

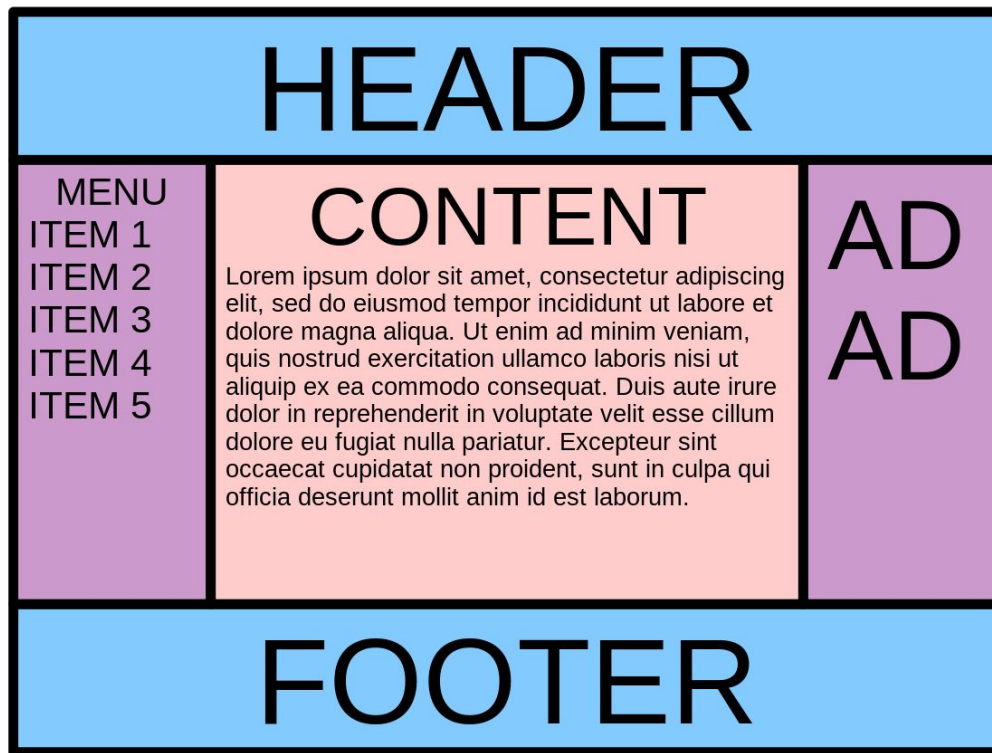
Module 3 Day 3

CSS: Responsive Design 1 - Grids

Today's Objectives

- Responsive Design
 - Holy Grail Layout
 - Basics: Desktop v. Web
- Grid
 - `grid-template-columns`
 - `grid-template-areas`
 - `grid-area`
 - `grid-gap`

The Holy Grail Layout

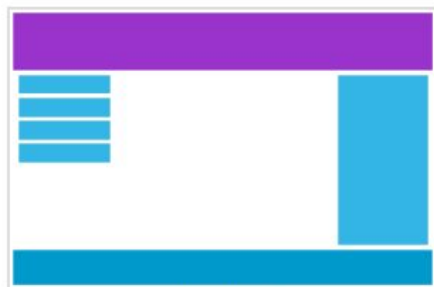


Responsive Design

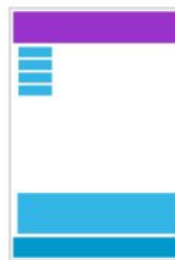
Designing For The Best Experience For All Users

Web pages can be viewed using many different devices: desktops, tablets, and phones. Your web page should look good, and be easy to use, regardless of the device.

Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device:



Desktop



Tablet



Phone

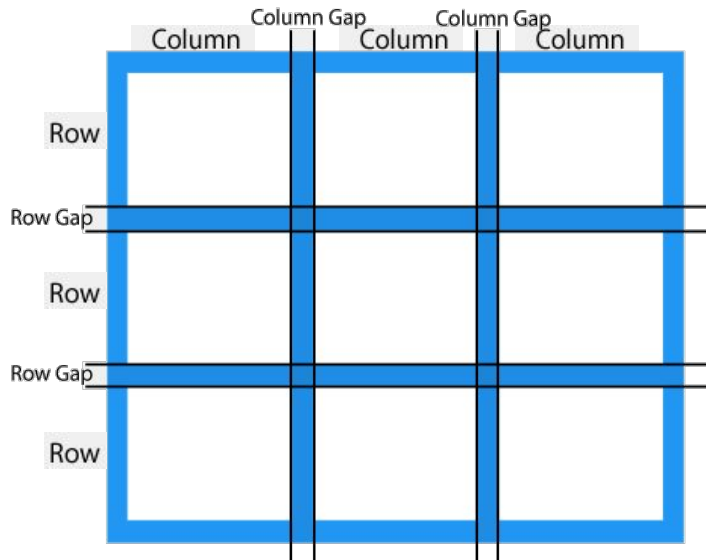
It is called responsive web design when you use CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.

CSS Grid Layout

[A Complete Guide to Grid](#)

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

- Grid is a 2-dimensional layout system for organizing data on web pages.
- It allows us to define relationships between HTML elements in terms of size, position, and layer.
- Vertical lines of grid items are **columns**.
- Horizontal lines of grid items are **rows**.
- The spaces between each column/row are called gaps.



CSS Grid: grid-template-columns

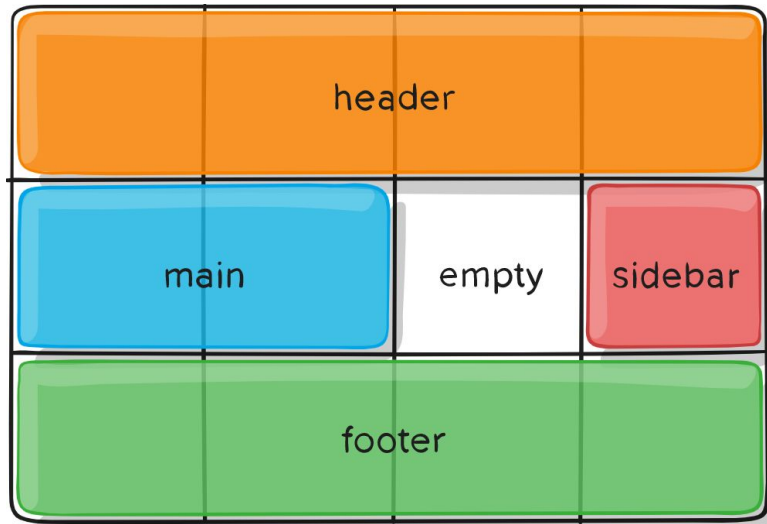
- Grid template columns allow us to proportionally allocate the size of our columns.
- They can be set using fr (fractional units), percentages, or auto. We will use fr, which is preferred.

```
/* Defines four columns for our grid, the first as one part,  
   the second as twice that width, the third as four times the  
   width of the first and the last as the same width as the first. */  
grid-template-columns: 1fr 2fr 4fr 1fr;
```

CSS Grid: grid-template-areas

- Grid template areas allow us to create named regions on our grid template.
- A dot/period means the cell should be blank.

```
grid-template-columns: 1fr 1fr 1fr 1fr;  
grid-template-areas:  
  "header header header header"  
  "main main . sidebar"  
  "footer footer footer footer";
```



CSS Grid: grid-area

- Items are added to the named area using the **grid-area** attribute.
- [CSS grid-template-areas property](#)

```
.item1 {  
  grid-area: myArea;  
}  
.grid-container {  
  display: grid;  
  grid-template-areas:  
    'myArea myArea . . .'  
    'myArea myArea . . .'  
  .';  
}
```


CSS Grid: grid-gap

- Grid gap is the space between the grid cells.
- CSS Syntax - `grid-gap: grid-row-gap grid-column-gap;`

```
.grid-container {  
  grid-gap: 20px 50px;  
}
```

What questions do you have?



[Grid Garden - A game for learning CSS grid](#)