

Module 3-2

CSS Layout

CSS Grids: Introduction

By defining a grid, we create a two dimensional layout composed of columns and rows allowing us to better organize our web page's contents.

CSS Grids: Defining

To define a grid we must specify a display attribute with a value of grid:

```
.myGrid {  
    display: grid;  
}
```

In this example, the CSS code will specify using a selector by class, that an html element with a class name of myGrid be defined as a grid.

CSS Grids: Columns and Grid Areas

grid-template-columns: This property defines the number of columns (and their respective width).

grid-template-areas: Matches each area of the grid to a specific HTML element. By virtue of how this works, it also defines the number of rows.

```
body {  
  display: grid;  
  grid-template-columns: 1fr 2fr 2fr 2fr 2fr 1fr;  
  grid-template-areas:  
    ". header header nav nav ."  
    ". main main main main ."  
    ". fall-festival fall-festival store store .";  
}
```

fr stands for fractional unit.
The 1st column will occupy 10% of the width, the second 20%, etc.

The period represents an empty space in the final grid layout.

Let's do some coding!

Flexbox: Introduction

Flexboxes are specialized containers that have within them several HTML elements arranged in a specific fashion.

The flexbox will also automatically adjust to external stimuli (i.e. resizing the window).

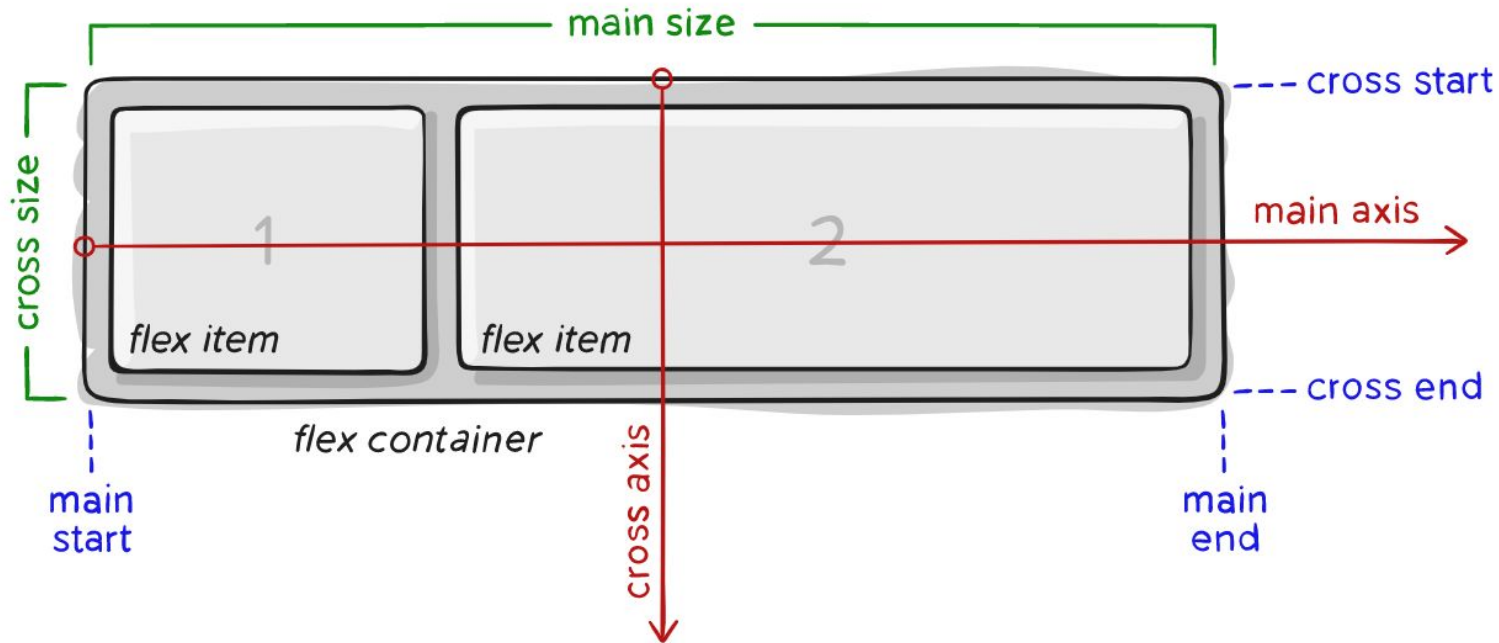
Flexbox: Defining

To define a flexbox we must specify a display attribute with a value of grid:

```
nav {  
    display: flex;  
}
```

Here, all HTML elements of type nav will be defined as flex boxes.

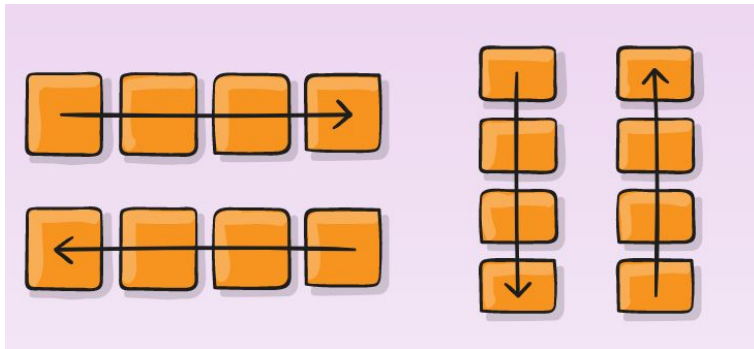
Flexbox: Anatomy



Flexbox: flex-direction

The flex-direction attribute determines whether or not the layout of items inside the box will be in either column or rows. **The default direction is row.**

```
nav {  
  display: flex;  
  flex-direction: column;  
}
```



Flexbox: justify-content

The justify-content property defines how items are aligned across the main axis.

```
nav {  
  display: flex;  
  justify-content: flex-start;  
}
```

flex-start



space-between



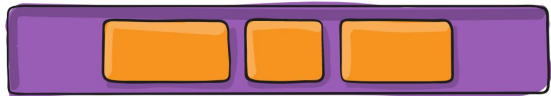
flex-end



space-around



center



space-evenly

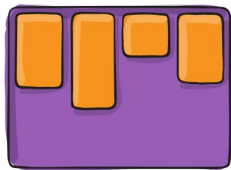


Flexbox: align-items

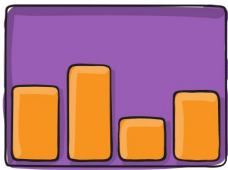
The justify-content property defines how items are aligned across the main axis.

```
nav {  
  display: flex;  
  align-items: flex-start;  
}
```

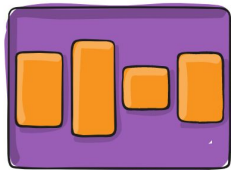
flex-start



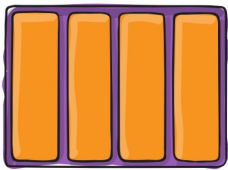
flex-end



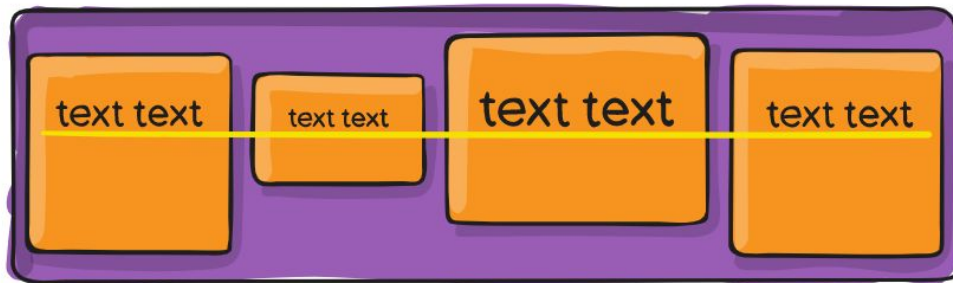
center



stretch



baseline



Let's do some coding!