

**FEWD WEEK 4 • CLASS 8:**

# Key Concepts

<https://slides.com/jennifermeade/fewd-4-8/live>



# OBJECTIVES

- Review key concepts in HTML & CSS
- Solve common UI challenges
- Understand where to go to find answers and get help

# CHALLENGES

- Align elements horizontally
- Align elements vertically
- Make images responsive
- Understanding flexbox

# **HORIZONTAL ALIGNMENT**

## **TEXT-ALIGN**

Align inline elements inside of a  
block element

# USING TEXT-ALIGN

*PARAGRAPHS  
ARE BLOCK*

*TEXT NODES  
ARE INLINE*

```
p {  
  text-align: center;  
}
```

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.

```
p {  
  text-align: left;  
}
```

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.

```
p {  
  text-align: right;  
}
```

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.

# ANY INLINE ELEMENT WORKS

```
div {  
  text-align: center;  
}
```

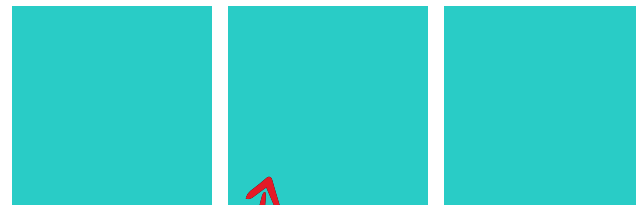
**DIVS ARE  
BLOCK**



**IMAGES ARE  
INLINE**

# INLINE-BLOCK WORKS TOO

```
div.outside {  
  text-align: center;  
}  
  
div.inner {  
  height: 100px;  
  width: 100px;  
  background: #29ccc6;  
  display: inline-block;  
}
```



*DIVS SET AS  
INLINE-BLOCK*



## MARGIN: AUTO

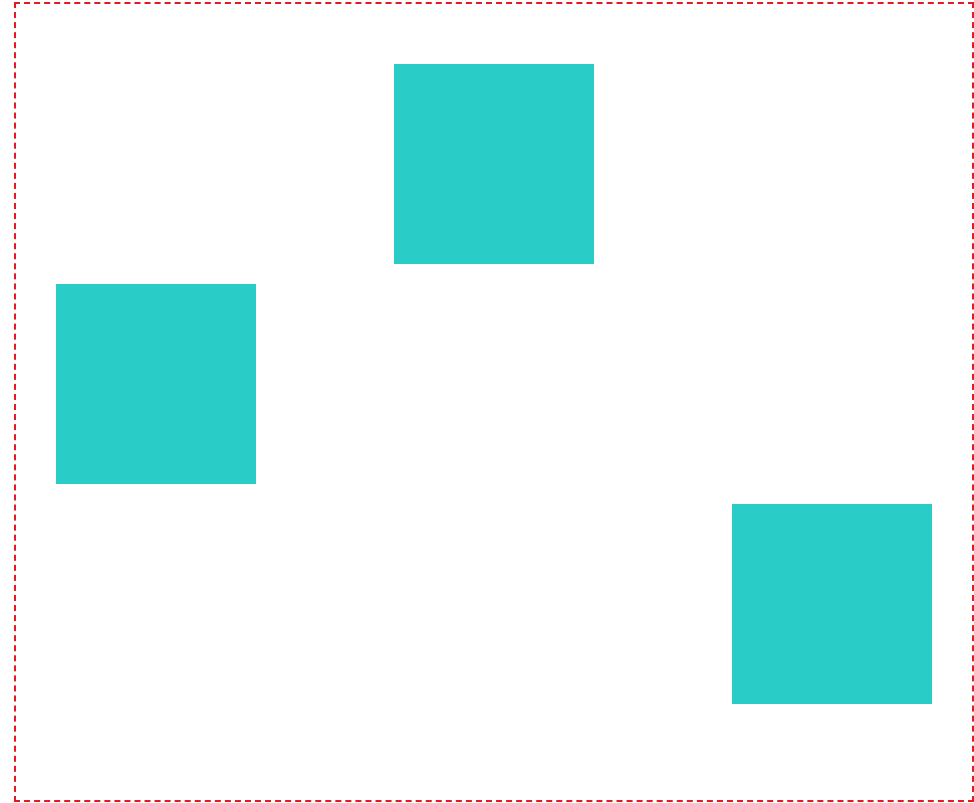
Align a block element inside of another block element



This technique requires the outer block element to be larger than the inner block element.

# USING MARGIN: AUTO

```
div.inner {  
  height: 100px;  
  width: 100px;  
  margin-top: 10px;  
}  
  
div.inner:nth-child(1) {  
  margin-right: auto;  
  margin-left: auto;  
}  
  
div.inner:nth-child(3) {  
  margin-left: auto;  
}
```



# FLOAT

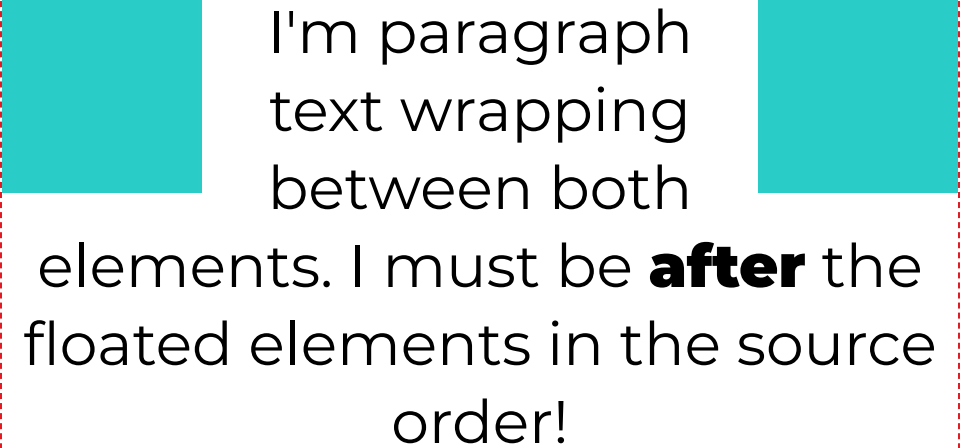
Align elements inside of a block element side-by-side



Float can only align elements to the left or right. There is no center option.

# USING FLOAT

```
div.inner {  
  height: 100px;  
  width: 100px;  
  background: #29ccc6;  
}  
  
div.inner:nth-child(1) {  
  float: left;  
}  
  
div.inner:nth-child(2) {  
  float: right;  
}
```



I'm paragraph  
text wrapping  
between both  
elements. I must be **after** the  
floated elements in the source  
order!

**FLEX**

Align elements inside of  
another element



Not supported in Internet Explorer.

**93.03%** support in the US

# USING FLEX

```
div.inner {  
  height: 100px;  
  width: 100px;  
  background: #29ccc6;  
  margin: 10px;  
}  
  
div.outside {  
  display: flex;  
  justify-content: center;  
}
```

**CENTER ALONG THE  
MAIN AXIS WITH  
JUSTIFY-CONTENT**



**FLEX DIRECTION IS  
ROW BY DEFAULT**

# **VERTICAL ALIGNMENT**

## **LINE-HEIGHT**

Center a single line of text  
vertically in its container.



# USING LINE-HEIGHT

```
p {  
  height: 500px;  
  line-height: 500px;  
  text-align: center;  
}
```

This is a line of text.

## **FLEX**

Align child elements vertically  
inside a flex container.



Not supported in Internet Explorer.

**93.03%** support in the US

# USING FLEX

```
div.outside {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}  
  
div.inner:nth-child(1) {  
  align-self: flex-start;  
}  
  
div.inner:nth-child(3) {  
  align-self: flex-end;  
}
```



## **POSITION**

Align child elements within a positioned ancestor.

# POSITION ONLY

```
section {  
  position: relative;  
  height: 400px;  
}  
  
div {  
  position: absolute;  
  bottom: 0;  
  right: 0;  
  background: #29ccc6;  
}
```



I'm aligned  
bottom-right

## POSITION + TRANSFORM

Align an element in the center of its positioned ancestor.

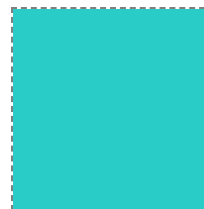


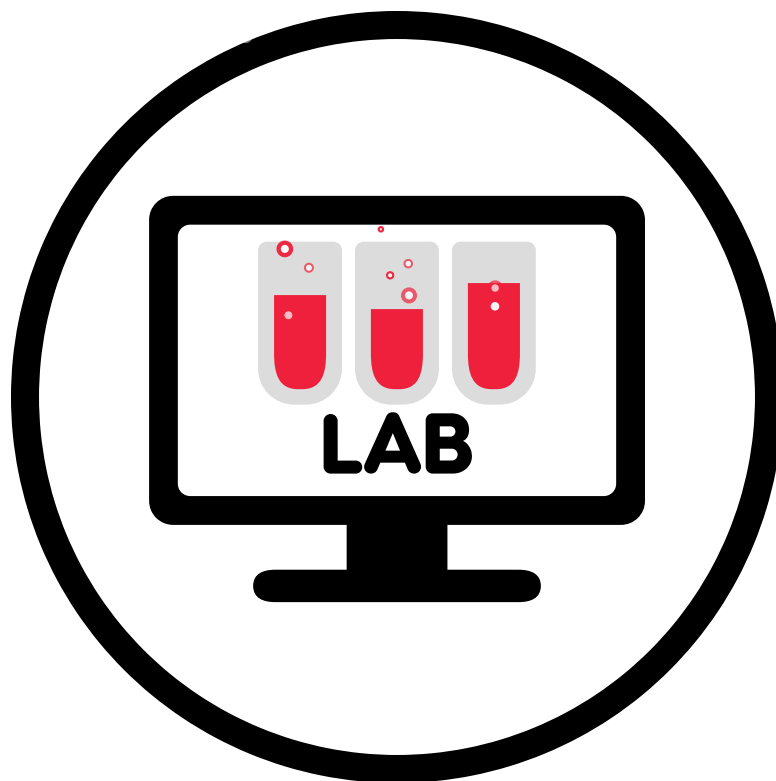
Transform is supported in Internet Explorer 9+.

**94.14%** support in the US

# TRANSFORM HACK

```
section {  
  position: relative;  
  height: 400px;  
}  
  
div {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  background: #29ccc6;  
  transform: translate(-50%,-50%);  
}
```





**CENTER STUFF**

<https://codepen.io/jme11/pen/pxXqGE>



# **RESPONSIVE IMAGES**

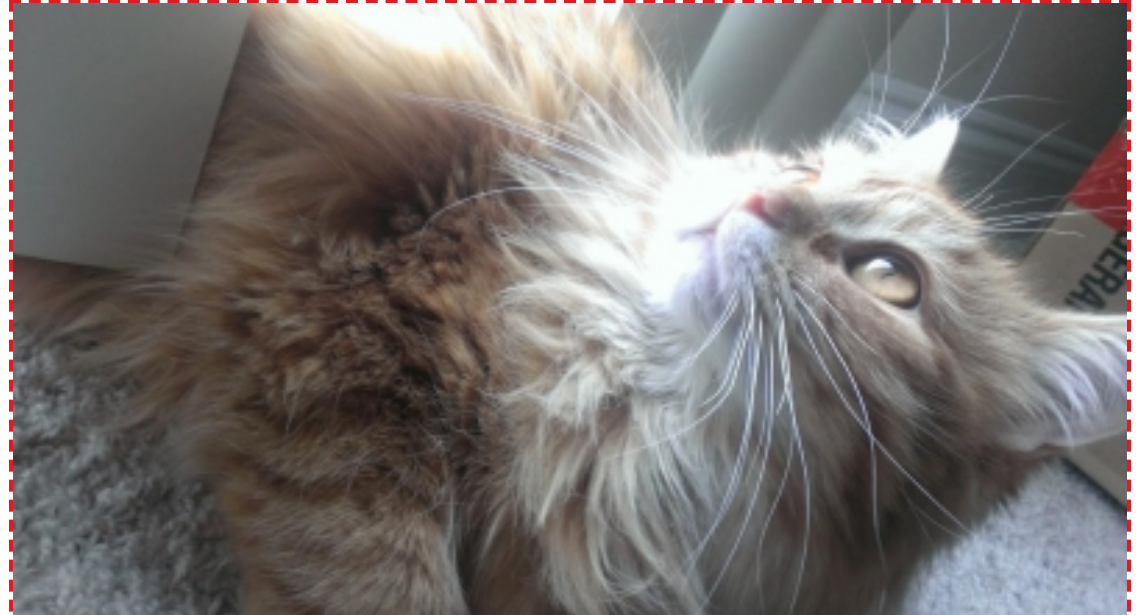
# IMAGES IN RESPONSIVE PAGES

- Remember that your images have an intrinsic height and width (the size of the image)
- By default, if you set either the height **or** width of an image, it will resize the other value proportionally
- Large images can slow your download. Optimize them and use srcset or picture, if needed.

# SETTING THE IMAGE WIDTH



This image will not fill its parent or scale to fit the viewport.



Set the width to 100% so the image fills its parent and scales with the viewport.

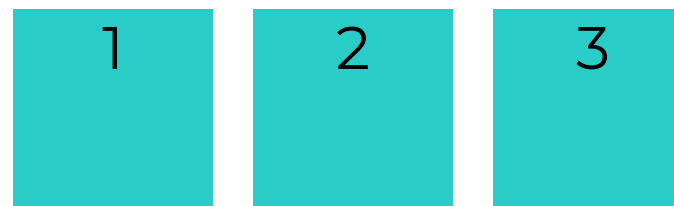
# **FLEX REVISITED**

# LEARN FLEXBOX

- Flexbox should be your goto unless you really need to support very old browsers (validate with analytics)
- Dig into all of its features and properties — it's more powerful than you know

# FLEX

```
div.outside {  
  display: flex;  
}  
  
div.inner {  
  height: 100px;  
  width: 100px;  
  background: #29ccc6;  
}
```



# FLEX-GROW

```
div.outside {  
  display: flex;  
}  
  
div.inner {  
  height: 100px;  
  background: #29ccc6;  
  flex-grow: 1;  
}
```



# FLEX-GROW

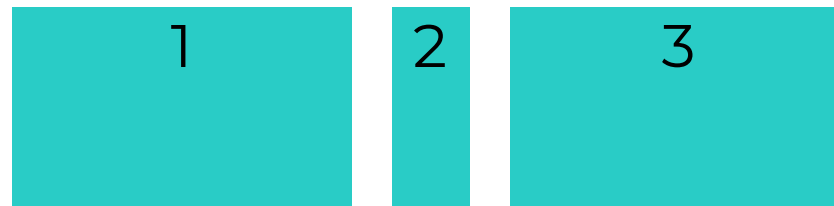
```
div.outside {  
  display: flex;  
}  
  
div.inner {  
  height: 100px;  
  background: #29ccc6;  
  flex-grow: 1;  
}  
  
div.inner:nth-child(2) {  
  flex-grow: 2;  
}
```





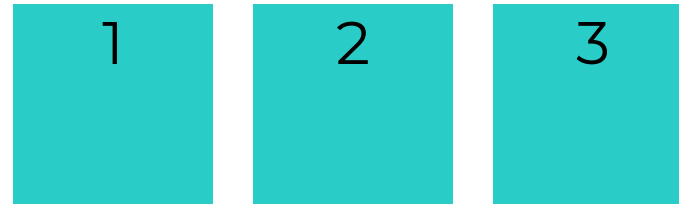
# FLEX-SHRINK

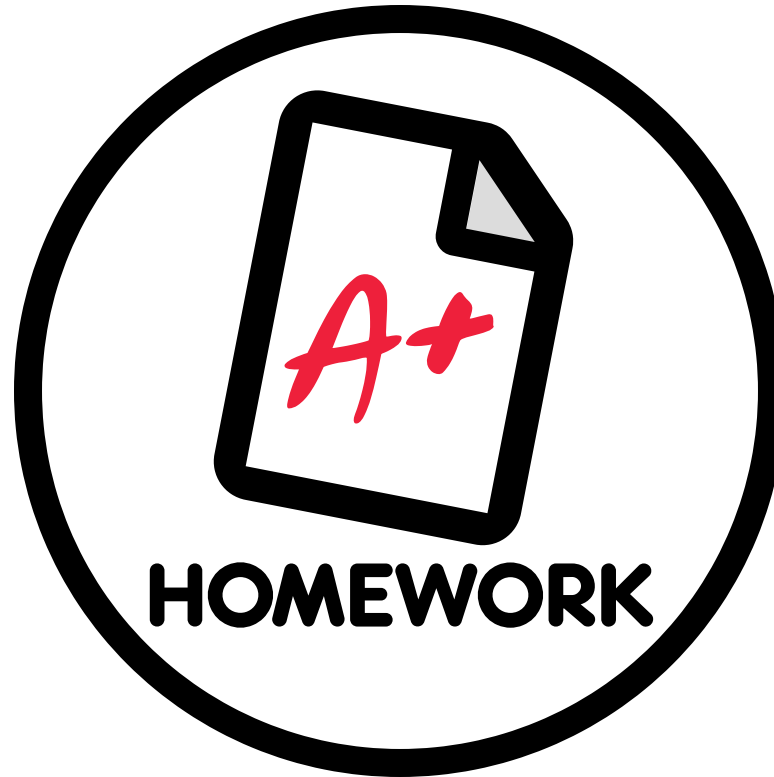
```
div.outside {  
  display: flex;  
}  
  
div.inner {  
  height: 100px;  
  background: #29ccc6;  
  width: 300px;  
}  
  
div.inner:nth-child(2) {  
  flex-shrink: 2;  
}
```



# FLEX-BASIS

```
div.inner {  
  height: 100px;  
  background: #29ccc6;  
  flex-basis: 100px;  
}  
  
div.outside {  
  display: flex;  
}
```





## **CLASS 8 HOMEWORK**

<https://github.com/jmeade11/FEWD/tree/master/Final%20Project>

# **HOMEWORK FOR NEXT CLASS**

- Let's make sure you have a final project proposal ready for Monday.
- Make sure you have a proposal and wireframes submitted. You can send them as a zip file to me over Slack.



Here's your chance to get creative. Your final project proposal is due on Monday. You'll need to have a clear concept and a set of wireframes (pencil drawing are fine).

# **EXIT SURVEY**

<https://goo.gl/EB4XFw>

**GO BUILD  
AWESOME THINGS!**