**CSS Grid System and Breakpoints**

**CSS Grid Layout** is a powerful layout system that allows you to arrange items into a grid structure. It provides more flexibility and control compared to traditional layout methods like floats or inline-block elements.

**Grid System Implementation:**

1. **Set the Display Property:** Apply display: grid; to the container element to enable grid layout.
2. **Define Grid Tracks:** Use the grid-template-columns and grid-template-rows properties to specify the number and size of grid tracks.
3. **Place Items:** Use the grid-column-start, grid-column-end, grid-row-start, and grid-row-end properties to position items within the grid.

**Responsive Design with Breakpoints:**

* **Media Queries:** Use media queries to define different styles for different screen sizes.
* **Breakpoints:** Specify the screen widths at which the layout should change.

**Example:**

HTML

<div class="container">

<div class="item">Item 1</div>

<div class="item">Item 2</div>

<div class="item">Item 3</div>

</div>

Use code [with caution.](/faq#coding)

CSS

.container {

display:

grid;

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

gap: 20px;

}

.item {

background-color: lightblue;

padding: 20px;

text-align: center;

}

@media (max-width: 768px) {

.container {

grid-template-columns: 1fr;

}

}

Use code [with caution.](/faq#coding)

**Explanation:**

* The .container element is set to use grid layout.
* The grid-template-columns property creates a grid with auto-fit columns, each with a minimum width of 200px and a flexible size (1fr).
* The gap property sets the spacing between grid items.
* The media query adjusts the grid layout for screens smaller than 768px, creating a single-column layout.

**By using CSS Grid and media queries, you can create responsive layouts that adapt to different screen sizes and devices.**

Sources and related content