**CSS Units & Responsive Design**

**1. CSS Units**

CSS uses different **units to define sizes, spacing, fonts, widths, heights, etc.**. Choosing the right unit is key for **responsive layouts**.

**A. Absolute Units**

| **Unit** | **Description** | **Usage** |
| --- | --- | --- |
| px | Pixels; fixed unit | Borders, small text, icons |
| pt | Points (1pt ≈ 1.333px) | Printing, rarely web |
| cm, mm, in | Physical units | Rarely used on screens |

**Example:**

div {

width: 300px;

height: 150px;

border: 2px solid black;

}

* **Pixels** = fixed → does not scale on different screen sizes.

**B. Relative Units**

| **Unit** | **Description** | **Usage** |
| --- | --- | --- |
| % | Percentage relative to **parent element** | Width, height, padding, margin |
| em | Relative to **font-size of parent** | Fonts, spacing |
| rem | Relative to **font-size of root (html)** | Fonts, spacing, layout |
| vh | 1% of viewport **height** | Full-screen sections |
| vw | 1% of viewport **width** | Full-width elements, typography |

**Examples:**

/\* Percentage relative to parent \*/

div {

width: 50%; /\* half of parent width \*/

padding: 10%; /\* 10% of parent width \*/

}

/\* em relative to parent font-size \*/

p {

font-size: 1.5em; /\* 1.5 times parent font size \*/

}

/\* rem relative to root font-size \*/

h1 {

font-size: 2rem; /\* 2 times root font size (default 16px -> 32px) \*/

}

/\* Viewport units \*/

section {

height: 100vh; /\* full viewport height \*/

width: 100vw; /\* full viewport width \*/

}

**Tips:**

* Use **px** for fixed precision, **em/rem** for scalable fonts.
* Use **%** for container-based sizing.
* Use **vh/vw** for **full-screen layouts and responsive typography**.

**2. Responsive Web Design Basics**

**Goal:** Make a website look **good on all screen sizes** (desktop, tablet, mobile).

**Techniques:**

1. **Fluid layouts** → % / em / rem units
2. **Flexible images** → max-width: 100%
3. **Media Queries** → apply styles for specific screen widths

**A. Media Queries**

* Apply CSS rules **only for specific screen sizes**.
* Syntax:

@media (condition) {

/\* CSS rules \*/

}

**Examples:**

1. **Mobile-first approach (max-width):**

/\* Default styles for desktop \*/

body {

font-size: 16px;

}

/\* Styles for devices ≤ 600px \*/

@media (max-width: 600px) {

body {

font-size: 14px;

}

}

1. **Tablet-specific layout**

@media (min-width: 601px) and (max-width: 1024px) {

.container {

width: 80%;

}

}

1. **Landscape vs Portrait**

@media (orientation: portrait) {

body {

background-color: lightblue;

}

}

**B. Responsive Images**

img {

max-width: 100%;

height: auto;

}

* Keeps image inside its container without distortion.

**C. Example: Responsive Layout**

<div class="container">

<p>Responsive Box</p>

</div>

.container {

width: 70%; /\* fluid width \*/

padding: 20px;

background: lightgreen;

}

@media (max-width: 768px) {

.container {

width: 90%; /\* expand on smaller screens \*/

padding: 10px;

}

}

@media (max-width: 480px) {

.container {

width: 100%; /\* full width on mobile \*/

padding: 5px;

}

}

**Explanation:**

* Container adjusts width based on screen size.
* Padding also adjusts for better spacing on smaller devices.

**3. Quick Tips / Interview Points**

1. **px** → fixed size, not scalable
2. **em** → relative to **parent font-size**, good for nested elements
3. **rem** → relative to **root font-size**, easier for global scaling
4. **%** → relative to parent → fluid layout
5. **vh/vw** → relative to viewport → full-screen sections
6. **Media Queries** → essential for responsive design
7. **Mobile-first approach** → write default styles for mobile, then larger screens
8. Use **flexbox/grid + relative units + media queries** → fully responsive layout

**4. Common Interview Questions**

1. Difference between **px, em, rem, %, vh, vw**?
2. When to use **em vs rem**?
3. Difference between **max-width and min-width in media queries**?
4. How to make an image **responsive**?
5. What is **mobile-first vs desktop-first** approach?
6. How do **viewport units** differ from percentage?