

MODULE 3

CSS Layouts with Grid and Flexbox



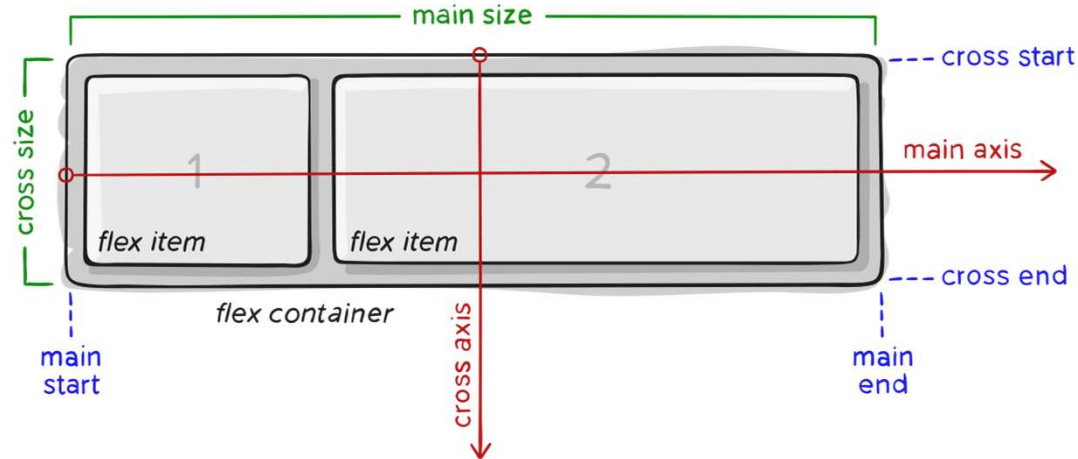


Yesterday

- What does HTML stand for?
- How is HTML organized?
- What does CSS stand for?
- What is a selector?
- What are some selectors?
- What is the difference between block and inline?

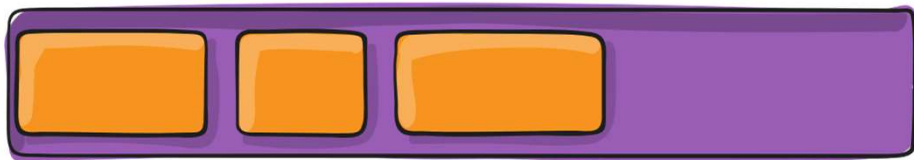
Modern Layout

- Flex box (a W3C Candidate Recommendation as of October 2017)
 - Providing a more efficient way to lay out, align and distribute space among items in a container
 - A flex container expands items to fill available free space, or shrinks them to prevent overflow

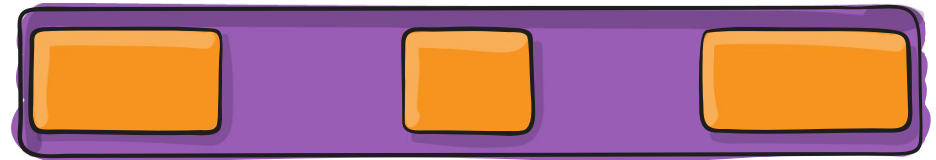


justify-content

flex-start



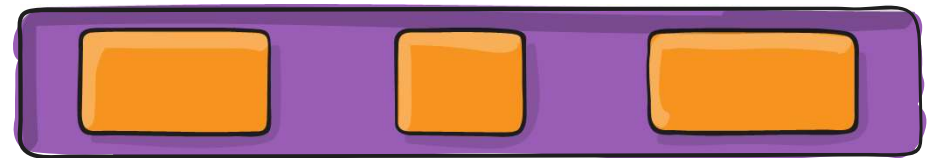
space-between



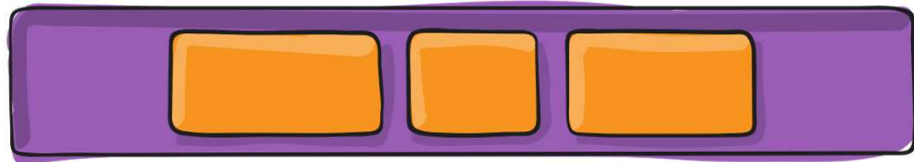
flex-end



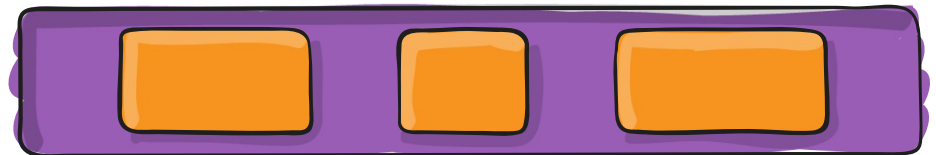
space-around



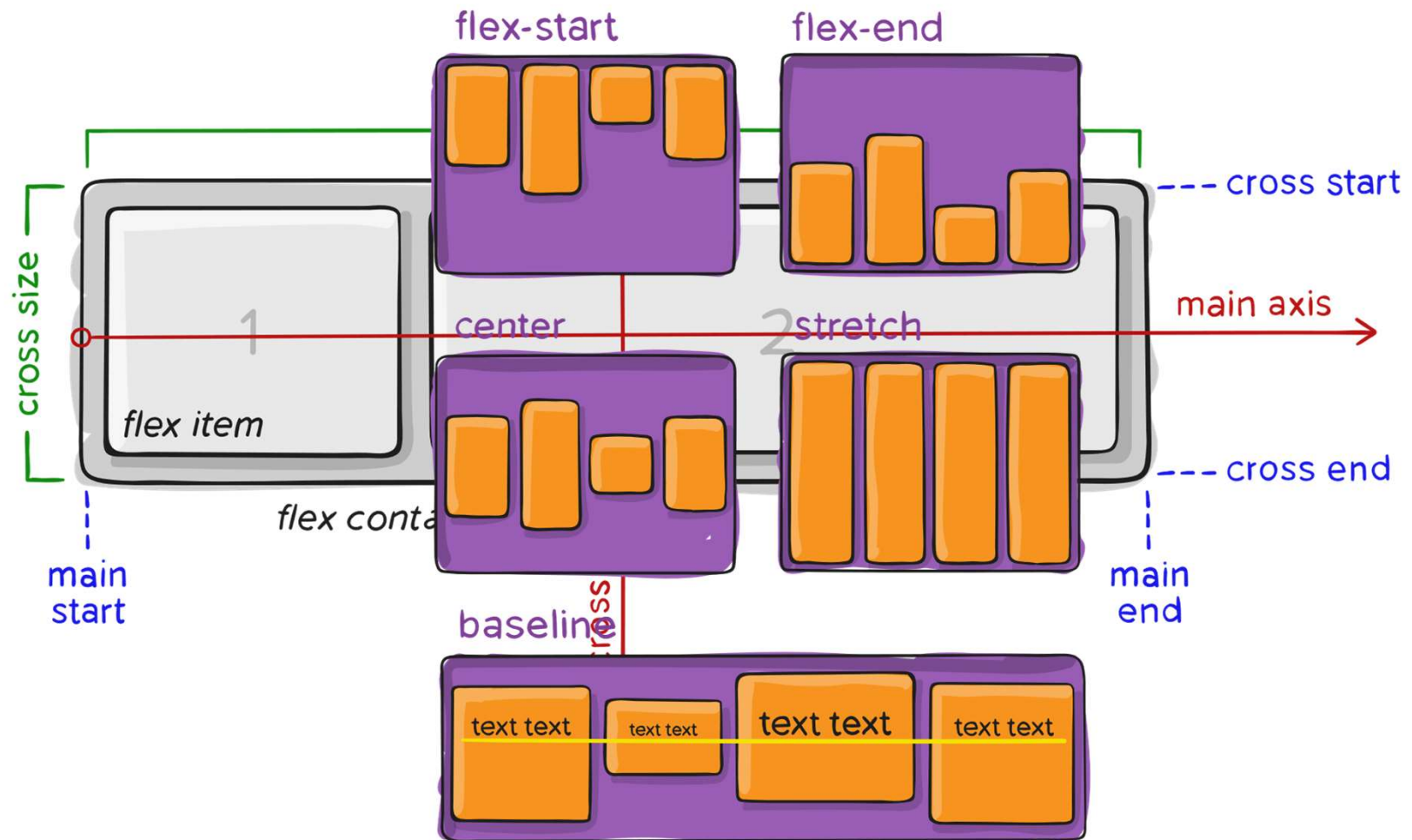
center



space-evenly



align-items



flex-wrap

- nowrap (default): all flex items will be on one line
- wrap: flex items will wrap onto multiple lines, from top to bottom.
- wrap-reverse: flex items will wrap onto multiple lines from bottom to top.

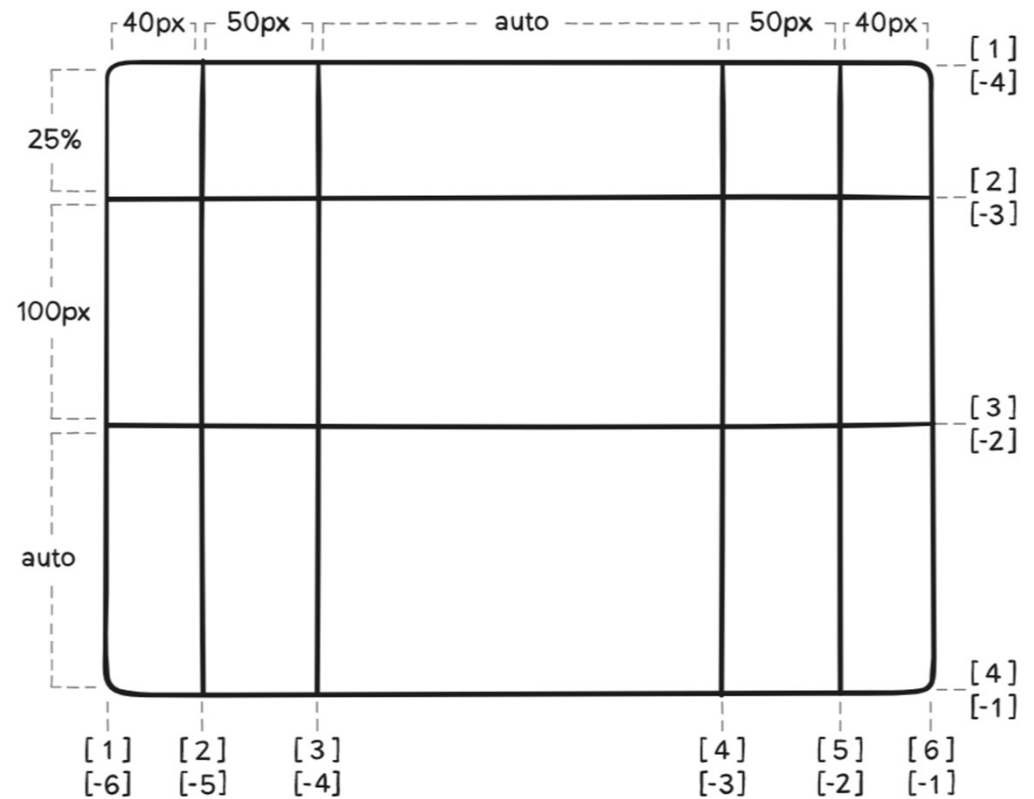
LET'S CODE!



ELEVATE  YOURSELF

CSS Grid Layout

- A two-dimensional grid-based layout system



grid-template-columns

- Lets you define the size of the columns of your grid
- Grid lines are defined as fractional units (fr)
- `grid-template-columns: 1fr 2fr 4fr 1fr;`

grid-template-areas

- Names for different sections

```
grid-template-columns: 1fr 1fr 1fr 1fr;
```

```
grid-template-areas:
```

```
"header header header header"
```

```
main main . sidebar
```

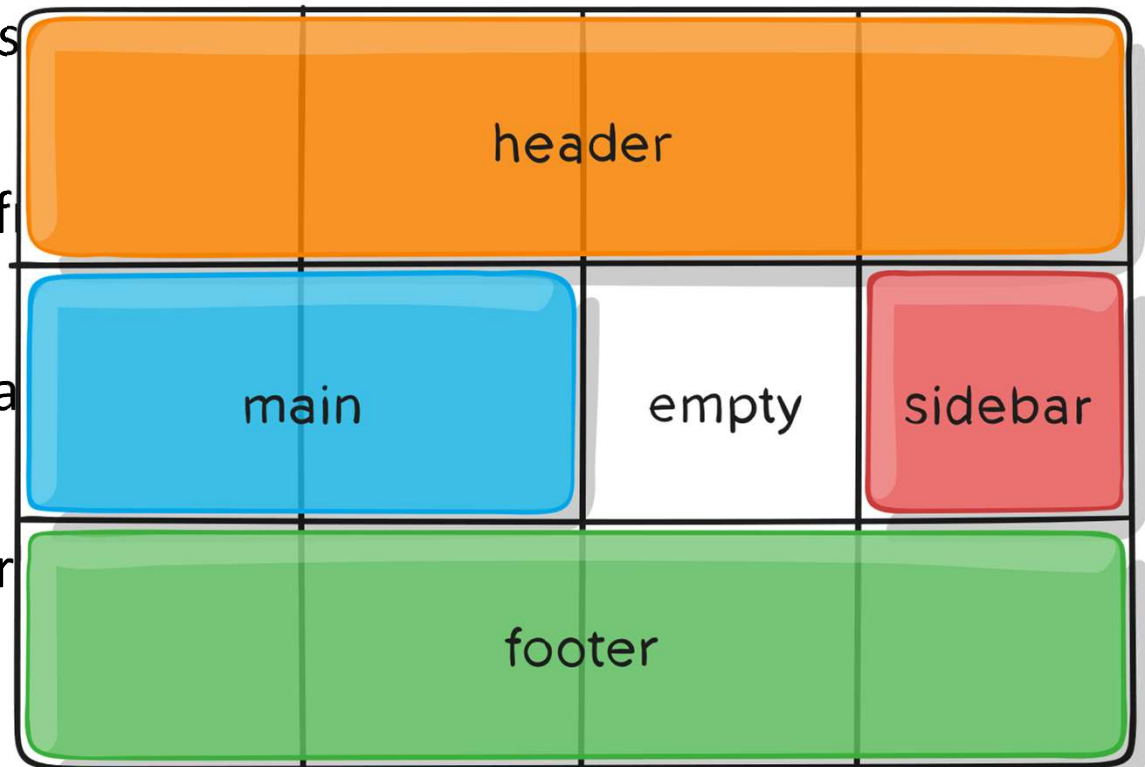
```
"footer footer footer footer";
```

```
grid-template-areas:
```

```
"header header header header"
```

```
"main main . sidebar"
```

```
"footer footer footer footer"
```



grid-area

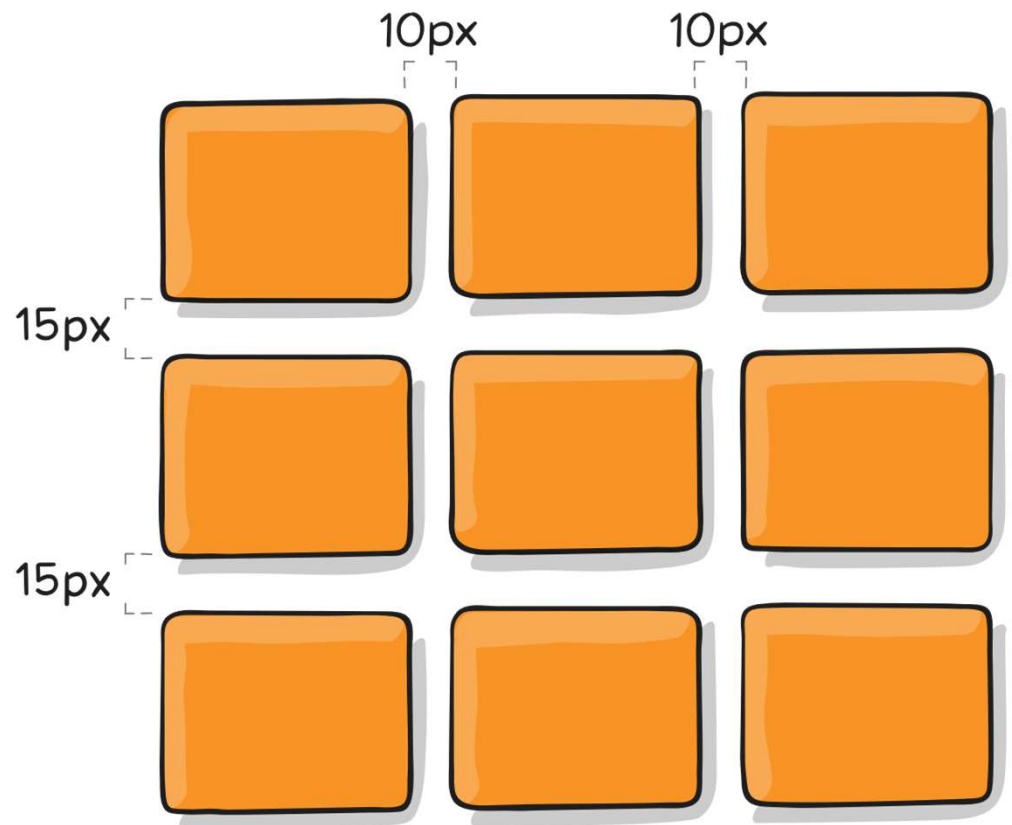
- Where a certain element will go in the grid
- The names will match the names that were defined in that element's grid-template-areas

```
.container {  
  grid-template-columns: 4fr 1fr 2fr;  
  grid-template-areas:  
    "header header header"  
    "main . sidebar"  
    "footer footer footer";  
}  
  
.item-a {  
  grid-area: header;  
}  
  
.item-b {  
  grid-area: main;  
}  
  
.item-c {  
  grid-area: sidebar;  
}  
  
.item-d {  
  grid-area: footer;  
}
```

grid-gap

- The space between the grid cells

```
.container {  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-column-gap: 10px;  
  grid-row-gap: 15px;  
}
```



LET'S CODE!



WHAT QUESTIONS DO
YOU HAVE?



Reading for tonight: **Introduction to MVC**

