



RESPONSIVE WITH CSS

RESPONSIVE DESIGN

Mobile-First Approach

Purpose:

Prioritizes mobile user experience, improves performance, and keeps the codebase cleaner.

Techniques:

- Start styling for mobile devices first, without media queries.
- Use @media (**min-width: value**) for larger screen styles.

EXAMPLE:

```
body {  
  font-size: 16px;  
  padding: 10px;  
}  
  
@media (min-width: 768px) {  
  body {  
    font-size: 18px;  
    padding: 20px;  
  }  
}
```



RESPONSIVE WITH CSS

RESPONSIVE DESIGN

@media Queries

Purpose:

Apply styles dynamically based on screen width or other device characteristics.

Techniques:

- @media (min-width: value): Styles apply for screens wider than the given value.
- @media (max-width: value): Styles apply for screens narrower than the given value.
- Define breakpoints as SCSS variables for better maintainability.

EXAMPLE:

```
$breakpoint-tablet: 768px;

.container {
  width: 100%;
}

@media (min-width: $breakpoint-tablet) {
  .container {
    width: 80%;
  }
}
```



RESPONSIVE WITH CSS

RESPONSIVE DESIGN

Grid

Purpose:

Provides two-dimensional layouts with rows and columns for complex designs.

Techniques:

- `display: grid`
- `grid-template-columns: repeat(3, 1fr)`
- `grid-gap: 10px`

EXAMPLE:

```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-gap: 10px;  
}
```