

1. Define Viewport Meta Tag:

In the <head> section of your HTML document, add a <meta> tag with the following attributes:

- name: Set this to "viewport"
- content: Set this to "width=device-width, initial-scale=1.0"

This tells the browser to set the width of the viewport to the device's width and the initial zoom level to 1.0.

2. Use Media Queries:

Media queries allow you to apply different styles to your content based on the screen size or resolution. You can define multiple media queries to target specific devices or screen sizes. Here is an example:

```
/* Styles for small screens (less than 768px) */
```

```
@media only screen and (max-width: 768px) {
```

```
  body {
```

```
    font-size: 16px;
```

```
  }
```

```
  .content {
```

```
    width: 100%;
```

```
  }
```

```
}
```

```
/* Styles for medium screens (768px to 992px) */
```

```
@media only screen and (min-width: 768px) and (max-width: 992px) {
```

```
  body {
```

```
    font-size: 18px;
```

```
  }
```

```
  .content {
```

```
    width: 70%;
```

```
  }
```

```
}
```

```
/* Styles for large screens (greater than 992px) */  
@media only screen and (min-width: 992px) {  
  body {  
    font-size: 20px;  
  }  
  .content {  
    width: 50%;  
  }  
}
```

3. Use Flexible Units:

Instead of using fixed pixel values for sizes, use flexible units like percentages, vw, and vh. This allows the content to scale proportionally with the screen size.

4. Responsive Images:

Use the <picture> element and the srcset attribute to provide different image versions for different screen resolutions. This ensures that images are displayed at the appropriate size and quality on all devices.

5. Responsive Navigation:

For larger screens, a horizontal navigation bar might be suitable. However, for smaller screens, a hamburger menu that collapses content might be more user-friendly.