
- grid-column, grid-row

Example:

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  gap: 10px;  
}  
  
.grid-container div {  
  background: #add8e6;  
  padding: 20px;  
  text-align: center;  
}
```