



CSS Flexbox

Modern Web Layout

Week 1 – Full-Stack DPP Course



Welcome

Manuel Cubuca

Technical Trainer

manuelc@justit.co.uk



Outline

- Web Layout History and Evolution
- What's Flexbox
- Flexbox Basics
- Flexbox Container
- Flex Flow Direction
- Flex Line Wrapping
- Display Order
- Flexibility
- Alignment



Learning Objectives

- Learners would have a good understanding of:
 - Layout Evolution From Tables, CSS1/2, Float Based Layout to Responsive Web Design
 - Flexible Box Model (as an Extension of the Display Property)
 - Core Concepts and Key Terminologies of Flexbox
 - Flexible Layouts
 - Distribute Extra Space
 - Align Content
 - Main Axis vs. Cross Axis



Learning Objectives cont.

- Learners would have a good understanding of:
 - Flow Content Horizontally o Vertically
 - Align Items on the Container or Individually
 - Stretch, Shrink and Grow
 - Change Display Order
 - Real World Examples



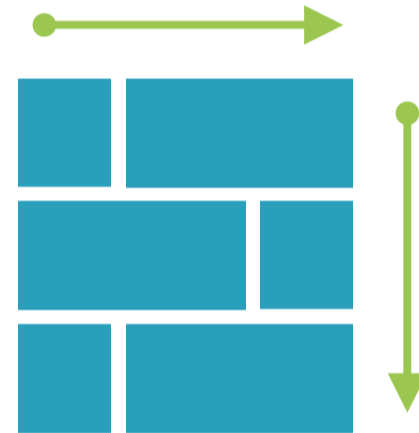
Flexbox vs. CSS Grid Layout

They are similar but Flexbox controls how items flow in **one dimension**, whereas, grid controls how items flow in **two dimensions**.



Flexbox

ONE DIMENSION



CSS Grid Layout

TWO DIMENSIONS



What is Flexbox

Short for **Flexible Box Model**

- Is an extension of the **Display Property**

A Set of CSS
Properties

Creating
Flexible
Layouts

Distributing
Extra Space

Aligning
Content



Layout Mode From CSS2.1

Did not provide enough **Flexible and Control for the web as we known it today**, which is why Flexbox was created.

Not Flexible Enough!

Block

Inline

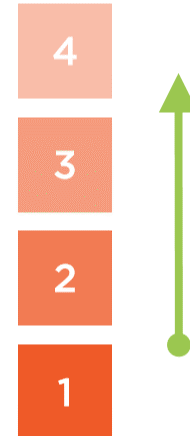
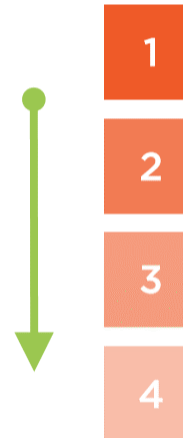
Table

Positioning



Flexbox

Provides the ability to arrange items in any directions, left to right, right to left, top to bottom or bottom to top.



Flexbox

Provides the ability to easily reverse or rearrange the visual display order of a set of items.

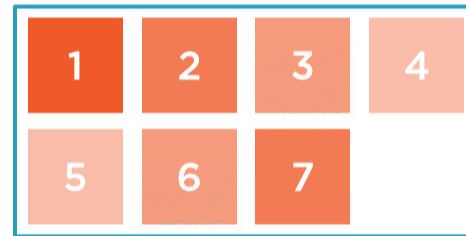


Flexbox

Provides the ability to lay items out linearly without wrapping or forcing them to wrap, when there is not enough room.



↑
NO WRAP

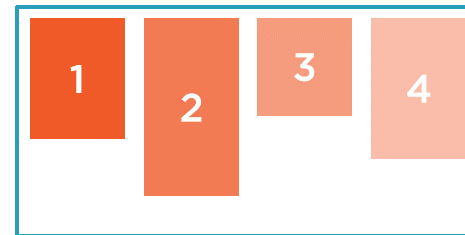
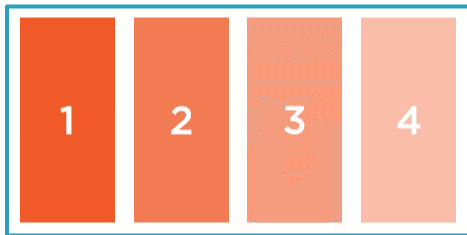


↑
WRAP



Flexbox

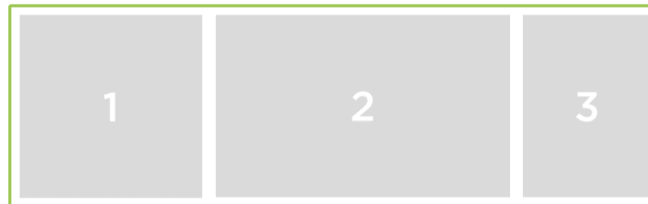
Ability to stretch and align items in relation to their parent container.



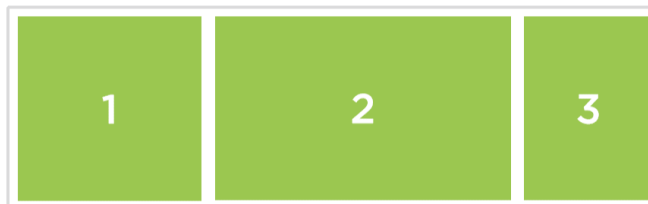
Flexbox the Basics

Flexbox is all about the relationship between a **PARENT CONTAINER** and its **direct** descendant's **CHILD ITEMS**.

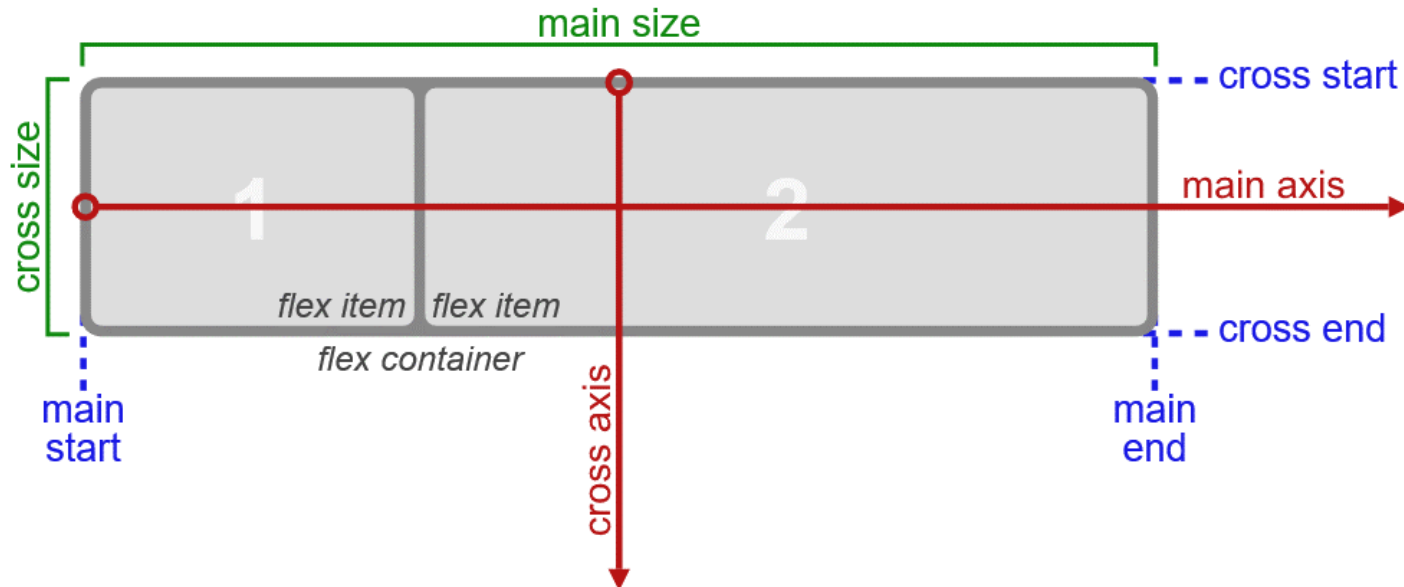
- The parent container is known as **FLEX CONTAINER**



- The direct descendants of this container are **FLEX ITEMS**



Main Axis vs Cross Axis



The Flex Container

Forms the containing block for its Flex Items and it acts very similar to other block level items.

To define or establish an element as Flex Container, we use the **DISPLAY PROPERTY**.

display: flex;
display: inline-flex;



Flexbox 1st Demo

Let's practice



Flex Flow Container

By default flex items will flow **horizontally** from left to right within their container.

To make it act differently, we can change this with a new flexbox property.

`flex-direction: row;`

This is the default behaviour



Flex Flow Container

Flow from right to left:

flex-direction: row-reverse;

Flow from top to bottom:

flex-direction: column;

Flow from bottom to top:

flex-direction: column-reverse;



Flexbox 2nd Demo

Let's practice



Flex Line Wrapping

How flex items respond when there's not enough room and the ways we can change their behaviours to suite our needs.

If you used **text-wrap** property before, it works in a very similar manner with **Flex Line Wrapping**.

`flex-wrap: nowrap;`

This is the default behaviour

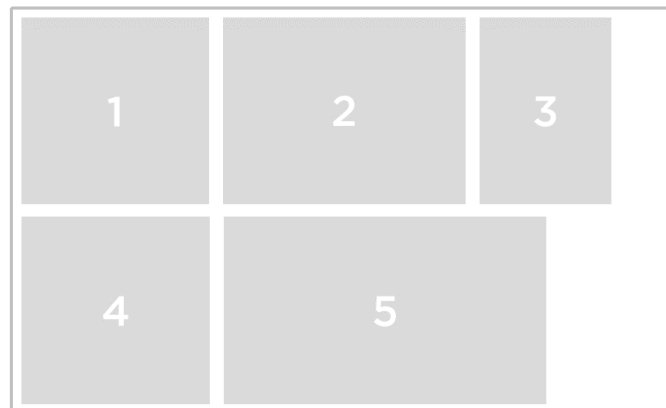


Single Line Formatting vs Multi-line Formatting

flex-wrap: nowrap;



flex-wrap: wrap;



Flex Line Wrapping **cont.**

Make items wrap above the first row when there's not enough room, items will flow from bottom to top.

flex-wrap: wrap-reverse;

We can also use **shorthand** to combine both flex direction property and flex wrap property with **flex-flow**.

flex-flow: column wrap;

We can use other combinations such as:

flex-flow: row-reverse wrap-reverse;



Flexbox 3rd Demo

Let's practice



Flex Display Order

How easy its to control the display order of flex items. You can easily change the order flex items are display within their container.

Source Order

```
<div class="flex-container">  
  <div class="flex-item">  
    1  
  </div>  
  <div class="flex-item">  
    2  
  </div>  
  <div class="flex-item">  
    3  
  </div>  
</div>
```

Display Order



Flex Display Order cont.

By default all flex items within a container are placed into one ordinal group, which has a value of **ZERO**.

We can use a new flex property called **order** to change the display order.

order: 0; // This is the default

To move the item to the very end (right side)

order: 1;

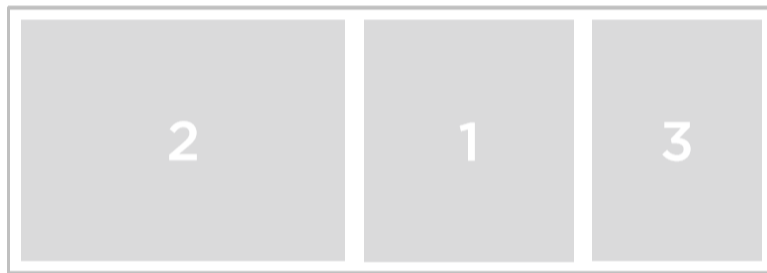
To move the item to the very beginning (left side)

order: -1;



Accessibility

The order property only changes the **visual order** of the content and not the **source order**. The key is to use order property only to alter the items for visual purposes only.



```
<div class="flex-container">  
  <div class="flex-item">  
    1  
  </div>  
  <div class="flex-item">  
    2  
  </div>  
  <div class="flex-item">  
    3  
  </div>  
</div>
```



Flexbox 4th Demo

Let's practice



Flexibility

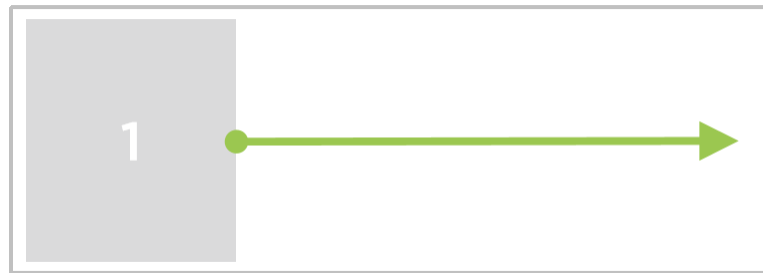
The most notable component of Flexbox is the ability to make flex items flex, to shrink and grow to feel available space within a flex container.

Flexible Box Model was created to allow us to more easily create flexible layouts that can change and adapt to their viewing environment.

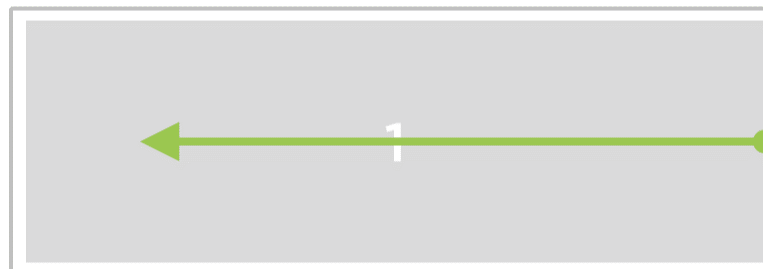


Flex Grow Factor and Flex Shrink Factor

flex-grow: 1;

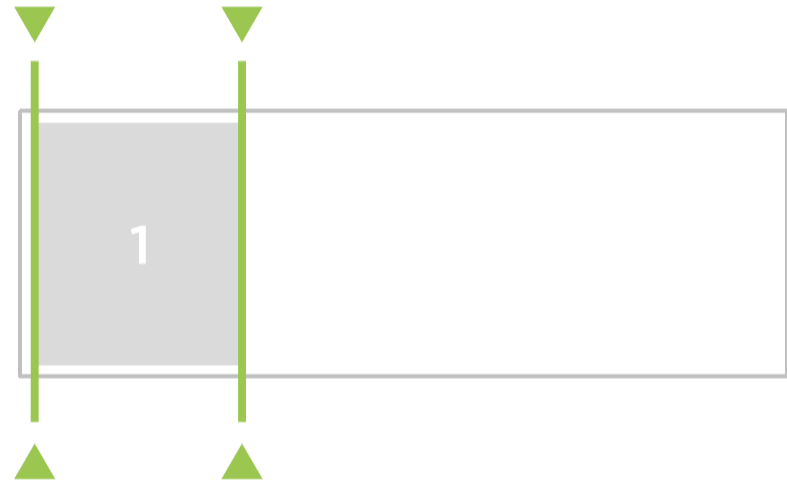


flex-shrink: 2;



Flex Basis Property and Shorthand

flex-basis: 10em;



The shorthand:

flex: flex-grow flex-shrink flex-basis;



Flexbox 5th Demo

Let's practice



Alignment

In CSS 2.1 we often find it very difficult to align items. To vertically align an item in a given container it requires very hacky CSS.

Flexbox leverages the CSS box alignment module, which allows us to use simple keywords to align items.

Flexbox alignment can be handled in two different ways:

- Can be applied to all the flex items in a given container via the **justify-content, align-items, or align-content properties**.
- Also, it can be applied to the individual flex items themselves via the **align-self property**.



Alignment cont.

1 2 3

1 2 3

1 2 3

1 2 3

1 2 3

1 2 3

1 2 3

1 2 3

1 2 3 4

1 2 3

4 5 6

1 2 3

4 5 6

justify-content align-items align-content



Alignment justify-content

Allows us to align the flex container's items along the main axis.

Notice the flex-start is the default:

justify-content: flex-start;

justify-content: flex-end;

Can be combine with flex-direction:

justify-content: flex-end;

flex-direction: column;



Alignment justify-content cont.

Can be combine with flex-direction to centre items:

justify-content: center;

flex-direction: column;

Can also add space between the items:

justify-content: space-between;

justify-content: space-around;



Align Items

align-items: flex-start;

align-items: flex-end;

align-items: flex-center;

align-items: baseline; // aligns the txt at same level

align-items: stretch; // is the default value



Align Self

To align individual items uniquely, can be achieved with the **align-self** property.

align-self: flex-start;

align-self: flex-end;

align-self: stretch;



Align Content

Will do nothing to single line flex containers.

align-content: flex-start;

align-content: flex-end;

align-content: center;

align-content: space-between;

align-content: space-around;

align-content: stretch; // is the default



Flexbox 6th Demo

Let's practice



Resources

- Pluralsight video tutorials

**Modern Web Layout with
Flexbox and CSS Grid**

**CSS Flexbox
Fundamentals**



Resources cont.

- Relevant websites for sample codes, examples and general information.

Browser Support

CSS Layout News

W3C

**A Complete Guide
to Flexbox**

**A visual Guide to
Flexbox Properties**

Solved by Flexbox

