

# **FEWD WEEK 2 • CLASS 4:** CSS Core Concepts

<https://slides.com/jennifermeade/fewd-2-4/live>

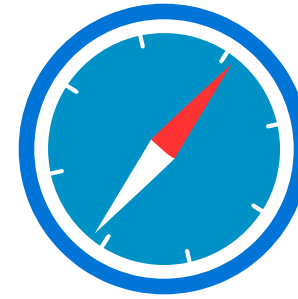


# OBJECTIVES

- Learn to use DevTools to debug your code
- Use the display property to change how elements are rendered
- Understand the 4 principle techniques used for layout in CSS

# **DEBUGGING WITH DEVTOOLS**

# ACCESSING THE DEVTOOLS



**Right-click** any element and choose **Inspect Element** (or Inspect in Chrome)



MAC shortcut keys:  Command +  Option + I

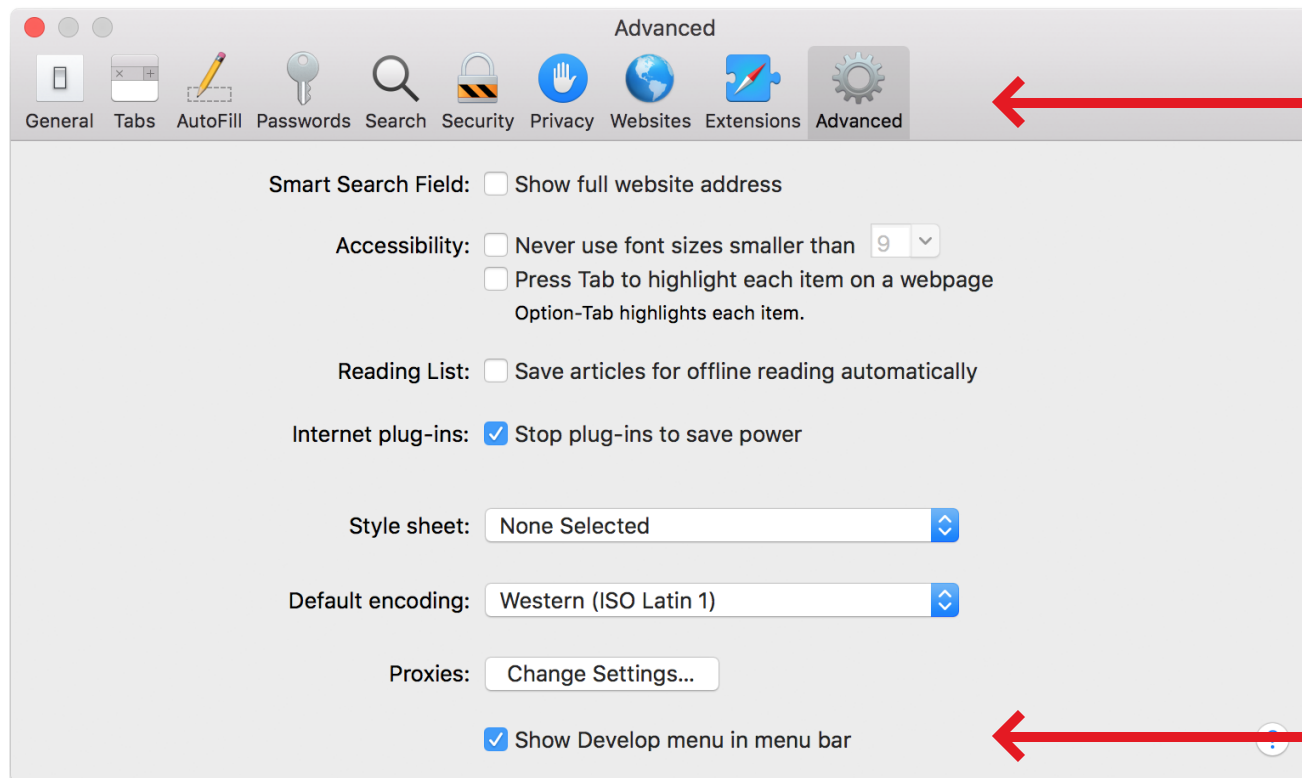
WIN shortcut keys:  CTRL +  SHIFT + I

IE/Edge Only:  F12

# ACTIVATE THE DEV MENU

## SAFARI ONLY

- Go to Safari > Preferences

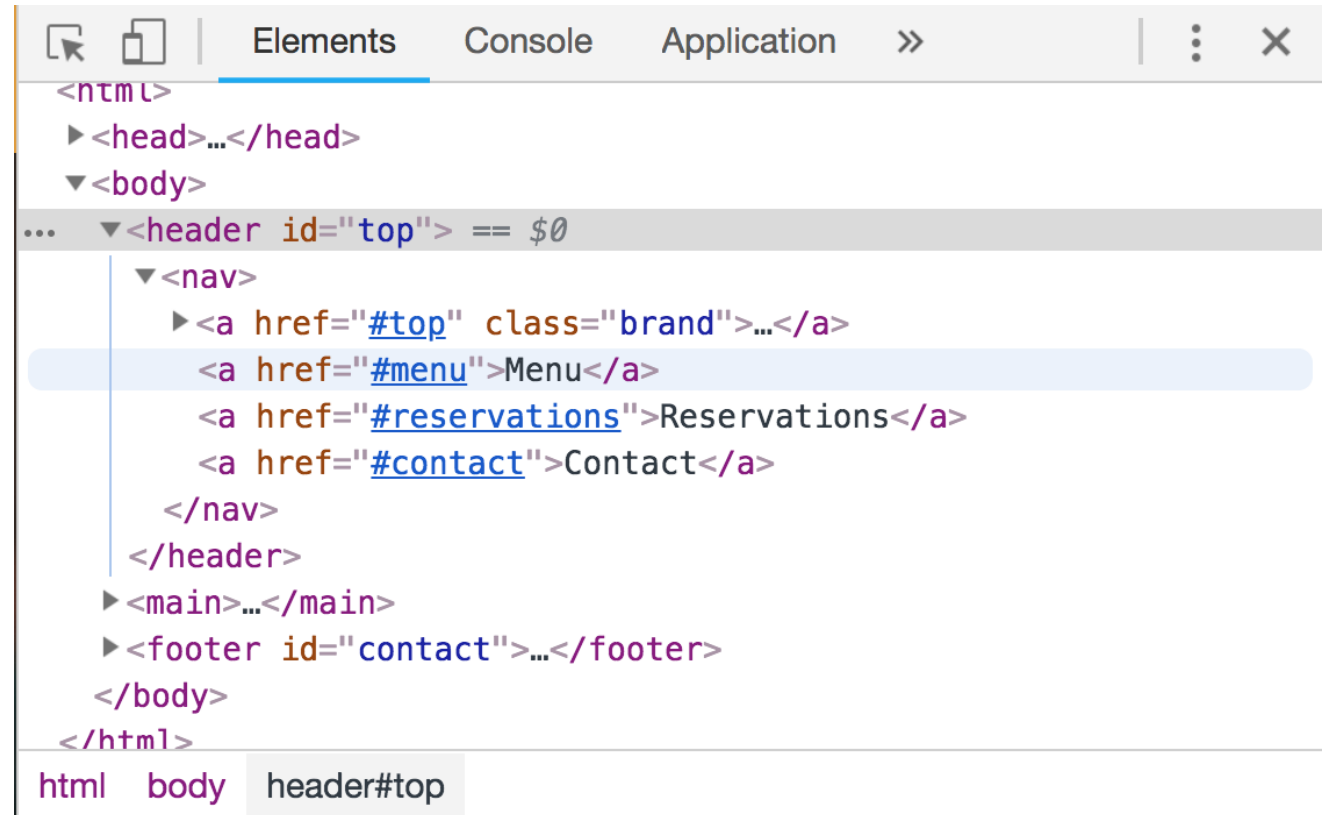


Go to the  
**Advanced** tab

Check show  
menu option

# INSPECTING HTML

- Click the top left (pointer) icon then select an element on the page to quickly find it in the code.
- Rollover an element in DevTools to highlight it on the page.
- Click the twisty to expand or collapse element contents
- Click the mobile devices icon to switch to mobile viewing



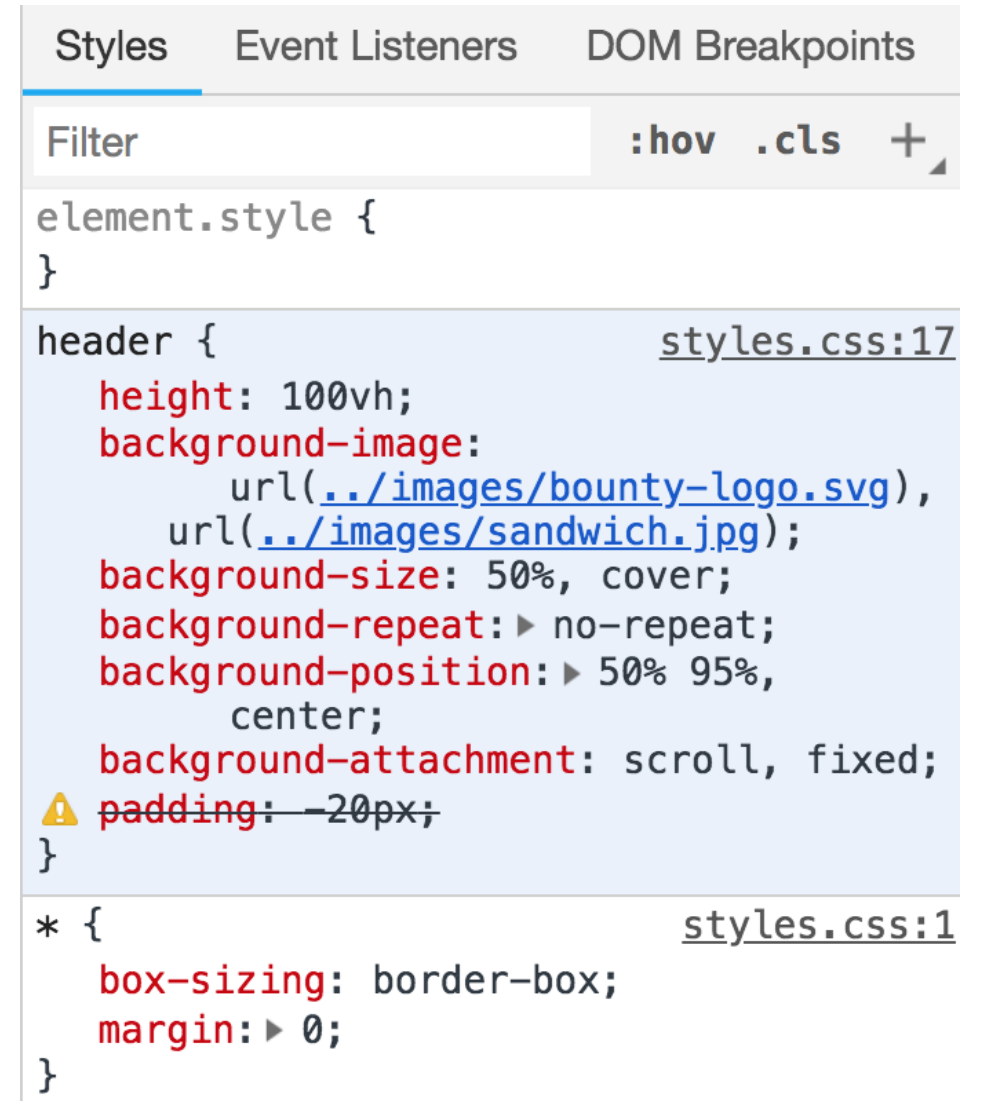
The screenshot shows the Chrome DevTools 'Elements' panel. The HTML tree is expanded to the `<nav>` element, which is a child of `<header id='top'>`. The `<a href='#menu'>Menu</a>` element is selected and highlighted in blue. The breadcrumb at the bottom indicates the path: `html > body > header#top`. The top of the panel shows tabs for 'Elements', 'Console', and 'Application', along with icons for zooming and a mobile device view toggle.

```
<html>
  ><head>...</head>
  ><body>
    ... ><header id="top"> == $0
      ><nav>
        ><a href="#top" class="brand">...</a>
        ><a href="#menu">Menu</a>
        ><a href="#reservations">Reservations</a>
        ><a href="#contact">Contact</a>
      </nav>
    </header>
    ><main>...</main>
    ><footer id="contact">...</footer>
  </body>
</html>
```

html body header#top

# INSPECTING CSS

- Styles for the currently selected element
- Strikethrough means not applied (either overridden or invalid)
- Warning icon is an invalid style
- The location is shown as filename:line-number (click it to see it in the Sources tab)
- Styles on top have more specificity
- Italicized styles are the browser defaults.





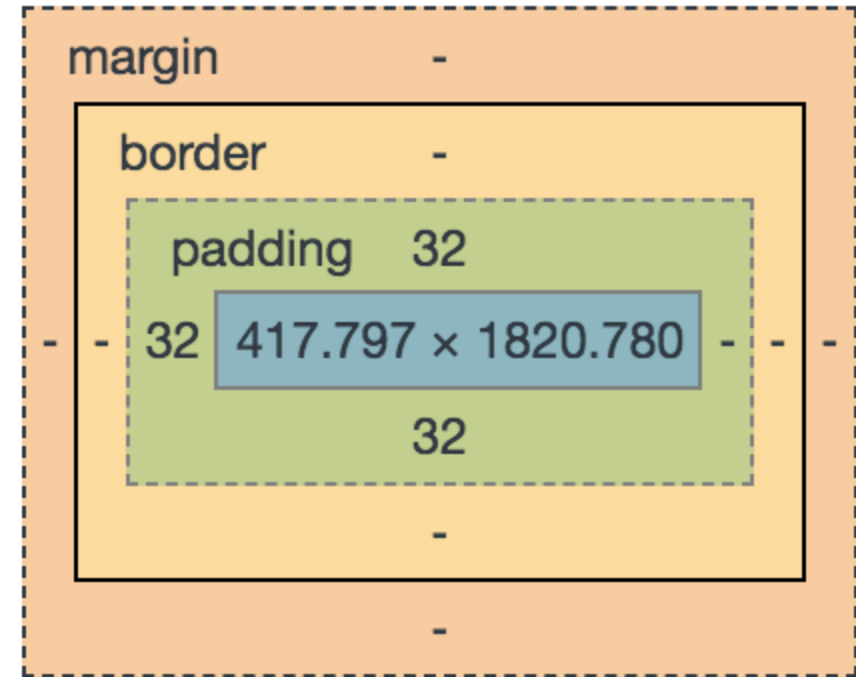
# COMPUTED PROPERTIES

Filter	<input type="checkbox"/> Show all
▼ background-attachment	scroll, fixed
scroll, fixed header	<a href="#">styles.css:17</a>
▶ background-image	url( <a href="#">file:///Users/ja...</a>
▶ background-position-x	50%, 50%
▶ background-position-y	95%, 50%
▶ background-repeat-x	
▶ background-repeat-y	
▶ background-size	50%, cover
▶ box-sizing	border-box
▶ display	block
▶ font-family	"Slabo 13px", serif
▶ font-size	20px
...	...



# BOX MODEL

- Shows the computed values (always in pixels) for the box model of the currently selected element
- Rollover the content, padding, border or margin box to highlight it on the element in the page.





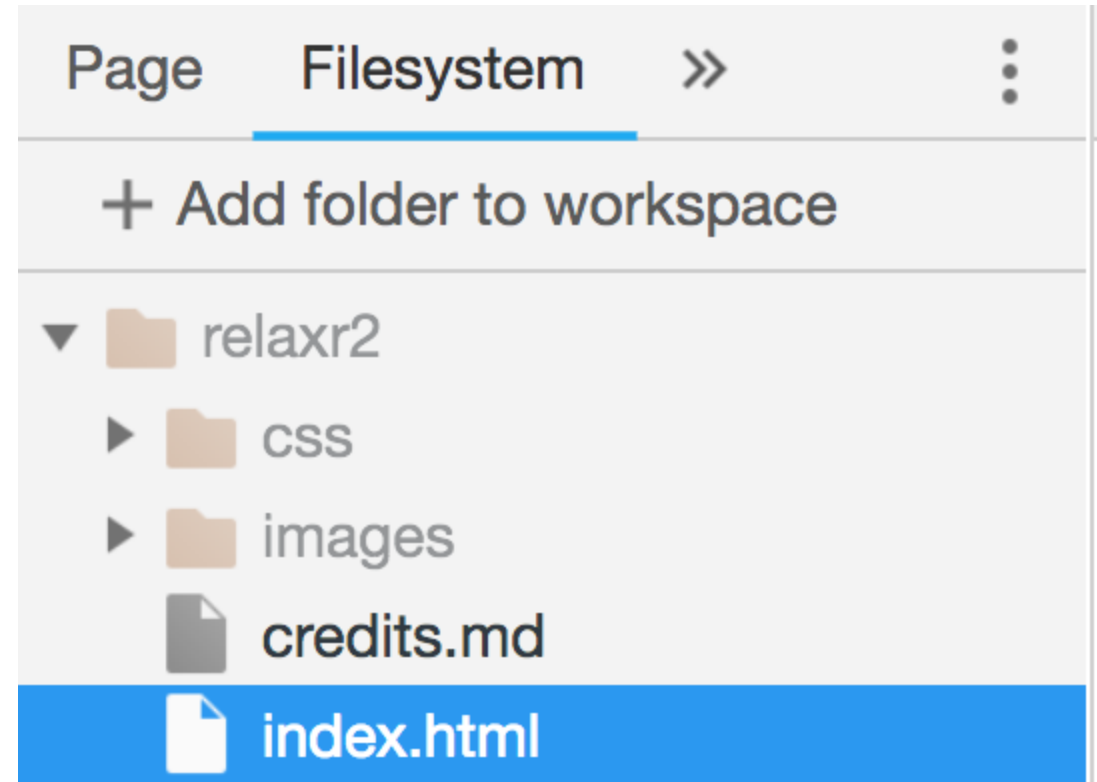
# **INSPECTING HTML AND CSS IN DEV TOOLS**

# EXPERIMENTING

- Dev Tools are great for experimenting.
- Changes you make are non-destructive.
- You can add, remove or modify HTML, attributes, style properties or values.
- You can see and modify styles associated with element states (click the :hov at the top of the Styles pane)

# EDITING IN THE BROWSER

- You can also edit and save files directly in the browser in the sources tab.
- Choose Sources, Filesystem, Add folder to workspace.
- Click Allow when prompted
- Click Page and edit and save the files.



# CSS DISPLAY PROPERTY

# THE CSS DISPLAY PROPERTY

```
div {  
  display: none; // Don't display this element!  
}  
  
h1 {  
  display: inline; // Treat this as an inline element!  
}
```

- The display property is **magic**!
- It allows us to change how an element is rendered by the browser.
- It also allows us to not render the element at all!

# BLOCK, INLINE OR BOTH!

```
img {  
  display: block; // Treat this as a block element!  
}  
  
a {  
  display: inline-block; // Treat this as both!  
}
```

- The `inline-block` value gives us the ability to treat elements as inline, but they can also have **height** and **width** applied!



# BLOCK, INLINE OR BOTH!

```
img {  
  display: block; // Treat this as a block element!  
}  
  
a {  
  display: inline-block; // Treat this as both!  
}
```

- The `inline-block` value gives us the ability to render elements inline, but they can also have **height** and **width** applied!

# TABLES & LIST ITEMS

- The display property also accepts several values that allow elements to be rendered as list-items and table-related elements
- To see these in action checkout the [W3schools](#) demo.

# GET TO KNOW DISPLAY

- The display property gets **a lot** of mileage in CSS and JS
- It is fundamental to modern layout techniques in CSS as we'll see in a moment.

# CSS LAYOUTS

# LAYOUTS IN CSS

- Floats
- Columns
- Grid
- Flexbox

# FLOATS

# WHAT IS FLOAT?

Float places an element on the **left** or **right** of its container and allows other elements to wrap around it.

```
/* Float accepts left, right or none
   The none value is the default */

.my-floated-element {
  float: right;
}
```



# CLEARING THE FLOAT

The **clear** property prevents elements from sitting next to the floated elements that precede it — forcing the cleared element below the floated element.

```
/* Clear accepts left, right or both */  
  
.my-cleared-element {  
    clear: both;  
}
```

# FLOAT EXAMPLES

HTML

CSS

Result

FORK ON  
CODEPEN


LIVE

```
1 div {  
2   height: 100px;  
3   width: 100px;  
4   background-color: rgba(255,0,0,.25);  
5   border: 5px solid #e41a23;  
6   float: right;  
7 }
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Eius id ipsa vel sint exercitationem ipsum veniam incidunt asperiores quis. Distinctio aliquid qui repellat incidunt dolorem officiis at numquam deleniti quo?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ullam officiis possimus debitis? Iure in animi id, optio illum nihil, hic facilis similique nam nesciunt nostrum nisi ullam est sint itaque!

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Reprehenderit, nulla. Quis at cupiditate nostrum distinctio soluta saepe molestias quam voluptate, officia ullam laudantium reprehenderit quia aspernatur et quaerat nemo tenetur.



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

# FLOAT PROS & CONS

## PROS

- Fairly simple to use
- Doesn't require additional outer elements
- Supported in all browsers

## CONS

- Content cannot stretch to fill available height
- Elements cannot be centered
- Containers collapse

# WHEN I USE IT...

- Whenever I have to support **every** older browser on the planet.
- When I just need to quickly push one element to the right (e.g., in Bounty20 when we floated the prices).
- When I want text to flow around an image or other element.

**COLUMNS**

# WHAT IS COLUMN?

When column-count is specified on an outer container the descendants of that container will be rendered into the specified number of columns as evenly as possible

```
/* column-count accepts a number and  
   column-gap a unit of measurement */  
  
.my-column-container {  
  column-count: 2;  
  column-gap: 1em;  
}
```

# COLUMN DESCENDANTS

Using the `break-inside: avoid` property can help to force the browser to keep certain contents together

```
.keep-my-contents-together {  
    break-inside: avoid;  
}
```



# COLUMN EXAMPLE

HTML

CSS

Result

FORK ON  
CODEPEN

1 ▾ <h1>CODEPEN EXAMPLE</h1>

2 ▾ <h3>Use the link below to visit this example</h3>

LIVE

CODEPEN  
EXAMPLE

Use the link below to visit this  
example

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

# COLUMNS PROS & CONS

## PROS

- Fairly simple to use
- Has the ability to control the gap width
- Works back to IE10

## CONS

- Requires an outer container
- Only works in vertical arrangement
- Difficult to control column breaks

# COLUMN SUPPORT TABLE

## CSS3 Multiple column layout - CR

Method of flowing information in multiple columns

Usage	% of all users	
Global	86.66%	+ 9.14% = 95.8%
unprefixed:	86.66%	+ 4.89% = 91.55%
U.S.A.	89.59%	+ 6.5% = 96.09%
unprefixed:	89.59%	+ 3.72% = 93.31%

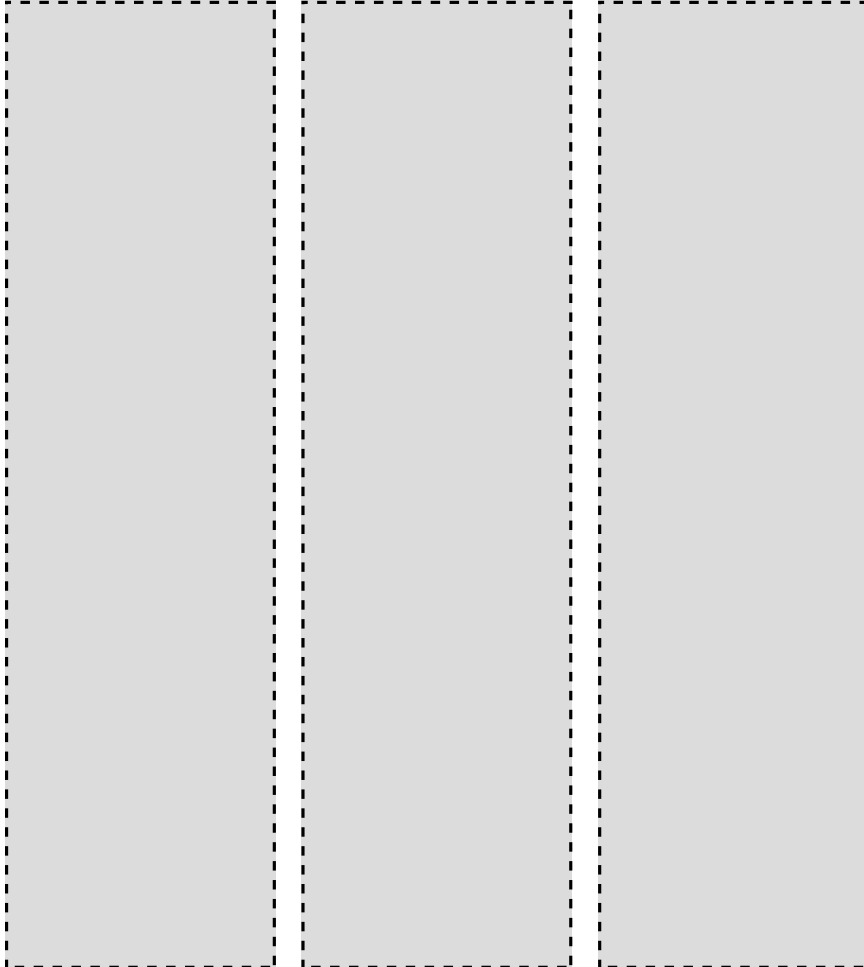
Current aligned	Usage relative	Date relative	Apply filters	Show all	?							
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile*	Chrome for Android	Firefox for Android
				<div>123.1-6.1</div>	10.1	<div>123.2-6.1</div>						
				<div>17-8</div>	11.5-12.1	<div>17-8.4</div>						
6-9		<div>12-51</div>	<div>124-49</div>	<div>19-9.1</div>	<div>1215-36</div>	<div>19-9.3</div>						
10	12-16	<div>152-61</div>	50-68	10-11.1	37-54	10-11.2		<div>122.1-4.4.4</div>	<div>127</div>	12-12.1		
11	17	<div>162</div>	69	12	55	11.4	all	67	<div>1210</div>	<div>146</div>	69	<div>162</div>
	18	<div>163-64</div>	70-72	TP		12						

## **WHEN I USE IT...**

- Whenever I want a Pinterest-style gallery
- If I have just a couple of elements in columns and I have to support older browsers

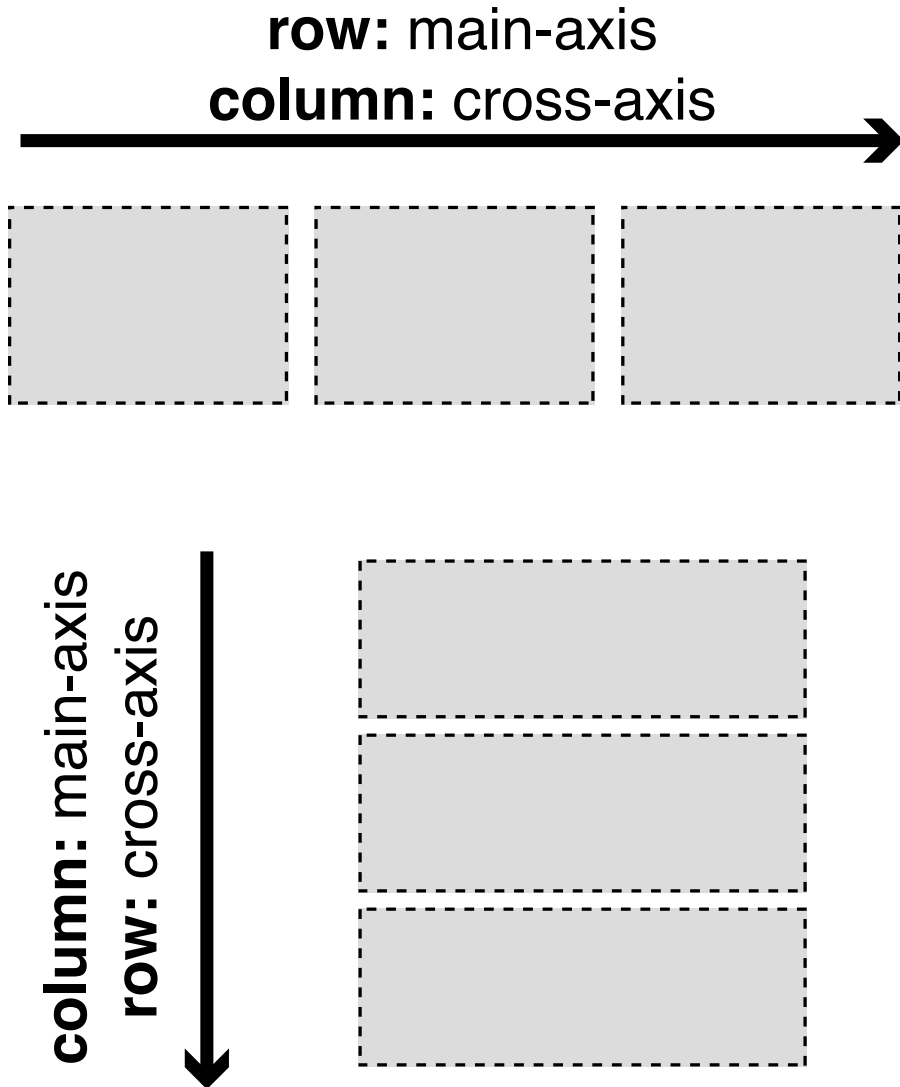
**FLEX BOX**

# WHAT IS FLEXBOX?



Flexbox is a newer standard. It allows us to more easily layout elements in 1 direction.

# FLEX CONTAINER



```
.parent {  
  display: flex;  
  /* row or column */  
  flex-direction: row;  
  /* main axis */  
  justify-content: space-between;  
  /* cross axis */  
  align-items: stretch;  
  /* wrap items or not*/  
  flex-wrap: wrap;  
}
```



# JUSTIFY-CONTENT PROPERTY

**flex-start**



**space-between**



**flex-end**



**space-around**



**center**



**space-evenly**

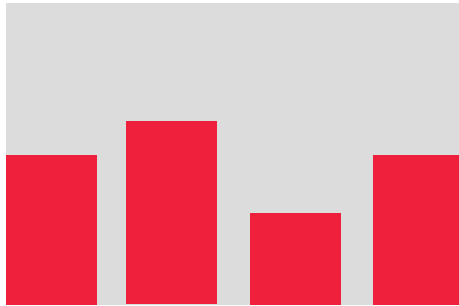


# ALIGN-ITEM PROPERTY

**flex-start**



**flex-end**



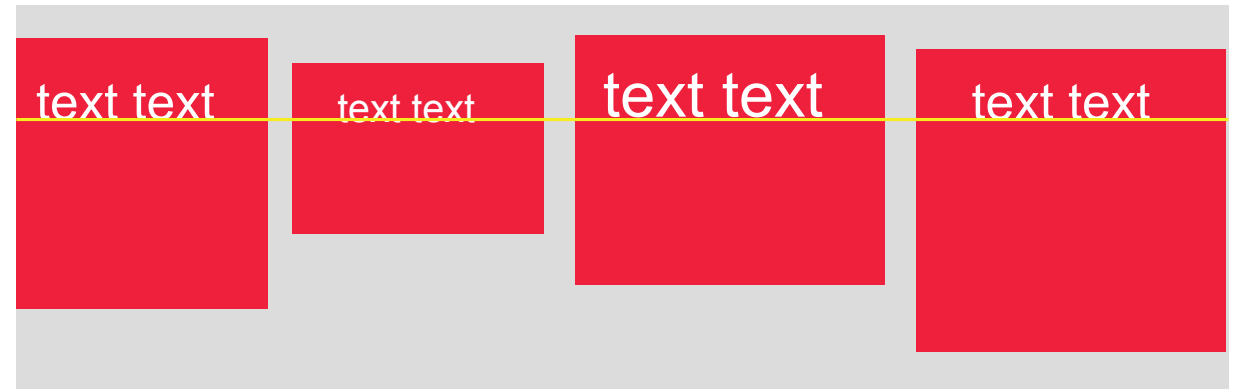
**center**



**stretch**



**baseline**



# FLEX ITEMS

- Alignments can be separately set for the individual items.

```
#item-2 {  
    align-self: center;  
}
```

# FLEXBOX EXAMPLE

HTML

CSS

Result

FORK ON  
CODEPEN

1 ▾ <h1>CODEPEN EXAMPLE</h1>

2 ▾ <h3>Use the link below to visit this example</h3>

LIVE

CODEPEN  
EXAMPLE

Use the link below to visit this  
example

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

# FLEXBOX PROS & CONS

## PROS

- Easily center elements vertically!!!!
- Ability set individual element alignment

## CONS

- Requires an outer container
- Only works in one direction at a time
- No flex box line break
- No ability to control the gap width

# FLEXBOX SUPPORT TABLE

## CSS3 Multiple column layout - CR

Method of flowing information in multiple columns

Usage	% of all users	
Global	86.66%	+ 9.14% = 95.8%
unprefixed:	86.66%	+ 4.89% = 91.55%
U.S.A.	89.59%	+ 6.5% = 96.09%
unprefixed:	89.59%	+ 3.72% = 93.31%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?							
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android
				<div>123.1-6.1</div>	10.1	<div>123.2-6.1</div>						
				<div>17-8</div>	11.5-12.1	<div>17-8.4</div>						
6-9		<div>12-51</div>	<div>124-49</div>	<div>19-9.1</div>	<div>1215-36</div>	<div>19-9.3</div>						
10	12-16	<div>152-61</div>	50-68	10-11.1	37-54	10-11.2		<div>122.1-4.4.4</div>	<div>127</div>	12-12.1		
11	17	<div>162</div>	69	12	55	11.4	all	67	<div>1210</div>	<div>146</div>	69	<div>162</div>
	18	<div>163-64</div>	70-72	TP		12						

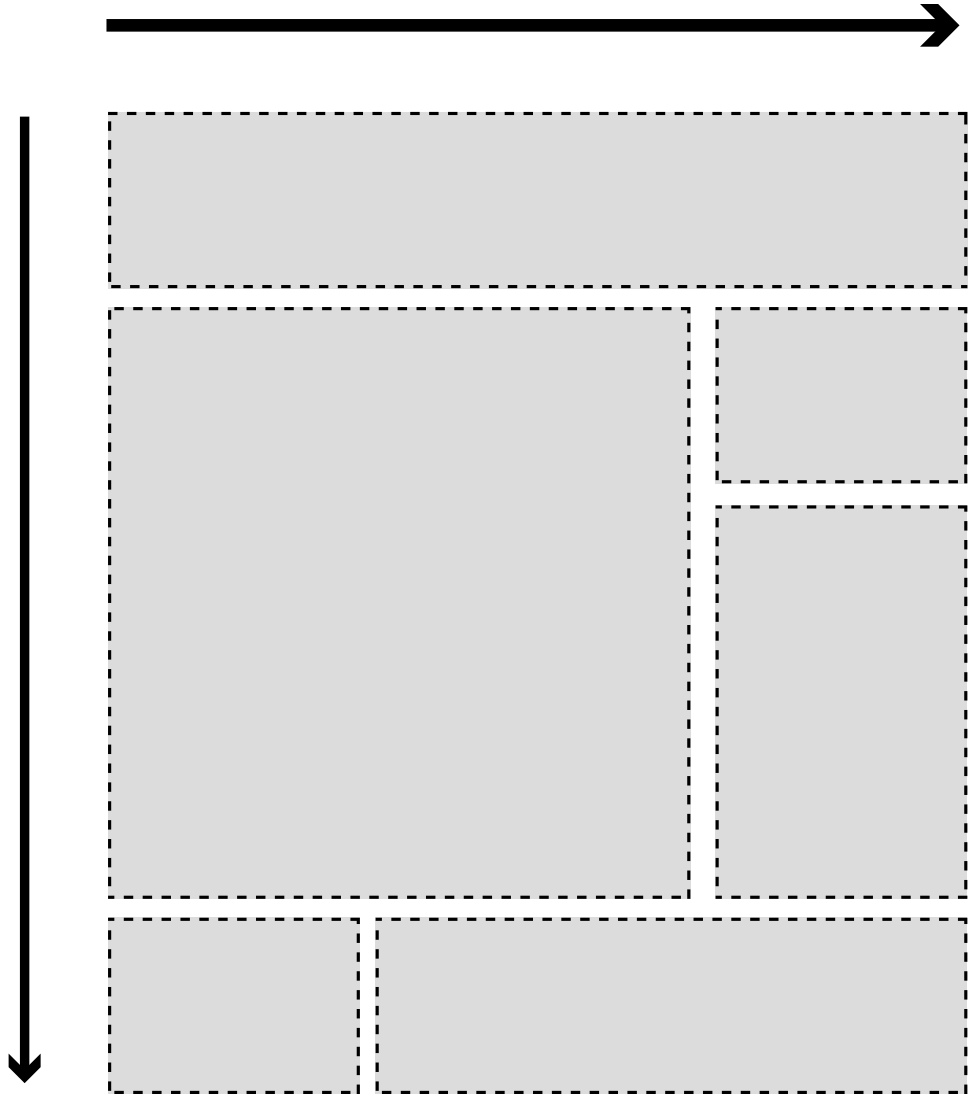
# WHEN I USE IT

- When I want to center something vertically.
- When I want elements to have different alignments.
- When I want elements to wrap in rows.
- When I don't have to support Internet Explorer.

**GRID**



# INTRODUCING CSS GRID



CSS Grid is the **newest** standard. It allows us to more easily layout elements in 2 directions.

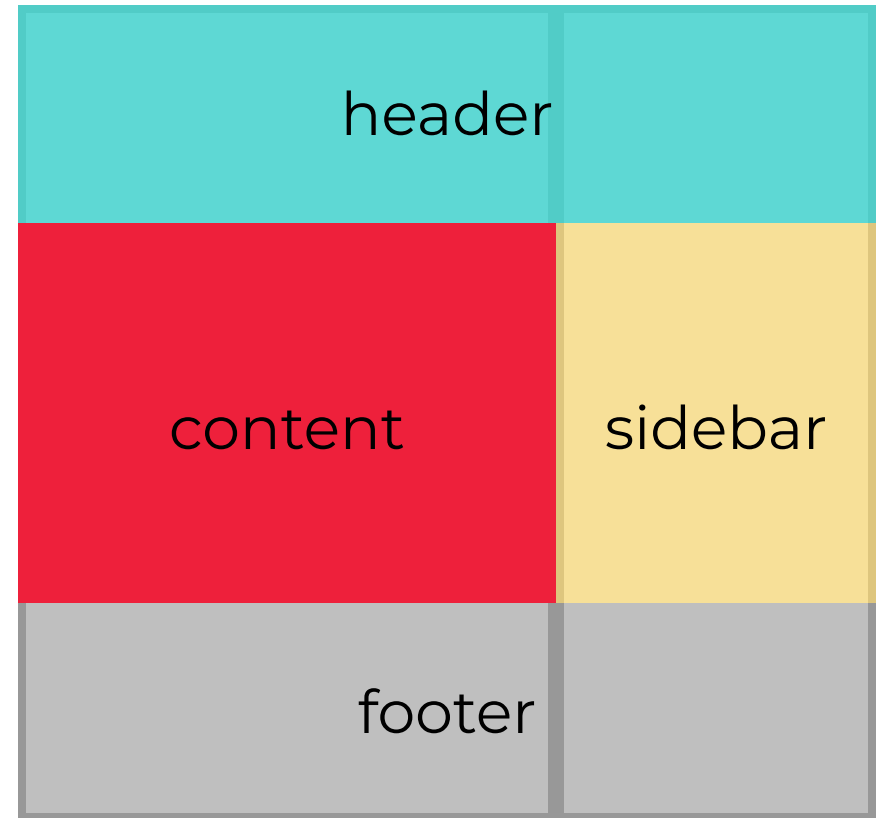
# GRID CONTAINER

Grids also start with a container that provides the basic grid definition.

```
.grid {  
  display: grid;  
  /* column definitions */  
  grid-template-columns: 75% 25%;  
  /* row definitions */  
  grid-template-rows: 10vh auto auto;  
  /* named areas */  
  grid-template-areas:  
    "header header"  
    "content sidebar"  
    "footer footer";  
}
```

# GRID TEMPLATE AREAS

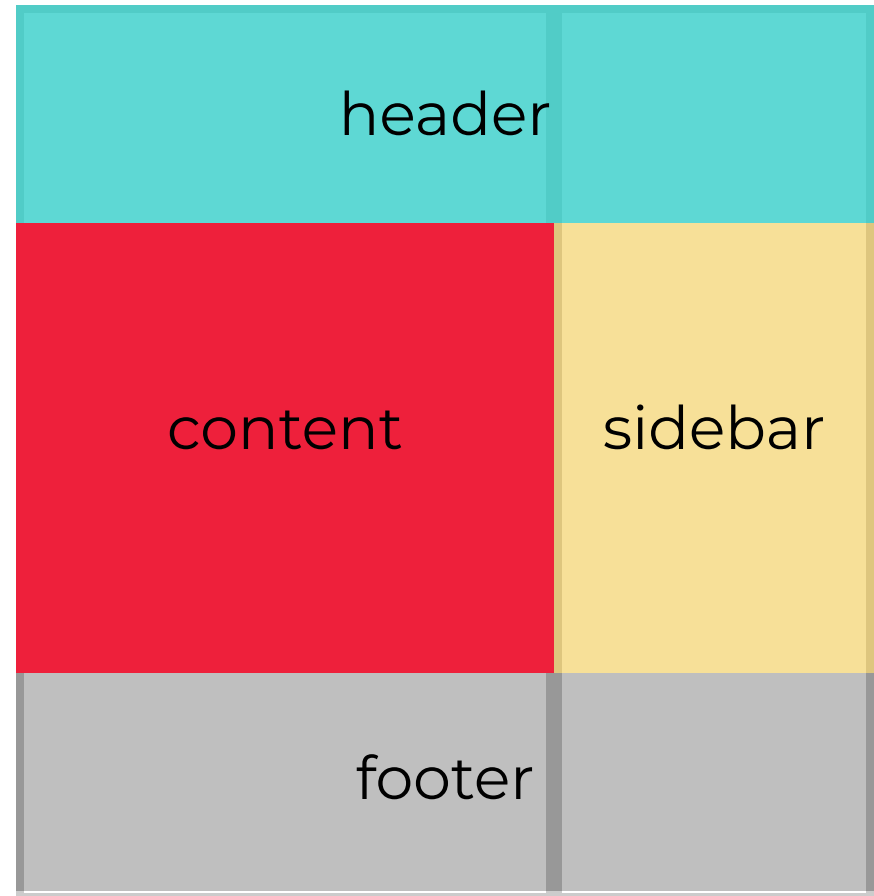
```
.my-grid-container {  
  
  grid-template-columns: 75% 25%;  
  
  /* named areas */  
  grid-template-areas:  
    "header header"  
    "content sidebar"  
    "footer footer";  
  
}
```



Each **row** is surrounded by `" "`.

# GRID AREAS

```
.my-header {  
  grid-area: header;  
}  
  
main {  
  grid-area: content;  
}  
  
aside {  
  grid-area: sidebar;  
}  
  
footer {  
  grid-area: footer;  
}
```



# GRID PROS & CONS

## PROS

- Control in two directions
- Allows elements to be placed precisely with different alignments
- Gap width control

## CONS

- Requires an outer container
- Lowest support of all layout techniques because it's the newest

# GRID SUPPORT TABLE

## CSS Grid Layout 📄 - CR

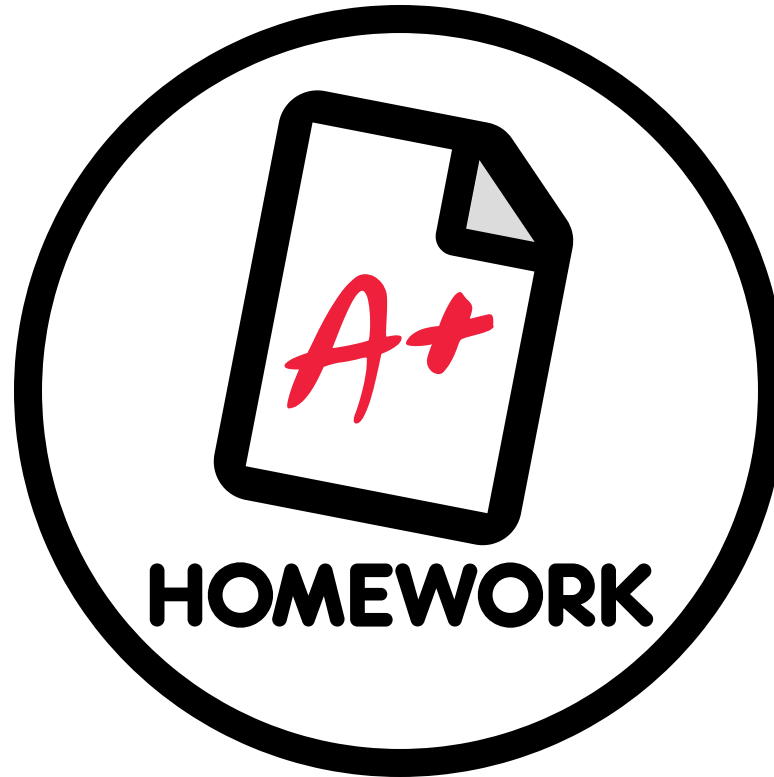
Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors. Includes support for all **grid-\*** properties and the **fr** unit.

Usage	% of all users
Global	84.95% + 2.91% = 87.85%
unprefixed:	84.95%
U.S.A.	87.75% + 2.57% = 90.31%
unprefixed:	87.75%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?								
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	
		2-18	4-24										
		19-39	25-28										
6-8		3 40-51	1 29-56	3.1-5.1	10-27	3.2-5.1							
9	2 12-15	4 52-53	4 57	6-10	1 28-43	6-10.2		2.1-4.1					
2 10	16	54-61	58-68	10.1-11.1	44-54	10.3-11.2		4.2-4.4.4	7	12-12.1			
2 11	17	62	69	12	55	11.4	all	67	10	46	69	62	
	18	63-64	70-72	TP		12							

# WHEN I USE IT

- Whenever I don't care about older browser support issues
- When I want to layout components (such as a card design like the homework).



## **WEEK 2 HOMEWORK**

<https://github.com/jmeade11/FEWD/Class4/homework>



# **HOMEWORK FOR NEXT CLASS**

- This week the assignment is much tougher, but you **can** do it. Understanding how to layout pages and page components is an absolute must, so make sure you really grok these techniques.
- Remember what you learned today about the DevTools. Rely on them if you run into any snags along the way.

# **EXIT SURVEY**

<https://goo.gl/EB4XFw>

**GO BUILD  
AWESOME THINGS!**