**Responsive Web Design Overview**

Responsive web design is about creating websites that look good on all devices, from desktop computers to tablets and smartphones. Here's a hands-on exercise to get you started with the basics of responsive web design using HTML and CSS.

**HTML Structure**

1. **You can always use this basic structure for HTML:**

HTML

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Responsive Web Design Exercise</title>

<style></style>

</head>

<body>

</body>

</html>

1. **Create file as above, use semantic tags inside, and call it index.html. We can use this site for random images of any size: https://picsum.photos/**

HTML

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Responsive Web Design Exercise</title>

<style></style>

</head>

<body>

**<h1>Welcome to My Responsive Website</h1>**

**<p>This is a sample website that adapts to different screen sizes.</p>**

**<img src="https://picsum.photos/seed/picsum/200/100" alt="Responsive Image">**

**<section>**

**<h2>Our Services</h2>**

**<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>**

**</section>**

**<footer>**

**<p>&copy; 2025 Responsive Web Design</p>**

**</footer>**

</body>

</html>

**Now for the CSS for Responsive Design.**

1. **Note the viewport:**

The viewport meta tag is essential for responsive web design. It instructs the browser on how to control the page's dimensions and scaling.

meta name="viewport" content="width=device-width, initial-scale=1.0”

1. **Fluid Images:**

Create a file called styles.css. Use the width property with a percentage value to make images responsive. This allows the image to scale up or down depending on the screen size.

CSS

img {

width: 100%;

}

1. **Make Content Responsive:**

Use the vw unit for font size to make text responsive. 1vw equals 1% of the viewport width. This ensures the text size adjusts based on the screen size. The relative font sizes. rem, em, and vw are all relative units for specifying size in CSS, but they differ in their reference points:

* **em:**
  + **Reference:** The font size of the *parent* element.
  + **Example:** If the parent element has a font size of 16px, then 2em would be 32px.
  + **Use Cases:** Useful for scaling elements within a specific context, like headings within a particular section.
* **rem:**
  + **Reference:** The font size of the *root* element (usually the <html> element).
  + **Example:** If the root font size is 16px, then 2rem would always be 32px, regardless of the parent element's font size.
  + **Use Cases:** Provides more consistent scaling across the entire document. Ideal for global font sizes and spacing.
* **vw:**
  + **Reference:** 1% of the *viewport width*.
  + **Example:** 1vw is equal to 1% of the width of the user's browser window.
  + **Use Cases:** Useful for creating layouts that scale responsively based on the screen size. For example, a full-width element would be 100vw.

**Benefits for each relative measurement:**

* **Consistency:** rem generally offers the most consistent scaling behavior across your entire website.
* **Specificity:** em is more specific to the parent element's font size, which can be useful for creating nested effects.
* **Responsiveness:** vw is ideal for elements that should scale directly with the viewport width, such as background images or full-width layouts.

CSS

h1 {

font-size: 3vw;

}

p {

font-size: 1.5vw;

}

1. **Add Media Queries:**

Media queries allow you to apply different styles for various screen sizes. Here's an example that changes the layout from a multi-column to a single-column layout on smaller screens.

CSS

/\* For screens smaller than 768px \*/

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

section, footer {

width: 100%;

}

}

**Wrapping up this exercise:**

1. Save the two files and open index.html in a web browser. Resize the browser window to see how the website responds to different screen sizes.
2. Experiment with different media query screen sizes to create a more complex responsive layout.
3. Add a CSS framework like Bootstrap that provide pre-built responsive components and utilities. You can use this CDN: https://www.bootstrapcdn.com/