# Web Development Club: Week 07

Introduction to responsive design: CSS Grids

# Last week recap:

Last week we had a work week and discussed moving forward with our lessons more in depth:

- Previously, we've covered how to use github pages, syntax, HTML element declarations (defining headers, divisions, text, images, etc), CSS properties including margins and padding, borders, background, typography, positioning
- Based on your feedback, we'll be dedicating alternating club weeks to new lessons and work/review
- Going forward, we'll be covering more complex uses of HTML/CSS and potentially javascript if we have time
  - Future topics might include selector syntax, flexboxes, display, filters and animations, lists, tables, etc.

# Table of contents

01

## Lesson 6 Recap

Brief recap of our previous lessons and ideas moving forward

02

## Intro to CSS Grids

We'll introduce the idea of responsive design and how CSS Grid systems work

03

## Examples

Examples of a grid layout using previous concepts

04

## Questions & Practice

You can describe the topic of the section here

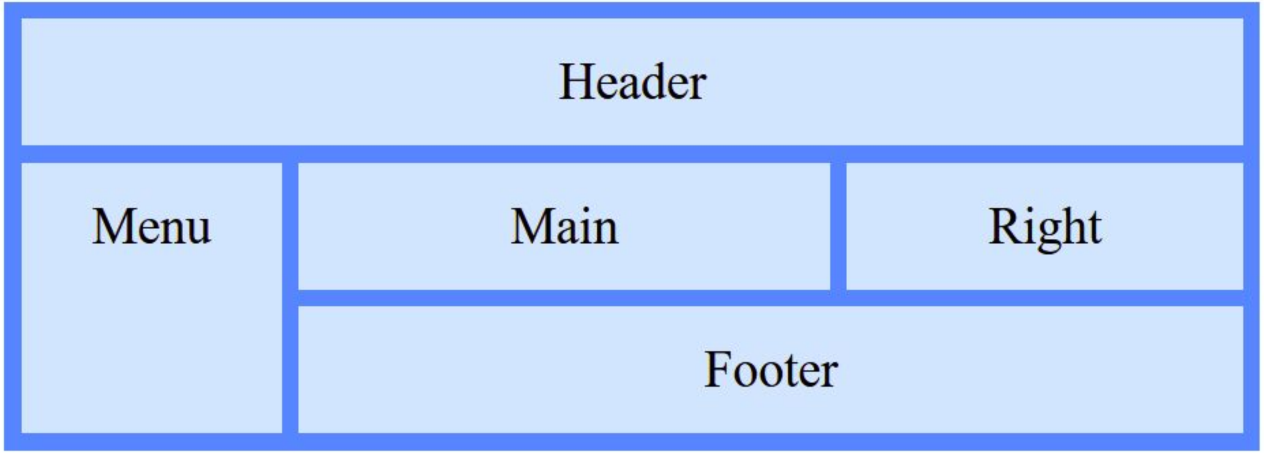
# What is responsive design?

Responsive design is a modern approach to web development that allows parts of a webpage to display by automatically adapting to different screen or window sizes

- Most basic web design only accounts for the window size that you develop it on, but responsive design allows your page to adapt to many different screen sizes
- This can be achieved through CSS grids, flexbox display, or viewport to mobile screen sizes

# Intro to CSS Grids:

CSS Grid module display allows for a layout system with rows and columns without having to use positioning or floats



Header

Menu

Main

Right

Footer

# HTML Grid layout definition

Grid Container:

- A grid container has one grid item for each defined column and row, but you can style them grid items so that they span multiple columns and/or rows.
- To define a grid layout in HTML, you first use <div> and class:
- <div class="gridcontainer">

  ```
          <div class="item1">Title</div>
          <div class="item2">Sidebar</div>
          <div class="item3">Main Content</div>
          <div class="item4">Media Right</div>
          <div class="item5">Footer</div>
   </div>
  ```

# CSS Grids Container

To style your grid container (main page layout, the number of sections):

- <style>

```
.gridcontainer {
        display: grid;
        max-width: 80%;
        margin: auto;
        background-color: darkgray;
        padding: 10px;
        gap: 10px;
}
```

You can also specify the default sizes of columns or rows with “grid-template-columns:” in the gridcontainer CSS

# CSS Grid Items

To style your grid items you can change the amount of space they take up and the style of individual sections:

-
  ```
  .item1 {
      background-color: lightgray;
      text-align: center;
      font-size: 30px;
      padding: 20px;
      grid-column: 1 / span 4;
      grid-row: 1 / span 1;
  }
  ```

```
.item2 {
    background-color: lightblue;
    padding: 10px;
    text-align: center;
    font-size: 15px;
    grid-column: 1 / span 1;
    grid-row: 2 / span 3;
}
```

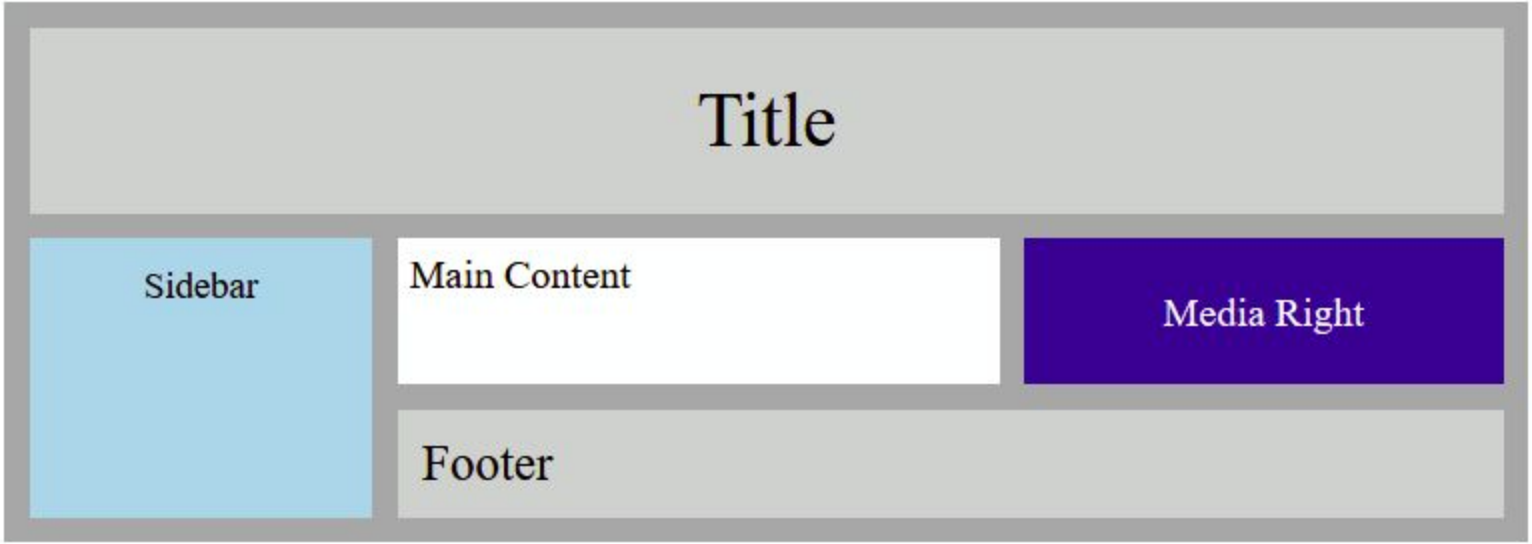
# CSS Grid Items

-
  ```
  .item3 {
          background-color: white;
          padding: 5px;
          grid-column: 2 / span 2;
          grid-row: 2 / span 2;
  }
  ```

-
  ```
  .item4 {
          background-color: darkblue;
          color: white;
          padding: 20px;
          text-align: center;
          grid-column: 4 / span 1;
          grid-row: 2 / span 2;
  }
  ```

```
.item5 {
        background-color: lightgray;
        padding: 10px;
        font-size: 20px;
        grid-column: 2 / span 3;
        grid-row: 4 / span 1;
    }
```

# CSS Grid Example



Title

Sidebar

Main Content

Media Right

Footer

# Practice

We'll use the remaining class time for independent practice, please feel free to ask questions as always!