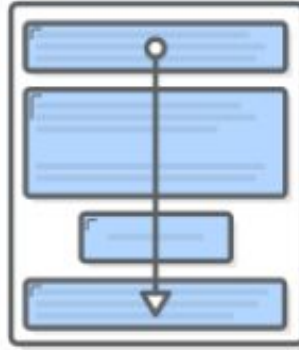
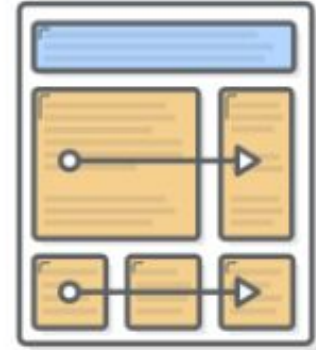


# < Edu-Prep-IX / >




VERTICAL FLOW



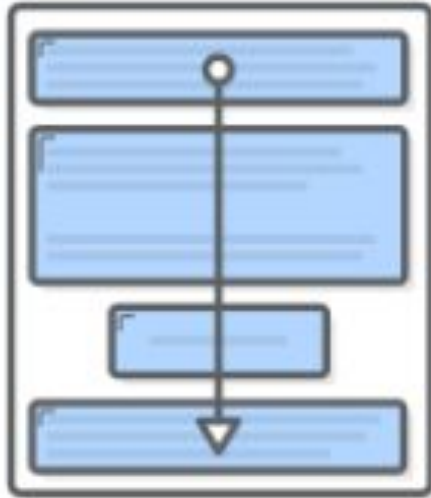
HORIZONTAL FLOW

# CSS

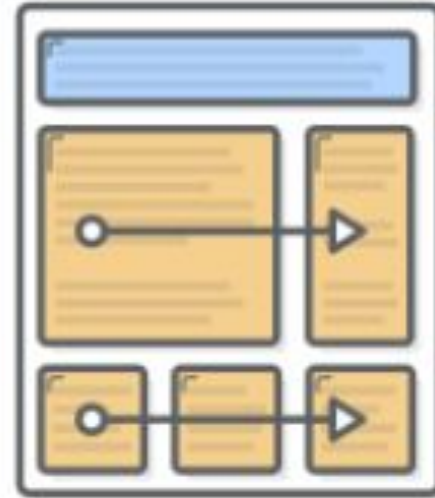
## *Layouts*



In the previous class we looked at the CSS Selectors. In CSS, **selectors** are used to **target the HTML elements** on our web page that we want to style. Today we shall cover CSS Layouts such as **Floats** and The Flexible Box or what is called **FlexBox**. Whereas floats only let us horizontally position our boxes, flexbox gives us complete control over the **alignment**, **direction**, **order**, and **size** of our boxes.



**VERTICAL FLOW**



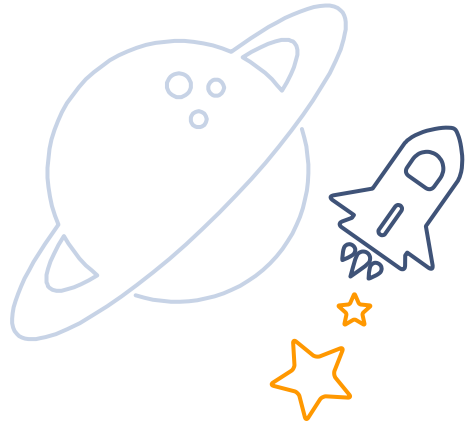
**HORIZONTAL FLOW**

# CSS Floats



**“Floats”** let you put block-level elements side-by-side instead of on top of each other. This is a big deal. It lets us build all sorts of layouts, including sidebars, multi-column pages, grids, and magazine-style articles with text flowing around an image. This is where we finally start creating real web pages.


# Float Property Values



The **float** property can have one of the following values:

- **left** - The element floats to the left of its container
- **right** - The element floats to the right of its container
- **none** - The element does not float (will be displayed just where it occurs in the text).  
This is default
- **inherit** - The element inherits the float value of its parent

In its simplest use, the **float** property can be used to wrap text around images.



Float-based layouts have mostly been replaced with Flexbox in modern websites. Whereas floats only let us horizontally position our boxes, flexbox gives us complete control over the alignment, direction, order, and size of our boxes.

# CSS FlexBox



The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using **float** or **positioning**. FlexBox Makes it much easier to work on Website layouts both Horizontally and Vertically



# FlexBox In Use



**ALIGNMENT**



**DIRECTION**



**ORDER**



**SIZE**

# Flexbox and Floats



## FLOATS

(MAGAZINE-STYLE LAYOUTS)



## FLEXBOX

(OVERALL PAGE STRUCTURE)

# FlexBox Overview



Flexbox uses two types of boxes that we've never seen before: “flex containers” and “flex items”. The job of a flex container is to group a bunch of flex items together and define how they're positioned.



"FLEX CONTAINER"



"FLEX ITEMS"

# Flex Container



The first step in using flexbox is to turn one of our HTML elements into a flex container. We do this with the **display property**, which should be familiar from the CSS Box Model chapter. By giving it a value of **flex**, we're telling the browser that everything in the box should be rendered with flexbox instead of the default box model. *This enables the flexbox layout mode—without it, the browser would ignore all the flexbox properties that we're about to introduce.*



The diagram shows a container box with a thick border. Inside, there are four smaller boxes. The top box is labeled 'NOT FLEXBOX' and 'DISPLAY: BLOCK;'. The middle box is labeled 'FLEXBOX' and 'DISPLAY: FLEX;', and it is highlighted with a blue background. The bottom-left box is labeled 'NOT FLEXBOX' and 'FLOAT: LEFT;'. The bottom-right box is labeled 'NOT FLEXBOX' and 'FLOAT: RIGHT;'. The 'FLEXBOX' box is wider than the two 'NOT FLEXBOX' boxes below it, demonstrating how Flexbox allows for different widths within a container.

**NOT FLEXBOX**

DISPLAY: BLOCK;

**FLEXBOX**

DISPLAY: FLEX;

**NOT FLEXBOX**

FLOAT: LEFT;

**NOT FLEXBOX**

FLOAT: RIGHT;

# Aligning an Item



After you've got a flex container, your next job is to define the horizontal alignment of its items. That's what the **justify-content** property is for.

```
.menu-container {  
  /* ... */  
  display: flex;  
  justify-content: center; /* Add this */  
}
```



**FLEX-START**



**CENTER**



**FLEX-END**

Other values for `justify-content` are shown below:

- `center`
- `flex-start`
- `flex-end`
- `space-around`
- `space-between`



# Distributing Items



The justify-content property also lets you distribute items equally inside a container.



SPACE-AROUND




SPACE-BETWEEN

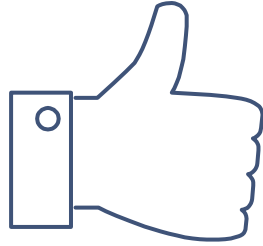
# Resources on CSS Layouts

*There is more to Flex-box please go through these resources.*

\*\* Try the Freecodecamp section on CSS Flexbox

- W3Schools - [CSS Layouts](#)
- Mozilla Developer Network - [CSS Layouts](#)

- 
- Do some practice as as regards what we have covered so far.
  - Utilize content on [FreeCodeCamp](#) and also use the resources in the curriculum.
  - Reach out incase you have a question.



*THANKS!*

#Read, Search and Don't be Afraid to ask