**\*Responsive website using HTML and CSS\***

--> Understanding Units

- px

- %

- vw , vh

- vmax , vmin

- em , rem

--> Layout of website

- absolute vs flex?

--> Flexbox

- Display flex

- aligning items in X and Y axis

- flex direction

- flex wrap

--> CSS Media Queries

- min height, min width

- min width, max width

**key points to keep in mind to make websites responsive.**

1) CSS Flexbox

2) CSS Units

3) Responsive Typography

4) Mobile-First Approach

5) Flexible Images and Media

**1. Pixels (px)**

**Definition:** Pixels are the most common unit in CSS. One pixel is one dot on the computer screen.

**Usage:** Use pixels when you need precise and fixed sizes for elements, such as borders, margins, padding, and font sizes. Pixels are resolution dependent.

**When to Use:**

* When you need exact dimensions.
* For non-responsive designs or elements that should not scale.

**2. Percentage (%)**

**Definition:** Percentages are relative to the parent element's dimensions. They provide a way to create fluid layouts that adapt to different screen sizes.

**Usage:** Percentages are commonly used for widths, heights, margins, and padding, allowing elements to resize based on the parent element.

**When to Use:**

* For creating responsive layouts.
* When you want an element to scale relative to its parent.

**3. Viewport Width (vw) and Viewport Height (vh)**

**Definition:** Viewport units are relative to the size of the browser’s viewport. 1vw is 1% of the viewport's width, and 1vh is 1% of the viewport's height.

**Usage:** They are ideal for responsive design, ensuring that elements adapt to the size of the viewport.

**When to Use:**

* For full-screen sections.
* When you need elements to scale based on the viewport size.
* For creating responsive typography that adapts to screen size.

**4. Viewport Min (vmin) and Viewport Max (vmax)**

**Definition:** Viewport min and max units are relative to the smaller (vmin) or larger (vmax) dimension of the viewport. 1vmin is 1% of the smaller dimension, and 1vmax is 1% of the larger dimension.

**Usage:** These units are useful for ensuring consistent scaling based on the more restrictive dimensions of the viewport.

**When to Use:**

* When you want to maintain aspect ratios that adapt to either width or height.
* For elements that need to be consistently sized relative to the viewport.

**5. Em (em) and Rem (rem)**

**Em:** **Definition:** An em is relative to the font-size of the parent element. If the font-size of the parent is 16px, 1em equals 16px.

**Usage:** Use ems for scalable elements. They are especially useful within nested elements where you want relative sizing.

**Rem:** **Definition:** A rem is relative to the font-size of the root element (**<html>**). It provides a consistent measurement across the document.

**Usage:** Rems are useful for global scalability, maintaining consistent sizing regardless of nesting.

**When to Use Em:**

* For components where relative scaling based on the parent element is needed.
* When designing modular elements that should adapt to their container's font-size.

**When to Use Rem:**

* For setting consistent, root-relative sizes.
* When you want to ensure uniform scaling across different parts of the document.
* Ideal for defining global typography and spacing.

**Summary**

* **px:** Fixed, precise sizes. Use for exact control.
* **%:** Relative to parent. Use for fluid, responsive layouts.
* **vw, vh:** Relative to viewport. Use for full-screen sections and viewport-based scaling.
* **vmin, vmax:** Relative to smaller/larger viewport dimension. Use for adaptive scaling based on the viewport's restrictive dimension.
* **em:** Relative to parent’s font size. Use for nested, scalable components.
* **rem:** Relative to root font size. Use for consistent, document-wide scaling.

**Flexbox: Key Concepts**

**1. display: flex;**

**Definition:** Setting **display: flex;** on a container enables Flexbox for its children.

**2. Aligning Items on X and Y Axis**

**Align Items on X Axis (justify-content):**

* **flex-start**: Align items to the start of the container.
* **flex-end**: Align items to the end of the container.
* **center**: Center items in the container.
* **space-between**: Distribute items with space between them.
* **space-around**: Distribute items with space around them.
* **space-evenly**: Distribute items with equal space around them.

**Align Items on Y Axis (align-items):**

* **flex-start**: Align items to the start of the container's cross axis.
* **flex-end**: Align items to the end of the container's cross axis.
* **center**: Center items along the container's cross axis.
* **baseline**: Align items along their baselines.
* **stretch**: Stretch items to fill the container's cross axis (default).

**3. Flex Direction**

**Definition:** The **flex-direction** property defines the main axis along which the flex items are laid out.

* **row**: Items are placed in a row (left to right, default).
* **row-reverse**: Items are placed in a row (right to left).
* **column**: Items are placed in a column (top to bottom).
* **column-reverse**: Items are placed in a column (bottom to top).

**4. Flex Wrap**

**Definition:** The **flex-wrap** property controls whether the flex container is single-line or multi-line, and the direction of the cross axis.

* **nowrap**: All flex items will be on one line (default).
* **wrap**: Flex items will wrap onto multiple lines, from top to bottom.
* **wrap-reverse**: Flex items will wrap onto multiple lines, from bottom to top.

**Summary**

* **Absolute Positioning:** Use for precise, fixed placements. Not ideal for responsive layouts.
* **Flexbox:** Use for flexible, responsive layouts. Adapts to screen sizes and container dimensions.

**Key Flexbox Properties:**

* **display: flex;**: Enables Flexbox.
* **Aligning Items:**
  + **justify-content**: Aligns items along the main axis (X axis).
  + **align-items**: Aligns items along the cross axis (Y axis).
* **flex-direction**: Defines the direction of the main axis.
* **flex-wrap**: Controls whether items wrap onto multiple lines.