Grids & Responsiveness

Instructor - Love Babbar

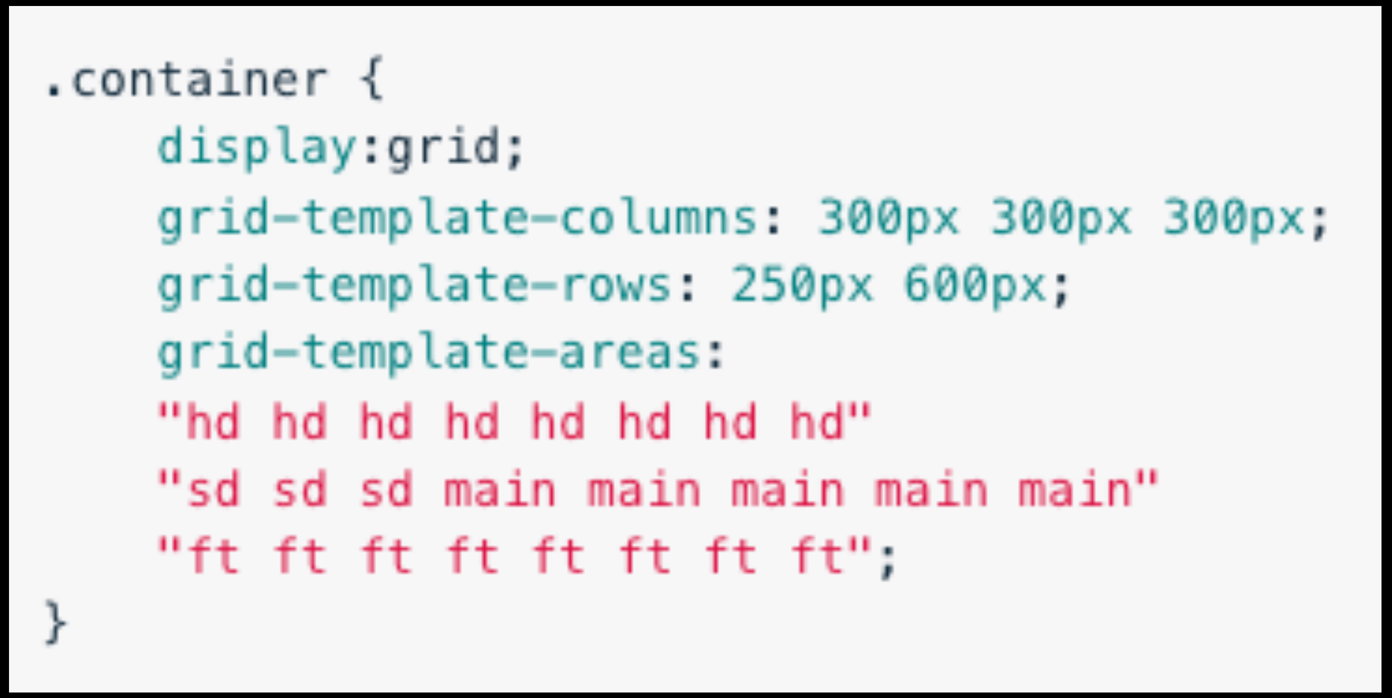
Grid-area

The grid-area property specifies a particular area or set of rows and columns that a grid item occupies. It is applied to the grid item itself with CSS. Here is an example:

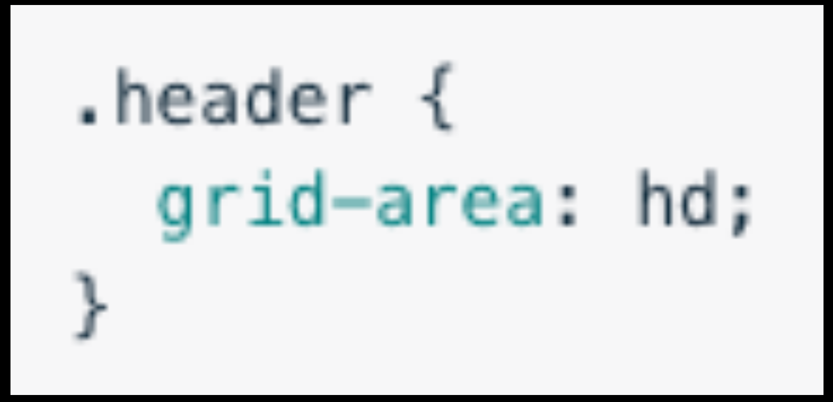
- .item { grid-area: 1/2/3/3 }
- Because grid-area is shorthand for the properties: grid-row-start, grid-column-start, grid-row-end and grid-column-end, the code above places the item from rows 1-3, and columns 2-3.

Grid-template-area:

grid-template-areas is the property used to name the rows and columns of a grid and to set its layout. It could look like this:



```
.container {
    display:grid;
    grid-template-columns: 300px 300px 300px;
    grid-template-rows: 250px 600px;
    grid-template-areas:
    "hd hd hd hd hd hd hd hd"
    "sd sd sd main main main main main"
    "ft ft ft ft ft ft ft ft";
}
```



```
.header {
  grid-area: hd;
}
```

Advanced Grid Concepts:

- Fr unit
- Repeat function
- Grid-auto-rows: minmax()

Grid Properties:

- Justify-content
- Align-content
- Justify-items
- Align-items
- Justify-self
- Align-self
- Place-items
- Place-self

Explore time:

- Find out the difference between grid and inline-grid.

Blog Website Layout

Link multiple pages together [CSS Grid allows you to create simple, elegant and professional quality webpage designs.]

Explore time:

Learn more

- Complex Grid Layouts
- Nesting & Overlapping of Grid Items

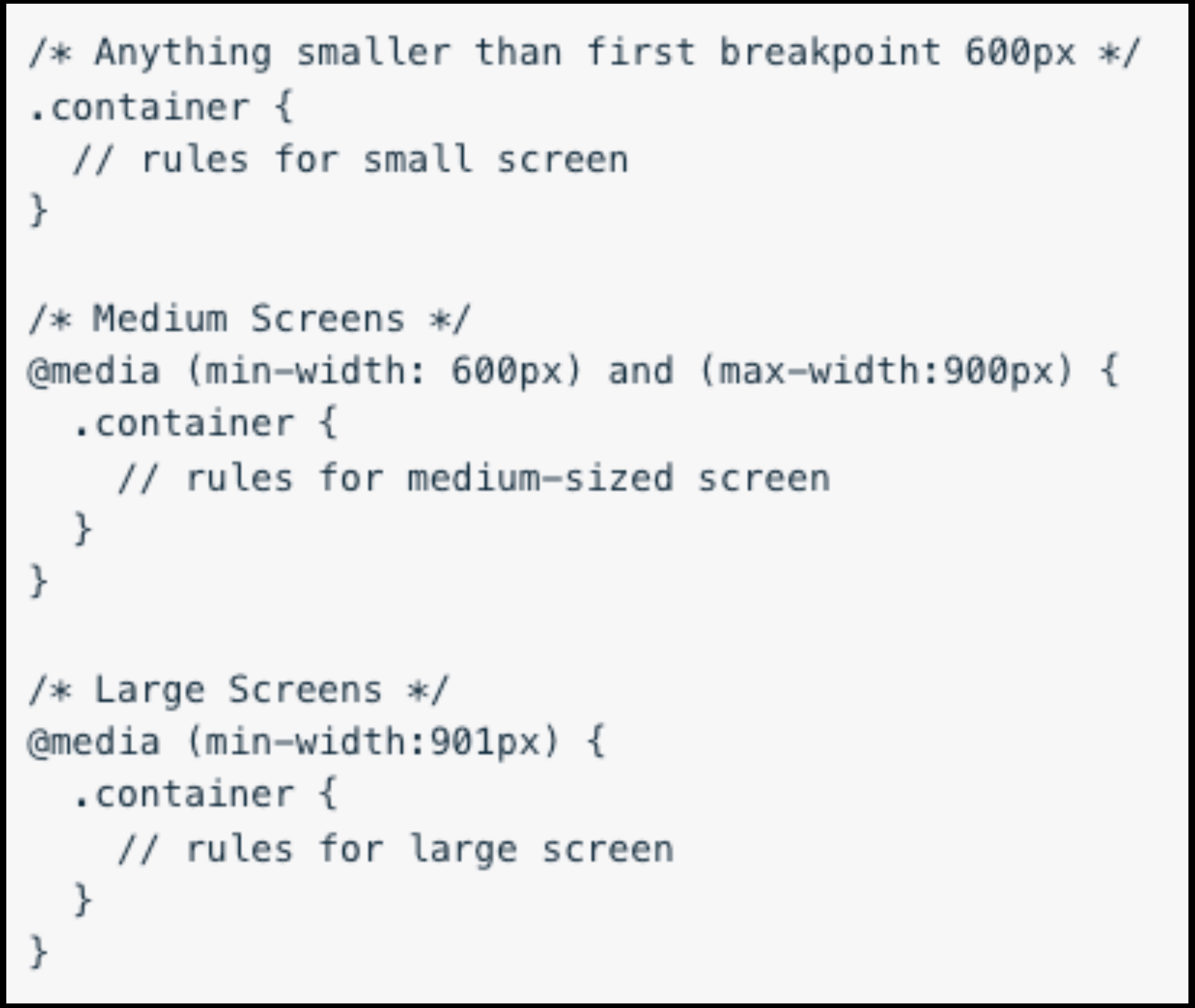
Media Queries:

- ‘Viewport’ - the area of the window in which web content can be seen. We use the dimensions of the viewport (usually the width, but sometimes the height) as the basis of our media queries.

- Media queries are used to set different style rules for different devices or sized screens. We use breakpoints to set the condition of a media query. The logic is: @media(feature:value)

- Essentially, media query breakpoints are pixel values that a developer/designer can define in CSS. When a responsive website reaches those pixel values, a transformation (such as the one detailed above) occurs so that the website offers an optimal user experience

Multiple BreakPoints:



```
/* Anything smaller than first breakpoint 600px */
.container {
  // rules for small screen
}

/* Medium Screens */
@media (min-width: 600px) and (max-width:900px) {
  .container {
    // rules for medium-sized screen
  }
}

/* Large Screens */
@media (min-width:901px) {
  .container {
    // rules for large screen
  }
}
```

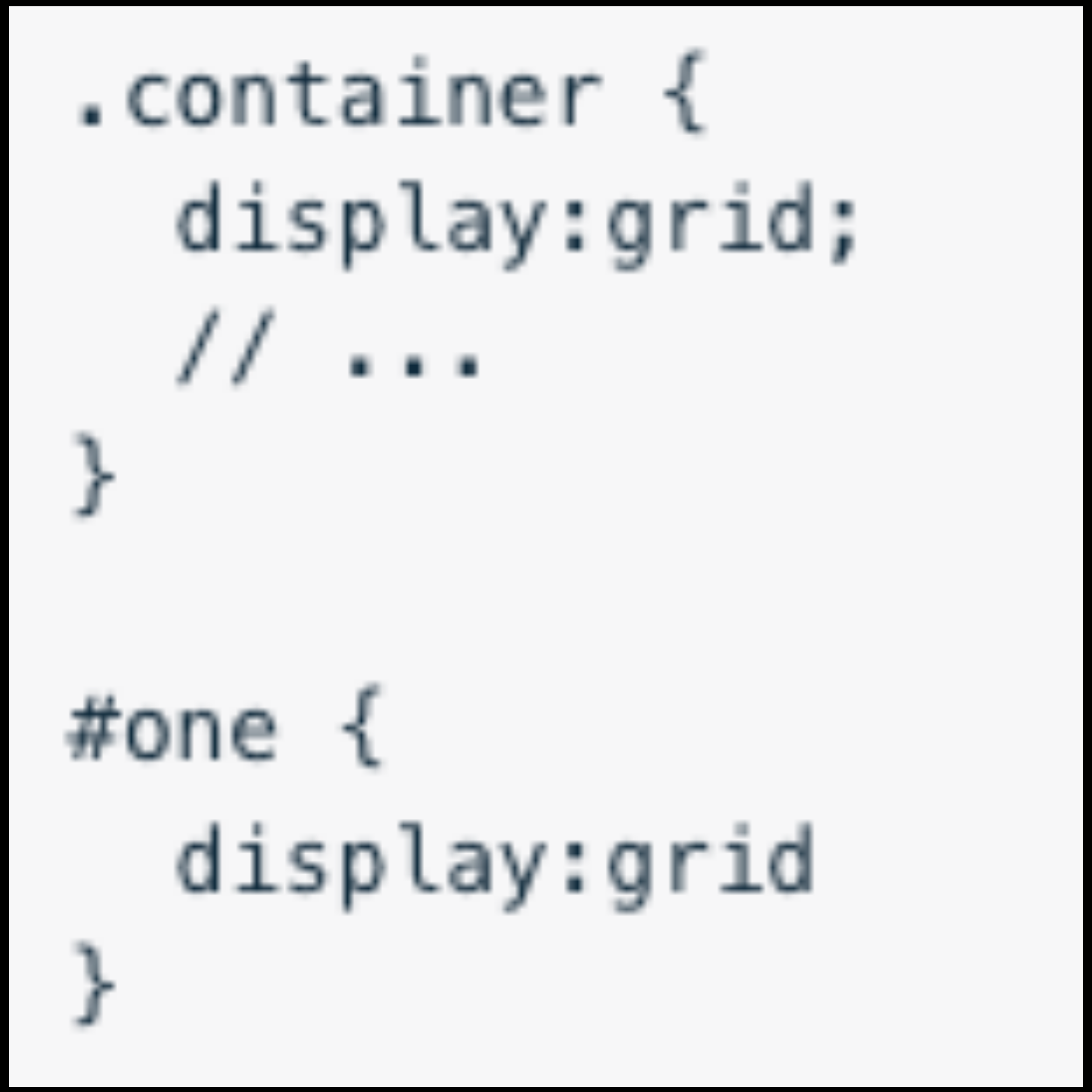
Explore time:

Create your own layout using Flex and Grids together

Nested Grids

Nesting CSS grids is simple and can be done simply by using the display:grid rule for both a parent and child element.

Here is how that could look with real code:



```
.container {
  display:grid;
  // ...
}

#one {
  display:grid
}
```