

Week 6

CSS Flexbox

Box

Box model

content

The picture itself

padding

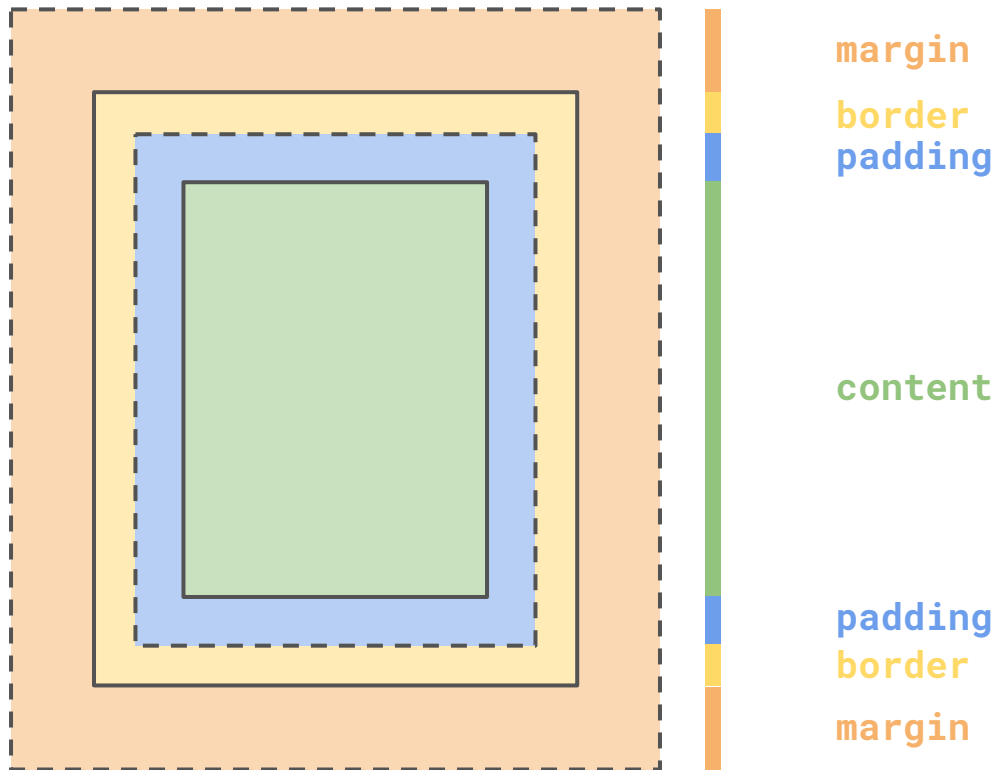
Space between the picture and the picture frame

border

The picture frame

margin

Spacing in between pictures



Grossness

Block element

Block elements want to occupy the entire "line" by itself.

Imagine some folks who want all the car back seats by themselves :(

Examples:

`<div>`, `<h1>`, `<h2>`, `<p>`

Inline element

Inline elements are fine with squeezing with others in the same "line".

Imagine you're on a rideshare but has to share the back seats with 5 other people

Examples:

``, ``, ``, ``

Block element

Can specify width & height

Can specify paddings

Can specify borders

Can specify margins

Inline element

Usually cannot specify width or height :(

Can specify paddings too :o (*but limited*)

Can specify borders :o (*but limited*)

Can specify margins too :o (*but limited*)

* `` is a **replaced inline** element (and you can specify width/height)

Web Design DeCal Spring 2019

House Rules

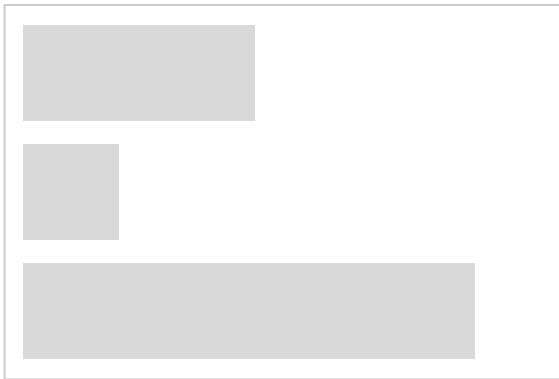
block can contain **inline**

block can contain **block**

inline can contain **inline**

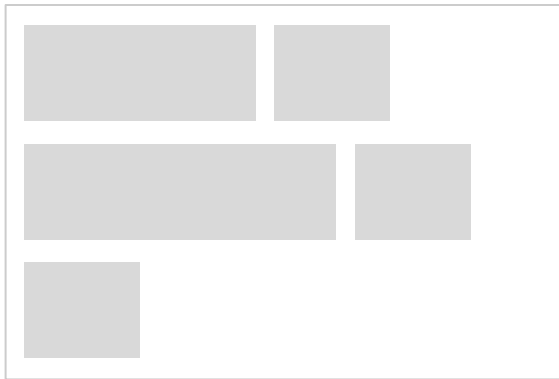
What's missing?

CSS display



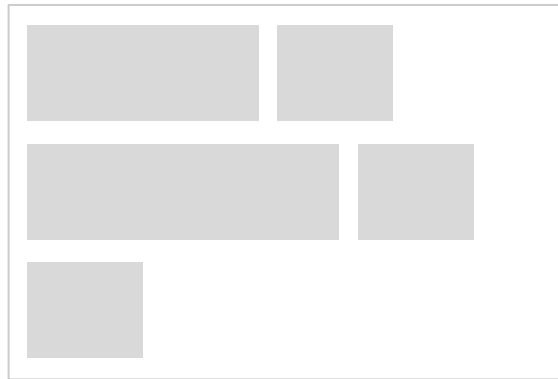
display: block;

Block element



display: inline-block;

*We want inline, but also
what block elements can do*



display: inline;

Inline element

* inline-block is considered as a "legacy" feature

Web Design DeCal Spring 2019

A few ways you can **position** an element

static (default)

Position the element according to the normal flow.

relative

Position the element according to the normal flow, and then **offset relative to itself**.

The offset doesn't affect the surrounding elements.

absolute

The element is **pulled out of the normal flow**,
and positioned **relative to its closest positioned ancestor**.

fixed

The element is **pulled out of the normal flow**,
and positioned **relative to the window/viewport**.

To center things in a **block/inline** layout...

Horizontally center **block** elements

Horizontally center **inline** elements

Vertically center **block** elements

Vertically center **inline** elements

Flexbox

Our displays

```
<div id="okay-box">  
  <p>I am an okay box</p>  
</div>
```

```
<div id="cool-box">  
  <p>I am a cool box</p>  
</div>
```

```
<div id="best-box">  
  <p>I am the best box</p>  
</div>
```

```
#okay-box {  
  display: block;  
}  
  
#cool-box {  
  display: inline;  
}  
  
#best-box {  
  display: flex;  
}
```

Our displays

```
<div id="okay-box">
  <p>I am an okay box</p>
</div>
```

```
<div id="cool-box">
  <p>I am a cool box</p>
</div>
```

```
<div id="best-box">
  <p>I am the best box</p>
</div>
```

```
#okay-box {
  display: block;
}
```

```
#cool-box {
  display: inline;
}
```

```
#best-box {
  display: flex;
}
```

The ~~best~~ new display

```
<div id="okay-box">
  <p>I am an okay box</p>
</div>
```

```
<div id="cool-box">
  <p>I am a cool box</p>
</div>
```

```
<div id="best-box">
  <p>I am the best box</p>
</div>
```

```
#okay-box {
  display: block;
}
```

```
#cool-box {
  display: inline;
}
```

```
#best-box {
  display: flex;
}
```

Fleshing out by flexing

```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

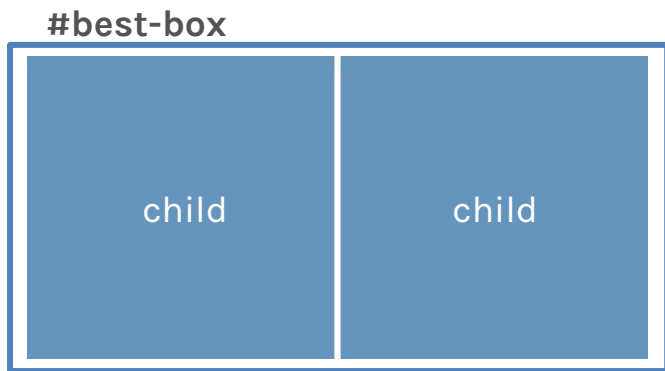
```
#best-box {
  display: flex;
}
```

Flex direction: row

```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: row;
}
```


Flex direction: row



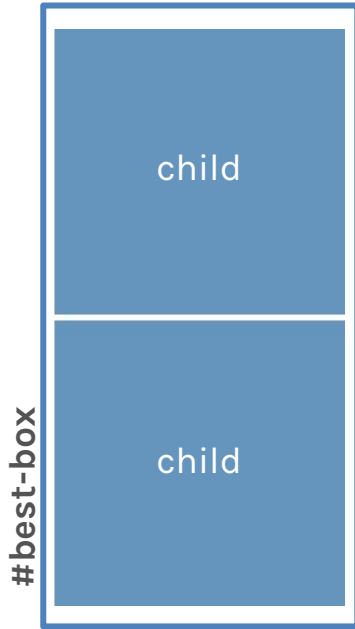
```
#best-box {  
  display: flex;  
  flex-direction: row;  
}
```

Flex direction: column

```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: column;
}
```

Flex direction: column



```
#best-box {  
  display: flex;  
  flex-direction: column;  
}
```

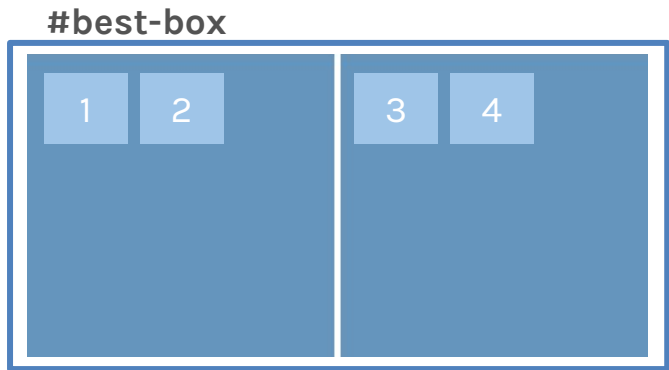
Flex direction: nested

```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: row;
}

.child {
  display: flex;
  flex-direction: row;
}
```

Flex possibilities: nested



```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
}
```

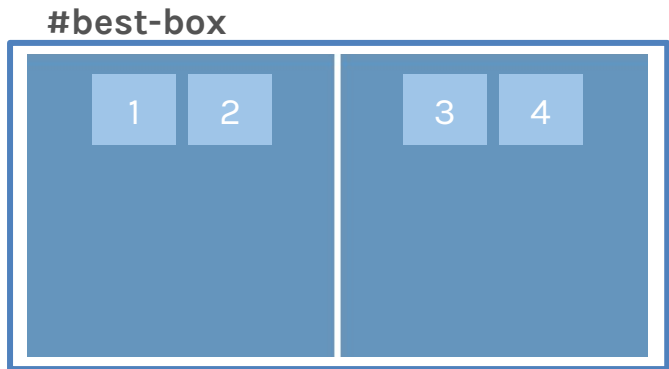
Flex possibilities: centering

```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: row;
}

.child {
  display: flex;
  flex-direction: row;
  justify-content: center;
}
```

Flex possibilities: centering



```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
}
```

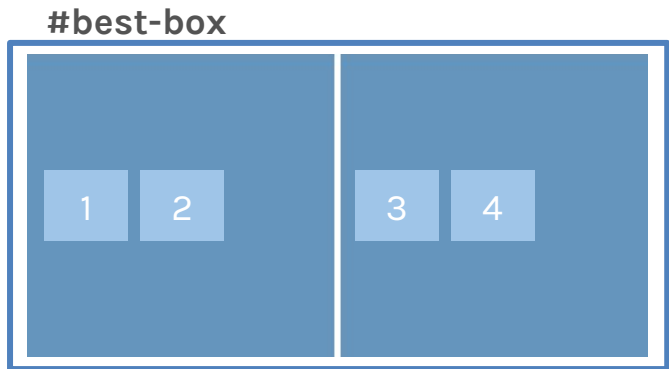
Flex possibilities: centering

```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: row;
}

.child {
  display: flex;
  flex-direction: row;
  align-items: center;
}
```


Flex possibilities: centering



```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  align-items: center;  
}
```

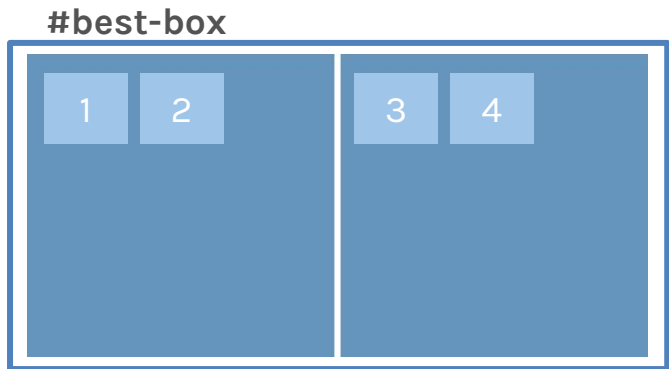
Flex possibilities: flex-start

```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: row;
}

.child {
  display: flex;
  flex-direction: row;
  justify-content: flex-start;
  align-items: flex-start;
}
```

Flex possibilities: flex-start



```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-start;  
  align-items: flex-start;  
}
```

Flex possibilities: flex-end

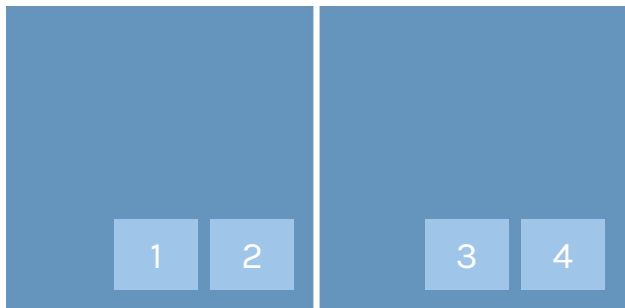
```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: row;
}

.child {
  display: flex;
  flex-direction: row;
  justify-content: flex-end;
  align-items: flex-end;
}
```

Flex possibilities: flex-end

#best-box



```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-end;  
  align-items: flex-end;  
}
```

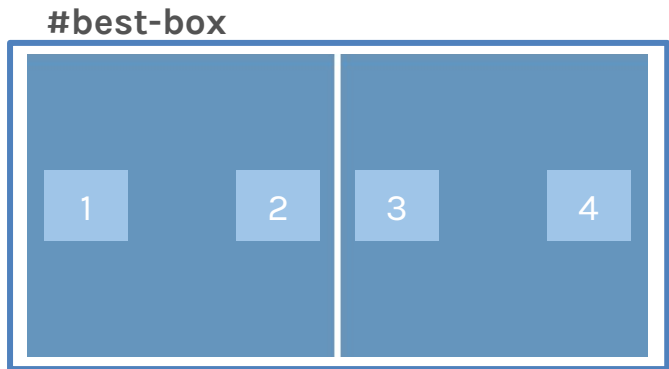
Flex possibilities: space-between

```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: row;
}

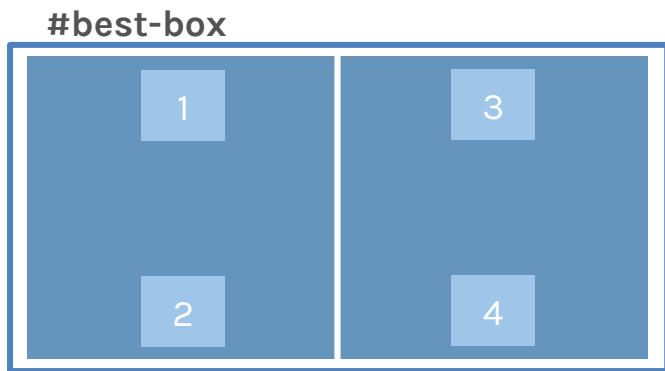
.child {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  align-items: space-between;
}
```

Flex possibilities: space-between



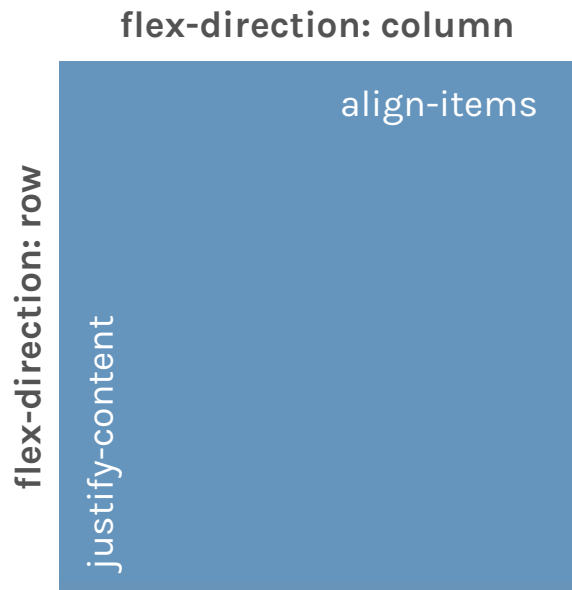
```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  justify-content: space-between;  
}
```

Flex possibilities: space-between



```
#best-box {  
  display: flex;  
  flex-direction: row;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  align-items: space-between;  
}
```


Row vs. column... rotate!



```
#best-box {  
  display: flex;  
  flex-direction: column;  
}  
  
.child {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
  align-items: center;  
}
```

Flexbox Practice

<https://tinyurl.com/wddflexy>

What you can do with **flexbox**

<https://tinyurl.com/wddphone>

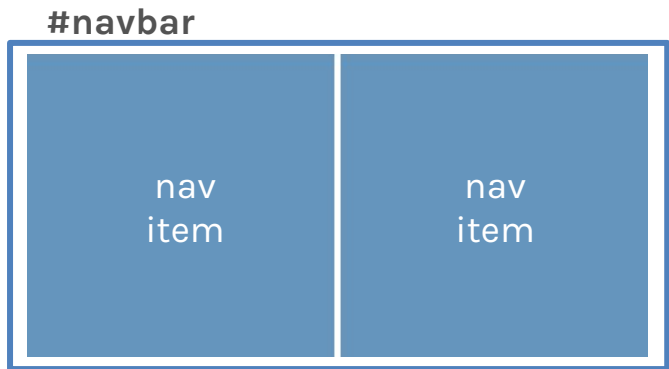
A realistic example: nav

```
<nav id="navbar">
  <div class="nav-item">
    Logo
  </div>
  <div class="nav-item">
    About
  </div>
</nav>
```

```
#navbar {
  display: flex;
  flex-direction: row;
}

#nav-item {
  display: flex;
  justify-content: center;
}
```

A realistic example: nav



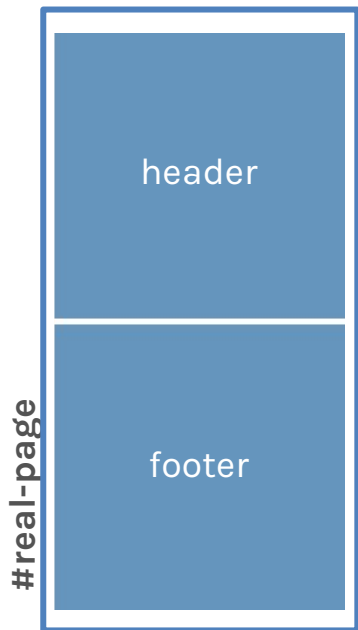
```
#navbar {  
  display: flex;  
  flex-direction: row;  
}  
  
#nav-item {  
  display: flex;  
  justify-content: center;  
}
```

A realistic example: layout

```
<div id="real-page">
  <header id="header">
    <p>Title</p>
  </header>
  <footer id="footer">
    <p>Contact</p>
    <p>Support</p>
  </footer>
</div>
```

```
#real-page {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
}
```

A realistic example: layout



```
#real-page {  
  display: flex;  
  flex-direction: column;  
  justify-content: space-between;  
}
```

Flex possibilities: miscellaneous

```
<div id="best-box">
  <div class="child">
    <p>1</p>
    <p>2</p>
  </div>
  <div class="child">
    <p>3</p>
    <p>4</p>
  </div>
</div>
```

```
#best-box {
  display: flex;
  flex-direction: column;
  justify-content: space-around;
  align-items: space-evenly;
  align-self: center;
  flex-wrap: wrap;
  flex-grow: 1;
  flex-shrink: 2;
  flex-basis: auto;
}
```

Flexbox Froggy

<http://flexboxfroggy.com/>

FLEXBOX FROGGY



◀ Level 5 of 24 ▶

Now use **align-items** to help the frogs get to the bottom of the pond. This CSS property aligns items vertically and accepts the following values:

- **flex-start**: Items align to the top of the container.
- **flex-end**: Items align to the bottom of the container.
- **center**: Items align at the vertical center of the container.
- **baseline**: Items display at the baseline of the container.
- **stretch**: Items are stretched to fit the container.

```
1 #pond {
2   display: flex;
3   align-items: flex-end
4 }
5
6
7
8
9
10
```

Submit



Additional links & guides

- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://medium.freecodecamp.org/an-animated-guide-to-flexbox-d280cf6afc35>
- <https://giphy.com/gifs/how-flexbox-zfv3Kog26SL04>
- <https://scotch.io/tutorials/a-visual-guide-to-css3-flexbox-properties>



Attendance