



Praktikum 6 - Web **RWD**



Normal Flow



```
<h1>Basic document flow</h1>
```

```
<p>I am a basic ...</p>
```

```
<p>By default we span ...</p>
```

```
<p>We are separated ...</p>
```

```
<p>inline elements <span>like this one</span> and</p>
```

Basic document flow

I am a basic block level element. My adjacent block level elements sit on new lines below me.

By default we span 100% of the width of our parent element, and we are as tall as our child content. Our total width and height is our content + padding + border width/height.

We are separated by our margins. Because of margin collapsing, we are separated by the width of one of our margins, not both.

inline elements like this one and this one sit on the same line as one another, and adjacent text nodes, if there is space on the same line. Overflowing inline elements will wrap onto a new line if possible (like this one containing text), or just go on to a new line if not, much like this image will do:



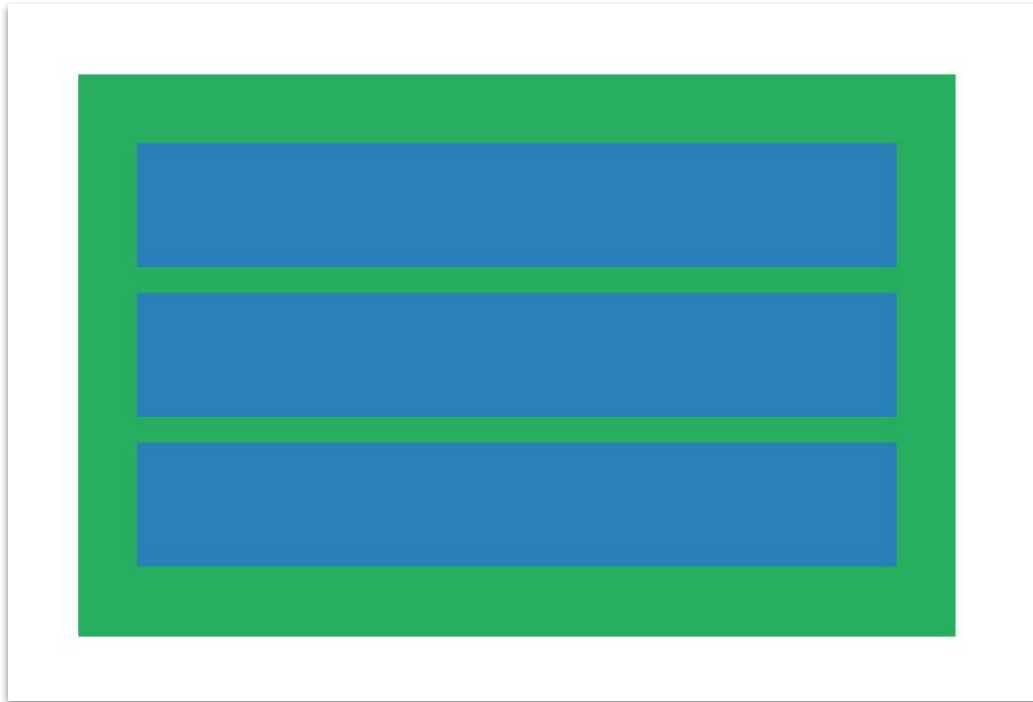
Flexbox - Terminology

Flexbox:

- Flex Container
- Flex Item



Flexbox - Create First Flex



Flexbox Container - Display



Flex Container:

- `display: flex`



Flexbox Container - Direction



Flex Container:

- flex-direction: row
- flex-direction: column



Flexbox Container - Wrap



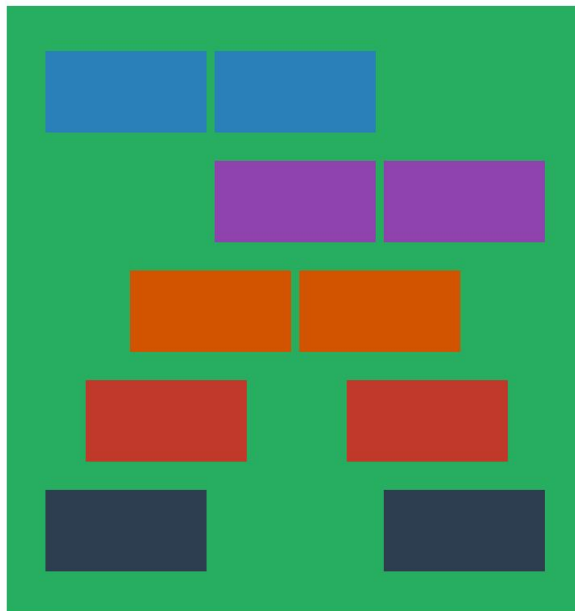
Flex Container:
- `flex-wrap: wrap`



Flexbox Container - Justify Content

Flex Container:

- `justify-content: flex-start`
- `justify-content: flex-end`
- `justify-content: center`
- `justify-content: space-around`
- `justify-content: space-between`

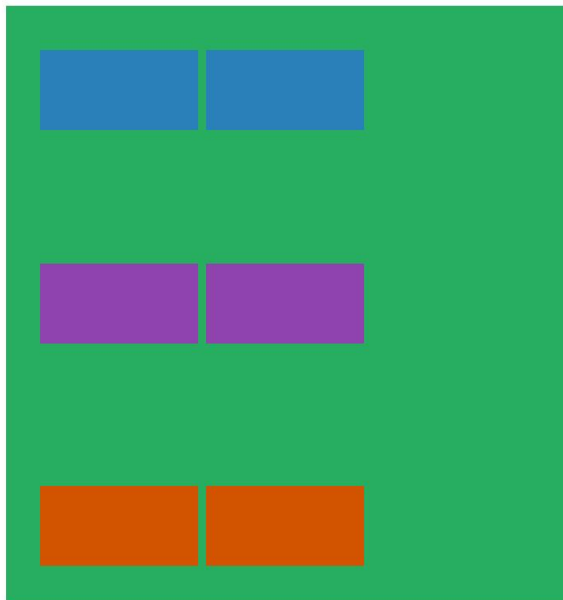


Flexbox Container - Align Items



Flex Container:

- align-items: flex-start
- align-items: flex-end
- align-items: center

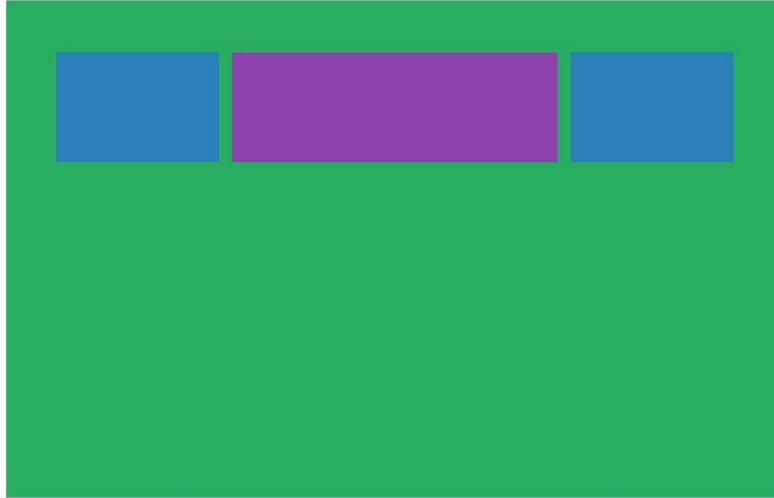


Flexbox Item - Flex Grow



Flex Item:

- flex-grow: 2

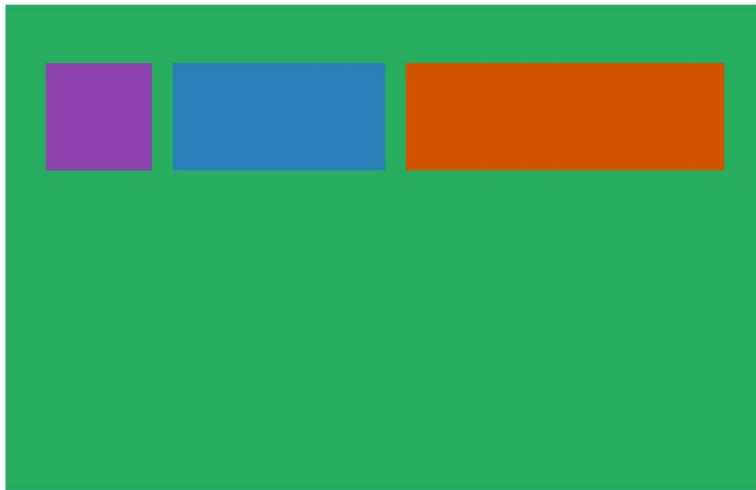


Flexbox Item - Flex Basis



Flex Item:

- flex-basis: 100px
- flex-basis: 200px
- flex-basis: 300px



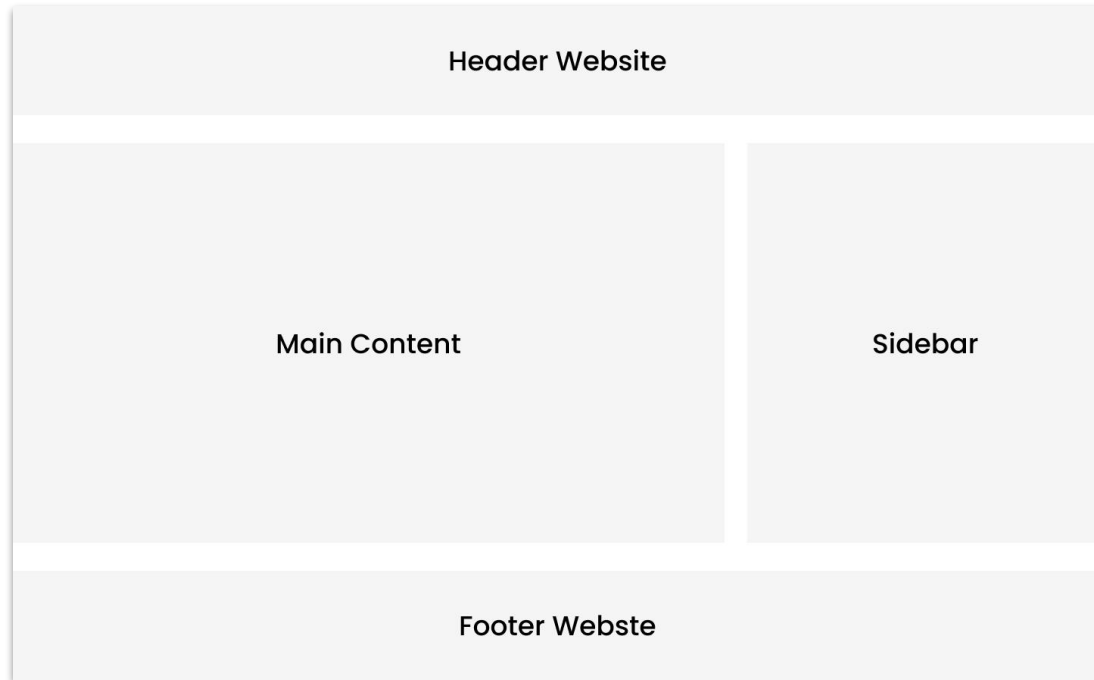
Flexbox Item - Flex Order



Flex Item:
- order: 3



Flexbox - Example



RWD - Terminology



A set of practices that allows web pages to alter their layout and appearance to suit different screen widths, resolutions, etc. It is an idea that changed the way we design for a multi-device web

RWD - Viewport



```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

RWD - Media Query



```
@media screen and (min-width: 800px) {  
  
}  

```


RWD - Breakpoint



Breakpoint

- small: 576px
- medium: 768px
- large: 992px

```
@media screen and (max-width: 576px) {  
  
}  
  
@media screen and (min-width: 768px) {  
  
}  
  
@media screen and (min-width: 992px) {  
  
}
```

RWD - Example

