Page layout

Web development I: Front-end engineering

Page layout strategies

Fixed layouts stay at a specific pixel width regardless of the size of the browser window or text size.

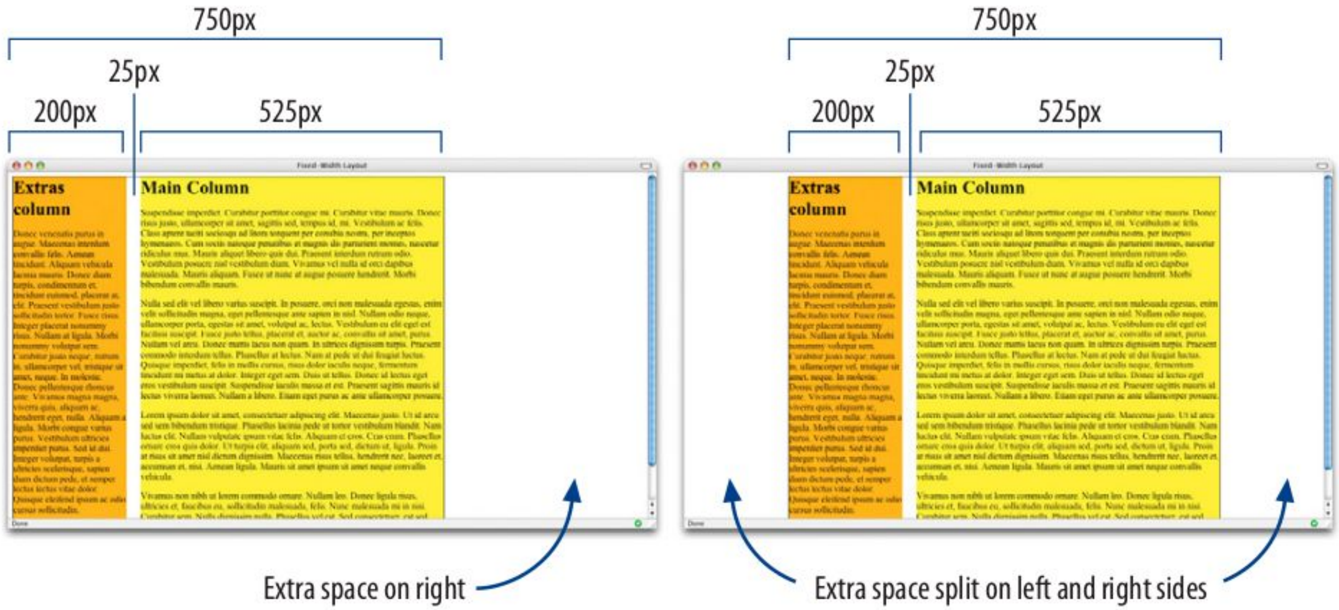
Fluid (or liquid) layouts resize proportionally when the browser window resizes.

Elastic layouts resize proportionally based on the size of the text.

Hybrid layouts combine fixed and scalable areas.

Fixed layouts

```
#wrapper {width: 750px;
  position: absolute;
  margin-left: auto;
  margin-right: auto;
  border: 1px solid black;
  padding: 0px;}

#extras {position: absolute;
  top: 0px;
  left: 0px;
  width: 200px;
  background: orange; }

#main {margin-left: 225px;
  background-color: yellow;}
```



Fixed layouts

Advantages:

Predictable layout.

Better control over line length.

Easier to design and produce.

Disadvantages:

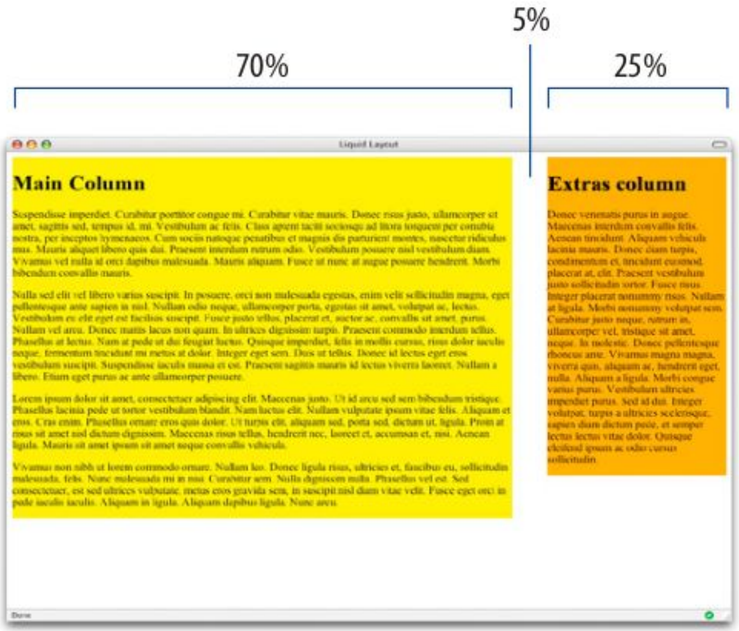
Horizontal scrolling on small screens.

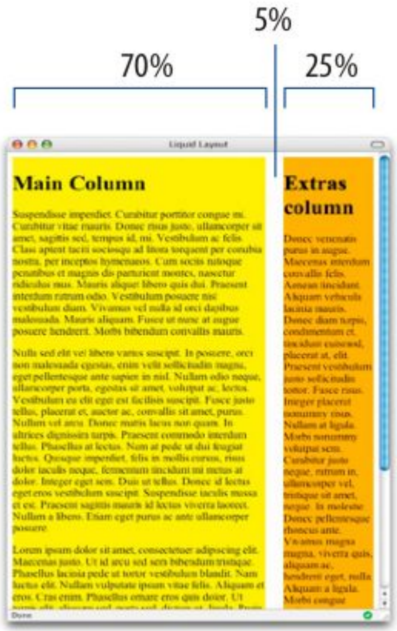
Lots of whitespace on large screens.

Short line lengths at large text sizes.

Not user-controllable.

Fluid layouts





```
div#main {
    width: 70%;
    margin-right: 5%;
    float: left;
    background: yellow;
    }

div#extras {
    width: 25%;
    float: left;
    background: orange;
    }
```

Fluid layouts

Advantages:

Follow the nature of the Web.

Avoid potentially empty spaces.

Users can control the width of content.

No horizontal scrollbars.

Disadvantages:

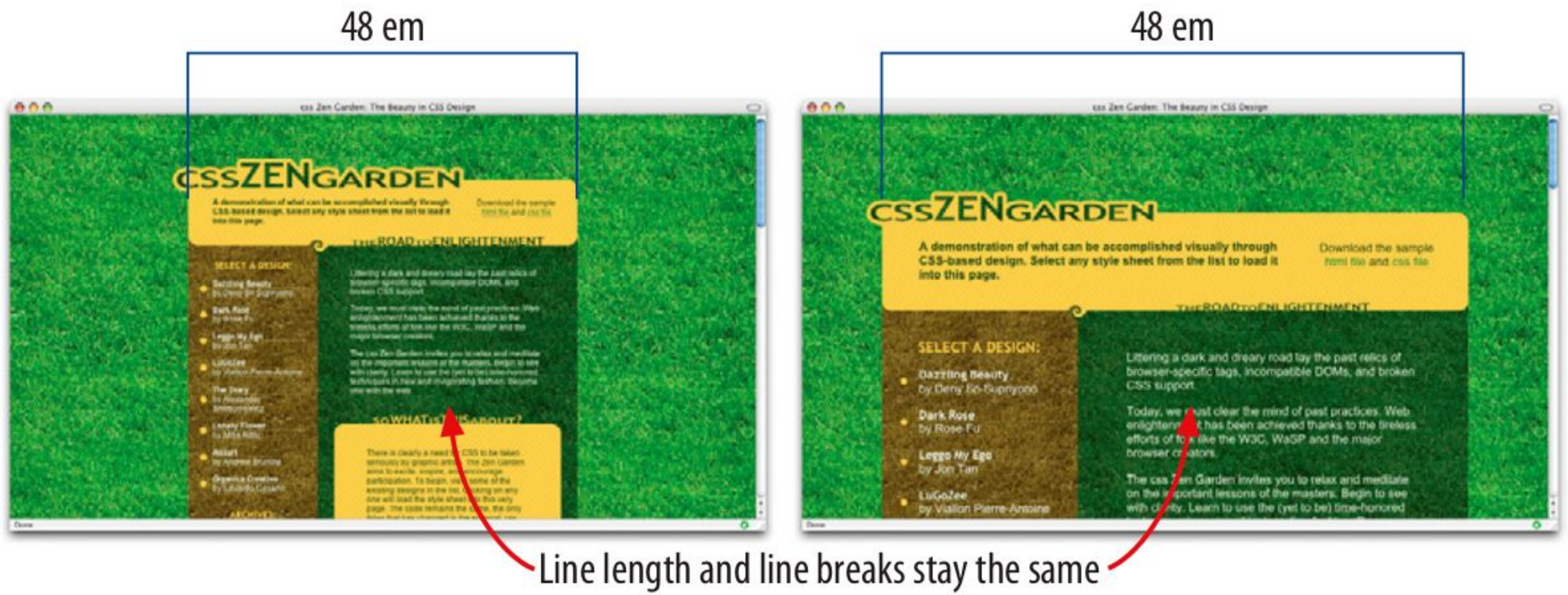
On large screens, line lengths can get very long.

Less predictable: elements may be too spread out or too cramped.

More difficult to achieve whitespace.

More complex calculation of measurements.

Elastic layouts



Elastic layouts

Advantages:

Consistent layout while flexible text sizes.

Better control over line lengths.

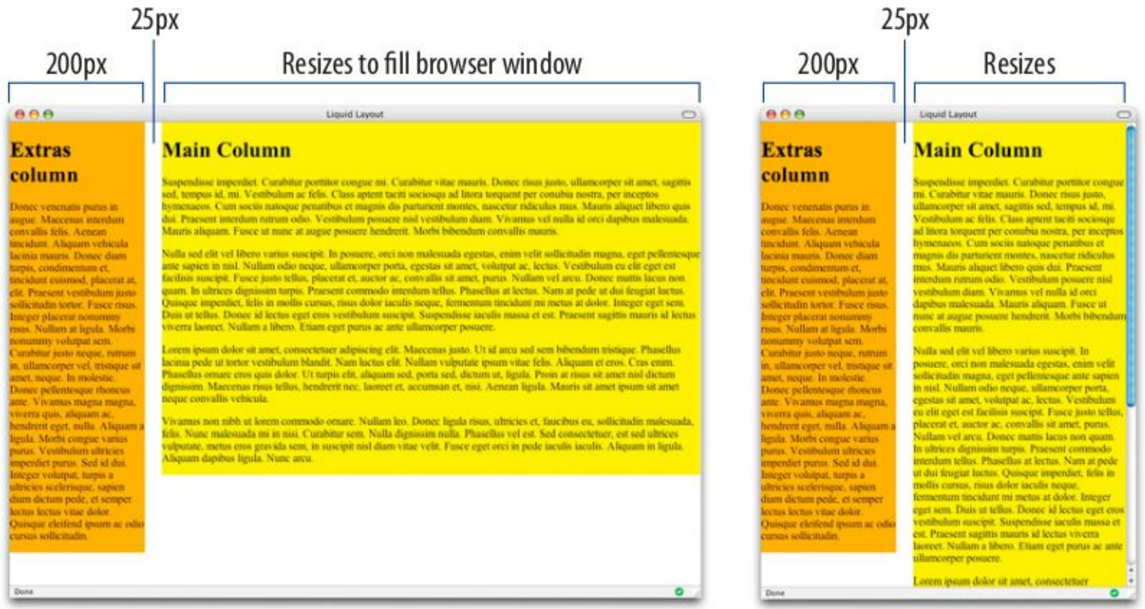
Disadvantages:

Rescaling of media is more challenging.

On large screens, window width might exceed device width.

More complicated to create than fixed-width layouts.

Hybrid layouts

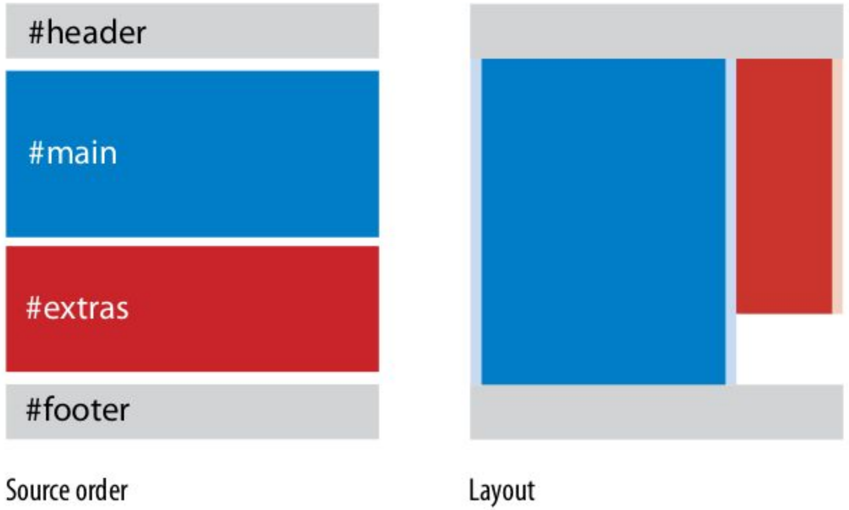


```
div#main {
    width: auto;
    position: absolute;
    top: 0;
    left: 225px;
    background: yellow; }

div#extras {
    width: 200px;
    position: absolute;
    top: 0;
    left: 0;
    background: orange; }
```

Page layout techniques

Two columns, fluid layout

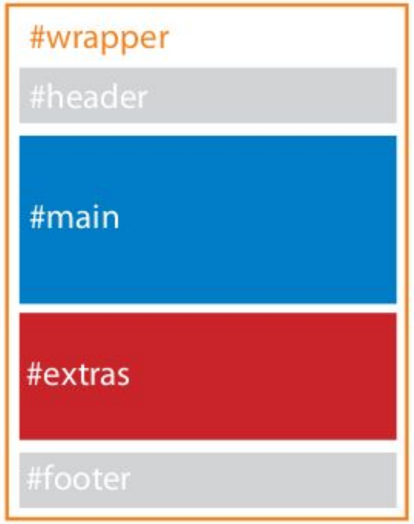
```
#main { float: left; width: 60%; }

#extras { float: left; width: 25%; }

#footer { clear: left; }
```



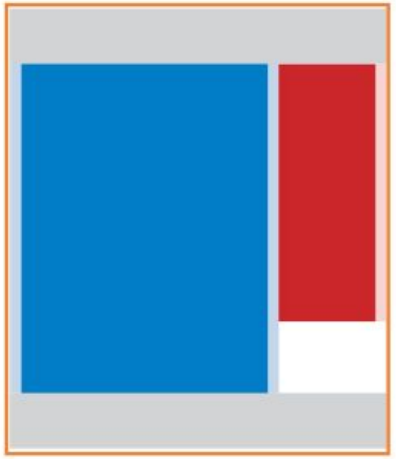
Page layout techniques

Two columns, fixed-width layout

```
#wrapper { width: 960px; }

#main { float: left; width: 650px; }

#extras { float: left; width: 250px; }

#footer { clear: left; }
```



Source order

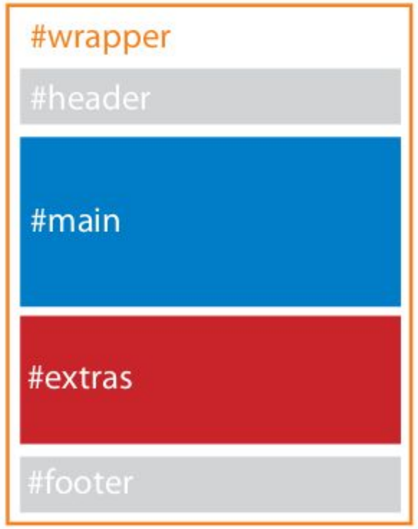


Layout

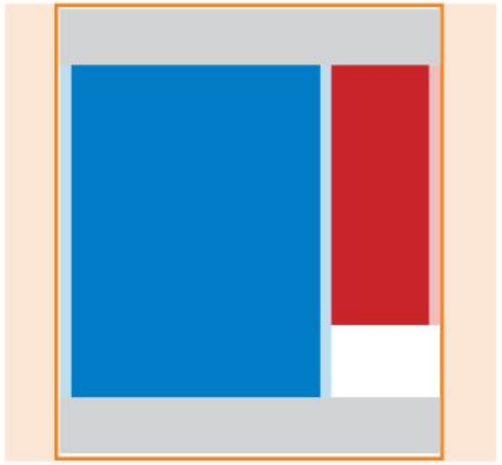
Page layout techniques

Two columns, fixed width, centered

```
#wrapper { width: 960px; margin: 0 auto; }

#main { float: left; width: 650px; }

#extras { float: left; width: 250px; }

#footer { clear: left; }
```



Source order

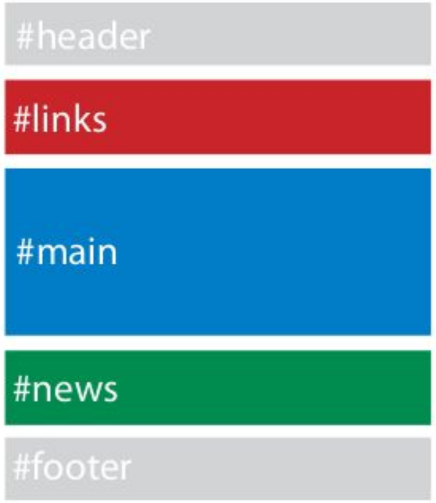


Layout

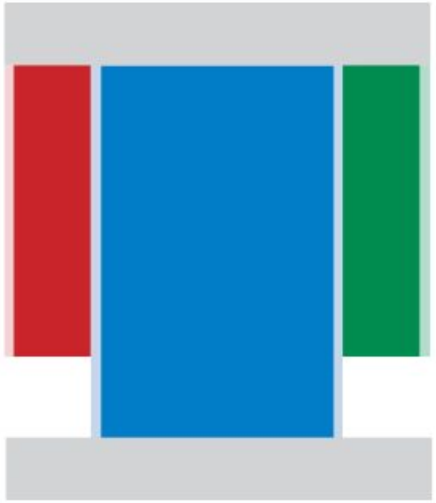
Page layout techniques

Three columns, fluid layout

```
#links { float: left; width: 22.5%; }

#main { float: left; width: 45%; }

#news { float: left; width: 22.5%; }

#footer { clear: left; }
```



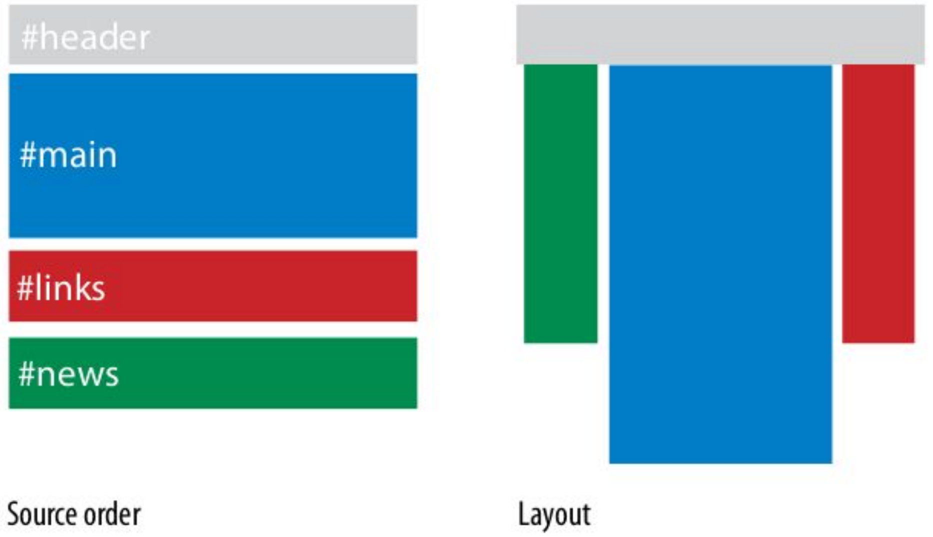
Source order


Layout

Page layout techniques

Three columns, positioned, fluid layout

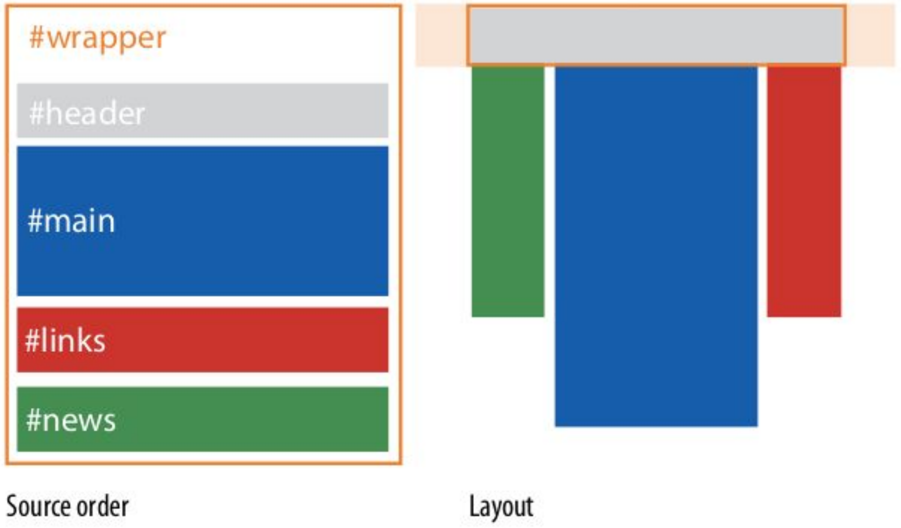
```
#content { position: relative; }

#main { width: 50%; position: absolute;
top: 0; left: 25%; }

#news { width: 20%; position: absolute;
top: 0; left: 2.5%; }

#links { width: 20%; position: absolute;
top: 0; right: 2.5%; }
```



Page layout techniques

Three columns, positioned, fixed

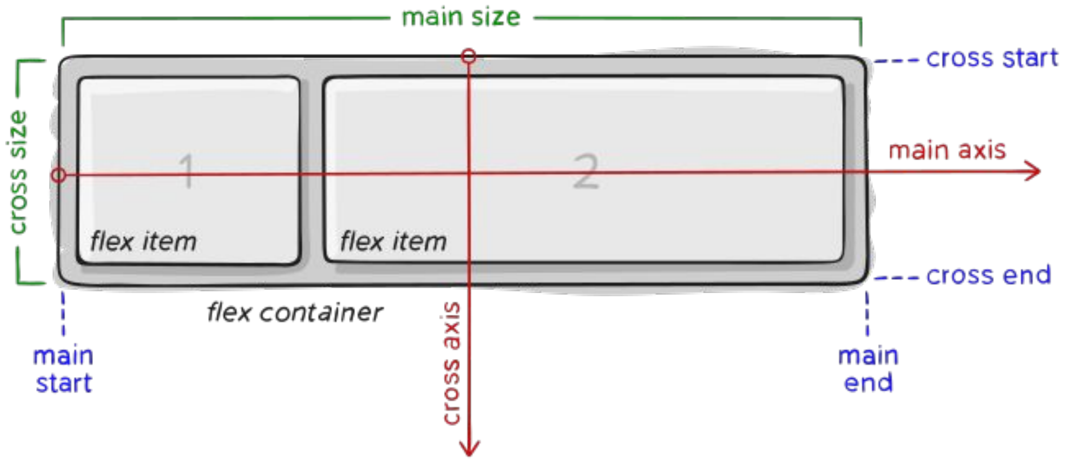
```
#wrapper { width: 960px; margin: 0 auto; }

#main { width: 520px; position: absolute;
top: 0; left: 220px; }

#news { width: 200px; position: absolute;
top: 0; left: 0; }

#links { width: 200px; position: absolute;
top: 0; right: 0; }
```



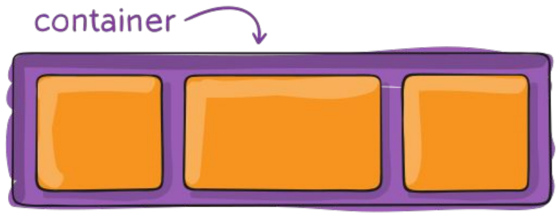
Flexbox

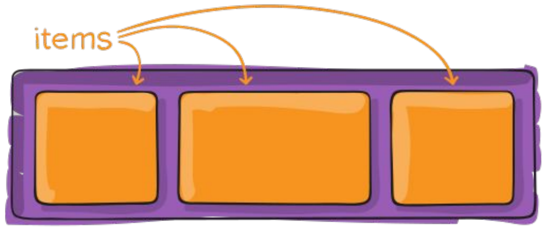
Regular layouts are based on block and inline flow directions.

The flex layout is based on **flex-flow directions**.



Flexbox



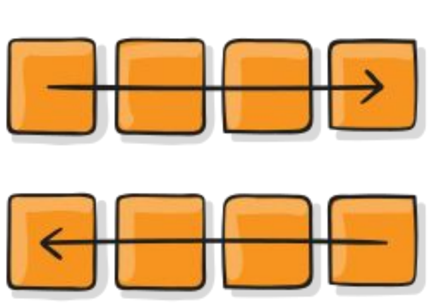


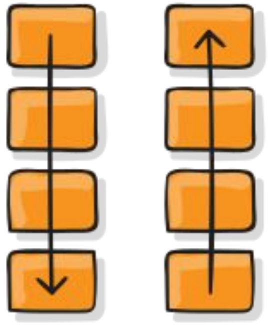
`display`	`order`
`flex-direction`	`flex-grow`
`flex-wrap`	`align-self`
`flex-flow`	`flex-shrink`
`justify-content`	`flex-basis`
`align-items`	`flex`
`align-content`	

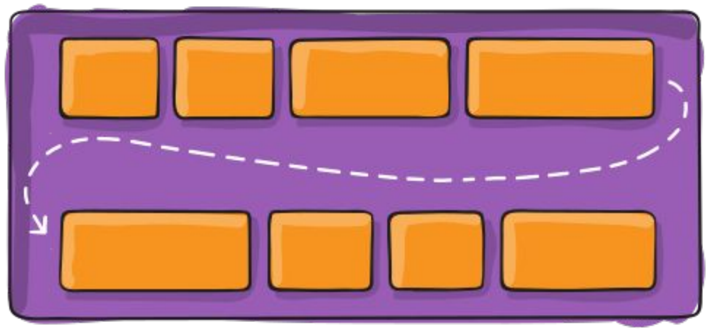
Flexbox: basic markup

```
.container {
  display: flex;
}
.items {
  flex: 1;
}
```

Flexbox: container







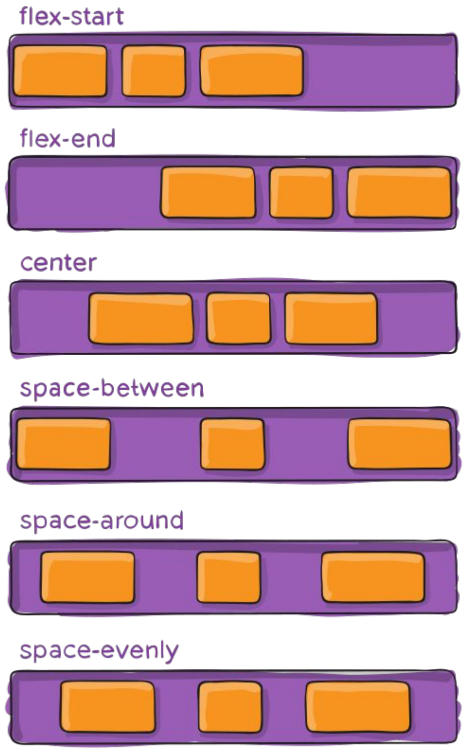
flex-direction: row | row-reverse | column | column-reverse

flex-wrap: nowrap | wrap | wrap-reverse;

Flexbox: container

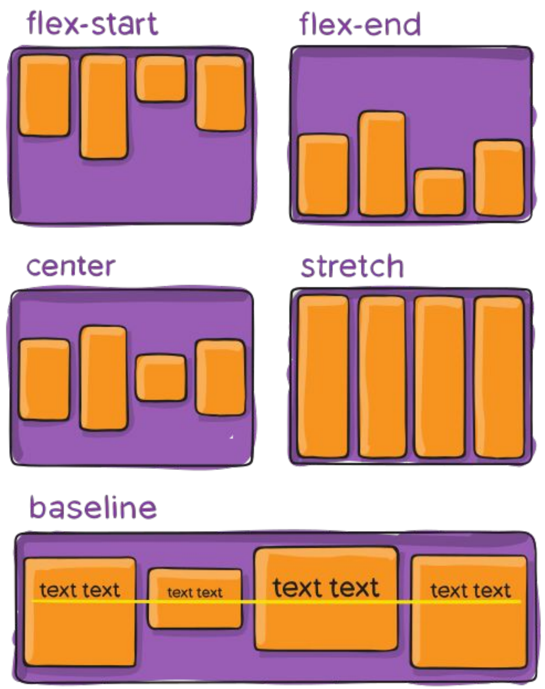
flex-flow: shorthand for `flex-direction` and `flex-wrap`

justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly | start | end | left | right ... + safe | unsafe



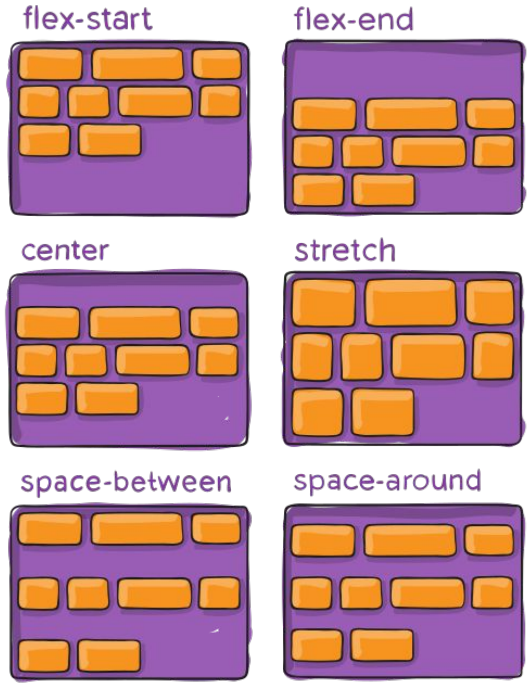
Flexbox: container

align-items: stretch | flex-start | flex-end | center | baseline | first baseline | last baseline | start | end | self-start | self-end + ... safe | unsafe



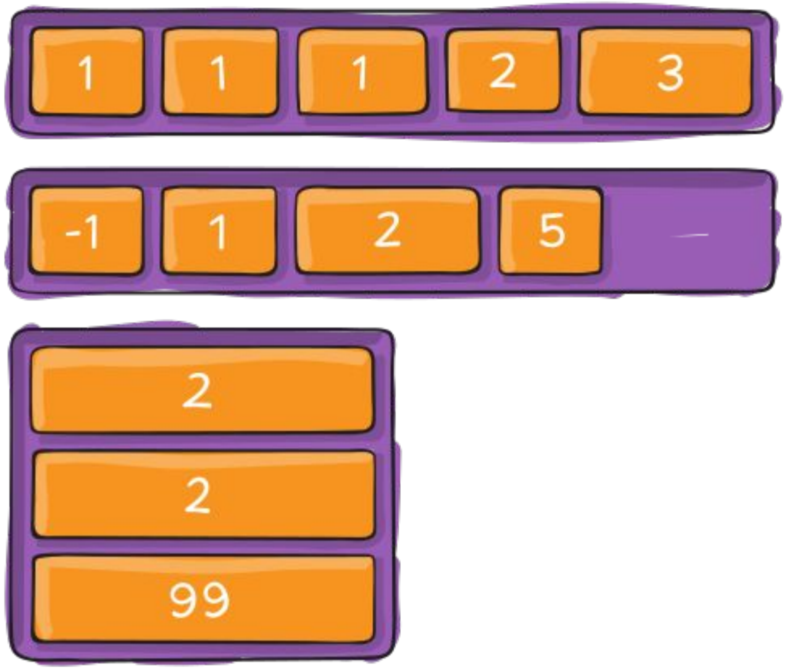
Flexbox: container

align-content: flex-start | flex-end | center | space-between | space-around | space-evenly | stretch | start | end | baseline | first baseline | last baseline + ... safe | unsafe



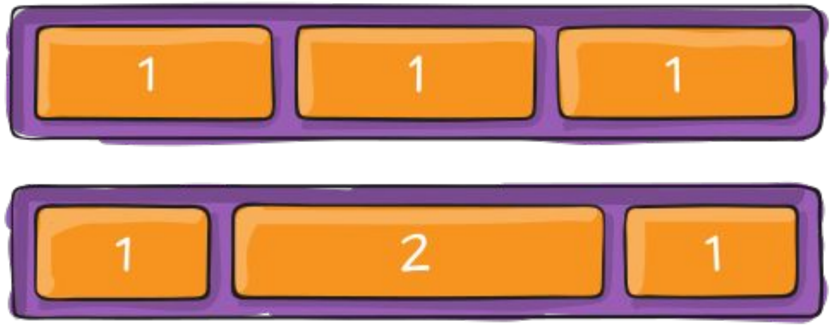
Flexbox: items

order: number (default: 0)



Flexbox: items

flex-grow: number (default: 0)



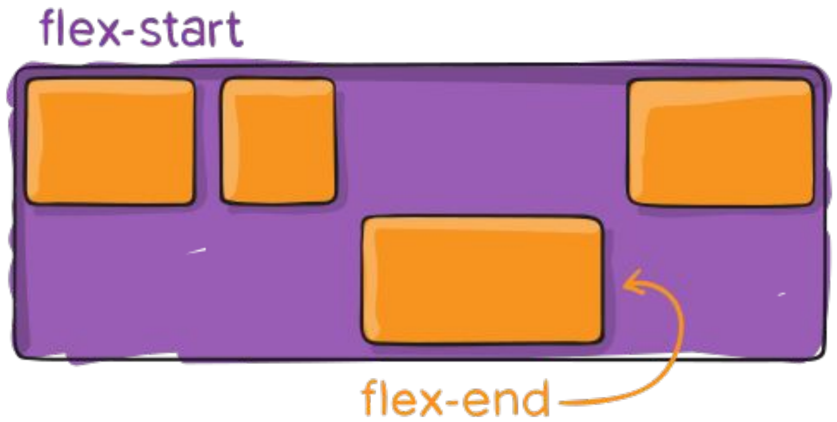
Flexbox: items

align-self: auto | flex-start | flex-end | center | baseline | stretch

flex-shrink: number (default: 1)

flex-basis: number | auto

flex: none | flex-grow flex-shrink | flex-basis



Grid system

Fairly advanced layout system

Not fully supported by all major browsers

https://css-tricks.com/snippets/css/complete-guide-grid/