CHEAT SHEETS HERO

Responsive Web Design Cheatsheet

A quick reference guide to building responsive websites, covering viewport settings, media queries, flexible layouts, and responsive images.

Viewport & Basic Setup

Setting the Viewport

The viewport meta tag controls how a webpage scales on different devices. It's essential for responsive design.

```html
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- `width=device-width` : Sets the width of the viewport to the width of the device.
- `initial-scale=1.0` : Sets the initial zoom level when the page is first loaded.

Example of disabling zooming:

```html
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no">
```

Basic HTML Structure

A basic HTML5 structure to start with:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Responsive Webpage</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <!-- Header Content -->
    </header>
    <main>
        <!-- Main Content -->
    </main>
    <footer>
        <!-- Footer Content -->
    </footer>
</body>
</html>
```

Media Queries

Media Query Syntax

Media queries allow you to apply different styles based on the characteristics of the device, such as screen size, orientation, and resolution.

```css
@media (condition) {
  /* CSS rules */
}
```

Common Media Features:

- `max-width` : Applies styles when the screen width is less than or equal to a specified value.
- `min-width` : Applies styles when the screen width is greater than or equal to a specified value.
- `orientation` : Applies styles based on the device orientation (portrait or landscape).

Common Breakpoints

Common breakpoints are screen sizes at which your layout changes to adapt to different devices. These are just suggestions; choose breakpoints that work best for your content.

- Extra small devices (phones, less than 576px)
- Small devices (portrait tablets, 576px and up)
- Medium devices (landscape tablets, 768px and up)
- Large devices (laptops/desktops, 992px and up)
- Extra large devices (large laptops and desktops, 1200px and up)

Example Media Queries

```css
/* For mobile devices */
@media (max-width: 575.98px) {
  body {
    font-size: 14px;
  }
}

/* For tablets */
@media (min-width: 576px) and (max-width: 767.98px) {
  body {
    font-size: 16px;
  }
}

/* For desktops */
@media (min-width: 768px) {
  body {
    font-size: 18px;
  }
}
```

Flexible Layouts

CSS Flexible Box Layout (Flexbox)

Flexbox is a powerful layout module that makes it easier to design flexible and responsive layouts.

- `display: flex;` - Creates a flex container.
- `flex-direction` : Specifies the direction of the flex items (row or column).
- `justify-content` : Aligns flex items along the main axis.
- `align-items` : Aligns flex items along the cross axis.

Example:

```css
.container {
  display: flex;
  flex-direction: row;
  justify-content: center;
  align-items: center;
}
```

CSS Grid Layout

CSS Grid Layout is a two-dimensional layout system that allows you to create complex layouts with rows and columns.

- `display: grid;` - Creates a grid container.
- `grid-template-columns` : Defines the columns of the grid.
- `grid-template-rows` : Defines the rows of the grid.
- `grid-gap` : Specifies the gap between grid items.

Example:

```css
.container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-gap: 20px;
}
```

Relative Units

Using relative units like percentages (`%`), ems (`em`), rems (`rem`), and viewport units (`vw` , `vh`) allows elements to scale proportionally with the screen size.

- `%` : Relative to the parent element.
- `em` : Relative to the font size of the element.
- `rem` : Relative to the root element's font size.
- `vw` : Relative to 1% of the viewport's width.
- `vh` : Relative to 1% of the viewport's height.

Responsive Images

The `srcset` Attribute

The `srcset` attribute in the `<img>` tag allows you to specify different image sources for different screen sizes.

```html
<img src="image-small.jpg" srcset="image-small.jpg 480w, image-medium.jpg 800w, image-large.jpg 1200w" alt="Responsive Image">
```

- The `w` descriptor specifies the width of the image source.

The `<picture>` Element

The `<picture>` element provides more control over responsive images, allowing you to specify different images based on media queries.

```html
<picture>
  <source media="(max-width: 480px)" srcset="image-small.jpg">
  <source media="(max-width: 800px)" srcset="image-medium.jpg">
  <img src="image-large.jpg" alt="Responsive Image">
</picture>
```

CSS `object-fit` Property

The `object-fit` property specifies how the content of a replaced element (like `<img>` or `<video>`) should be resized to fit its container.

- `object-fit: cover;` - The image fills the entire container, potentially cropping the image.
- `object-fit: contain;` - The image is scaled to fit within the container, preserving its aspect ratio.
- `object-fit: fill;` - The image stretches to fill the container, potentially distorting the image.