Week 7: Bootstrap

INFSCI 2560
Web Standards & Technologies

Housekeeping

Assignments are now PROJECTS.

Next week

- Exam 1
- Project 2 will be assigned

Topics

- CSS - advanced topics
- Javascript - advanced topics
- **Break**
- Bootstrap Overview
- Activity

Digging Deeper with CSS

- **Pseudo-element:** a keyword added to a selector that lets you style a specific part of the selected element(s). (documentation)
  - Examples - `::first-line`, `::before`, `::after`, `::first-letter`, `::selection`

- **Pseudo-classes:** is used to define the special state of an element. (documentation)
  - Examples - `:hover`, `:active`, `:checked`, `:disabled`, `:last-child`, `:nth-child`

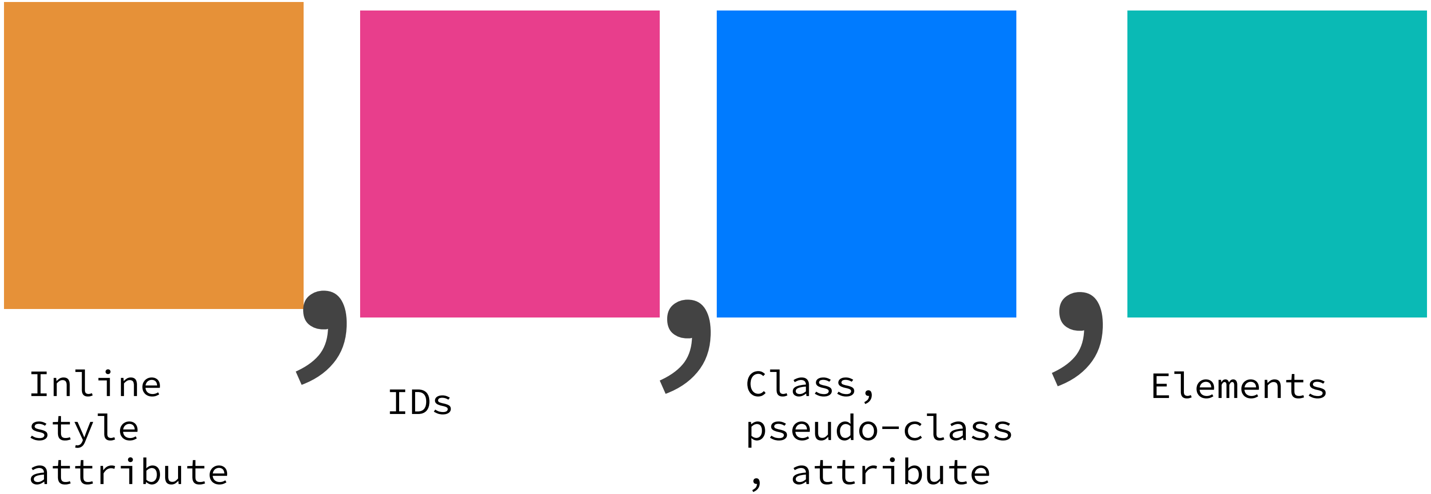
CSS: Rule of Specificity

- When selectors have an equal specificity value, the latest rule is the one that counts.
- When selectors have an unequal specificity value, the more specific rule is the one that counts.
- Rules with more specific selectors have a greater specificity
  - Type selectors (least specific)
  - Class selectors
  - ID selector (most specific)
- `!important` - overrides any other declarations.

What this means:

- The latest rule in the CSS counts
- Embedded styles override external styles
- Class selectors are more specific than element selectors
- If two selectors apply to the same element, the one with higher specificity wins.
-

Calculating the Rule of Specificity



Inline style attribute

IDs

Class, pseudo-class, attribute

Elements

Most specific > Least specific

The Universal Selector (*)

*, body * and similar selectors have zero specificity.

0,0,0,0

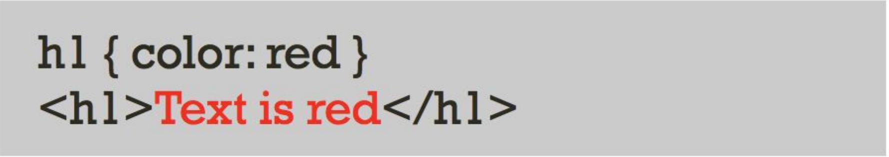
Inherited values also have 0 specificity

Elements and Pseudo-Elements

All element selectors

Includes pseudo-elements like :before and :after.

0,0,0,1

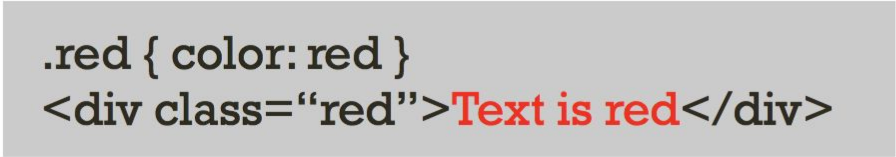


```
h1 { color: red }
<h1>Text is red</h1>
```

Classes, pseudo-classes and attributes

This group includes `.classes`, `[attributes]` and pseudo-classes such as `:hover`, `:focus` etc.

0,0,1,0

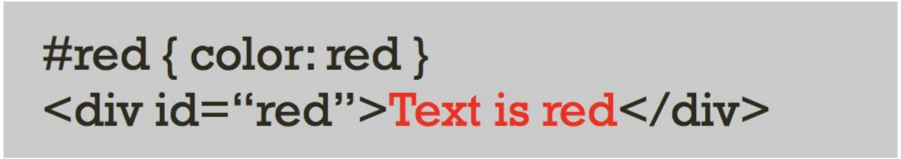


```
.red { color: red }
<div class=“red”>Text is red</div>
```

IDs

ID is an identifier for your page elements, such as #red.

0,1,0,0

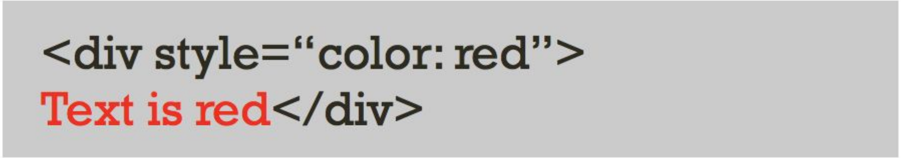


```
#red { color: red }
<div id="red">Text is red</div>
```

Inline styles

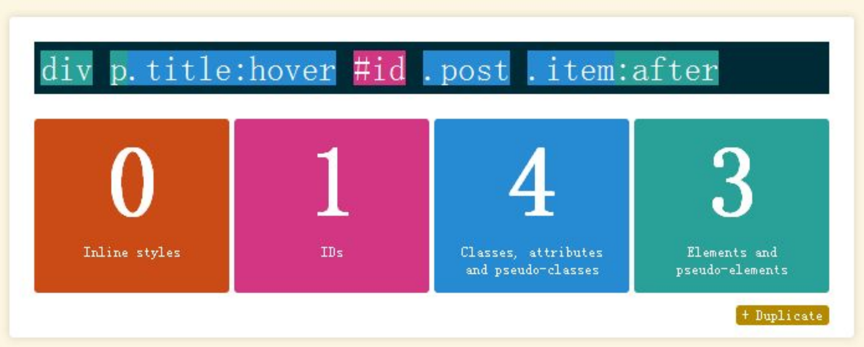
An inline style lives within your HTML document. It is attached directly to the element to be styled.

1,0,0,0



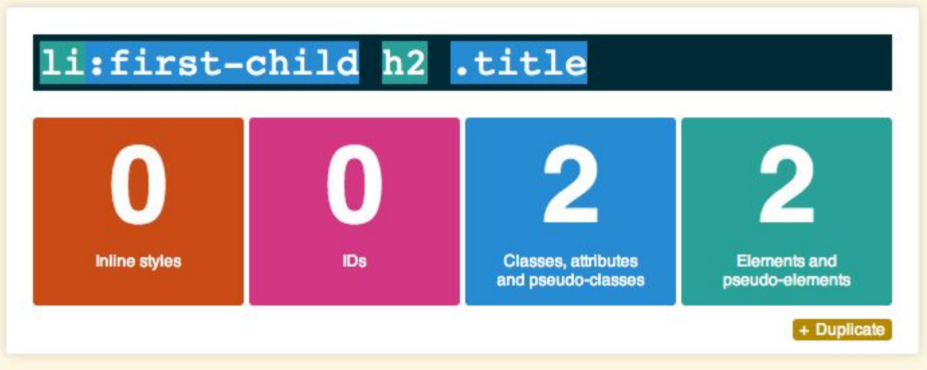
```
<div style="color: red">
Text is red</div>
```

Specificity Examples



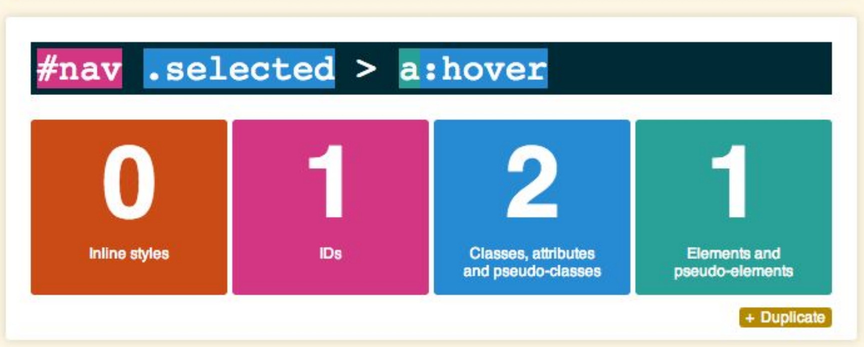
```
div p.title:hover #id .post .item:after
```

0	1	4	3
Inline styles	IDs	Classes, attributes and pseudo-classes	Elements and pseudo-elements



```
li:first-child h2 .title
```

0	0	2	2
Inline styles	IDs	Classes, attributes and pseudo-classes	Elements and pseudo-elements



```
#nav .selected > a:hover
```

0	1	2	1
Inline styles	IDs	Classes, attributes and pseudo-classes	Elements and pseudo-elements

CSS Resources

Avoid using !important

Great article explaining rule of specificity and another one

Digging Deeper with JavaScript

Common JavaScript String Operations

```
var str = 'The quick brown fox jumps over the lazy dog.’
```

```
str.length

str.includes(“fox”) //true

str.indexOf(val) //16

var arr = str.split(‘ ‘)

//[‘The’, ‘quick’, ‘brown’, ...‘dog’]

str.split(‘ ‘,2)

//[‘The’, ‘quick’]
```

```
var regex = /dog/gi;
str.replace(regEx, replaceWithThis)
p.replace(regex, ‘monkey’)

str.toLowerCase();

str.toUpperCase();
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp

Common Javascript Array Methods

```
const ages = [19, 20, 22, 16, 12];
```

```
ages.length    //5

console.log(...ages);
//19 20 22 16 12

ages.includes(“30”)    //false

arr.some(function(person){
             return person >18;
         });) //true
```

```
arr.every((person) => person >18);
//false

ages.concat(moreAges)

filteredAges = arr.filter(person) =>
person >18) // [19, 20, 22]
```

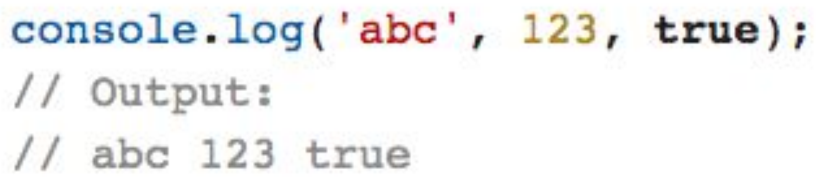
Debugging with JavaScript

Printing to the console

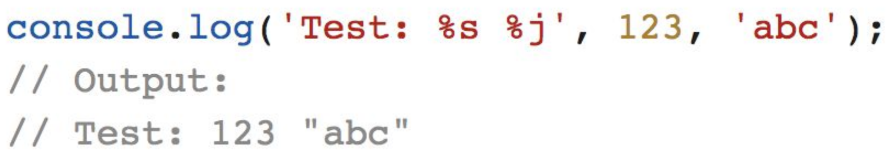
`Console.log()`

`Console.error()`

The **`debugger`** Keyword

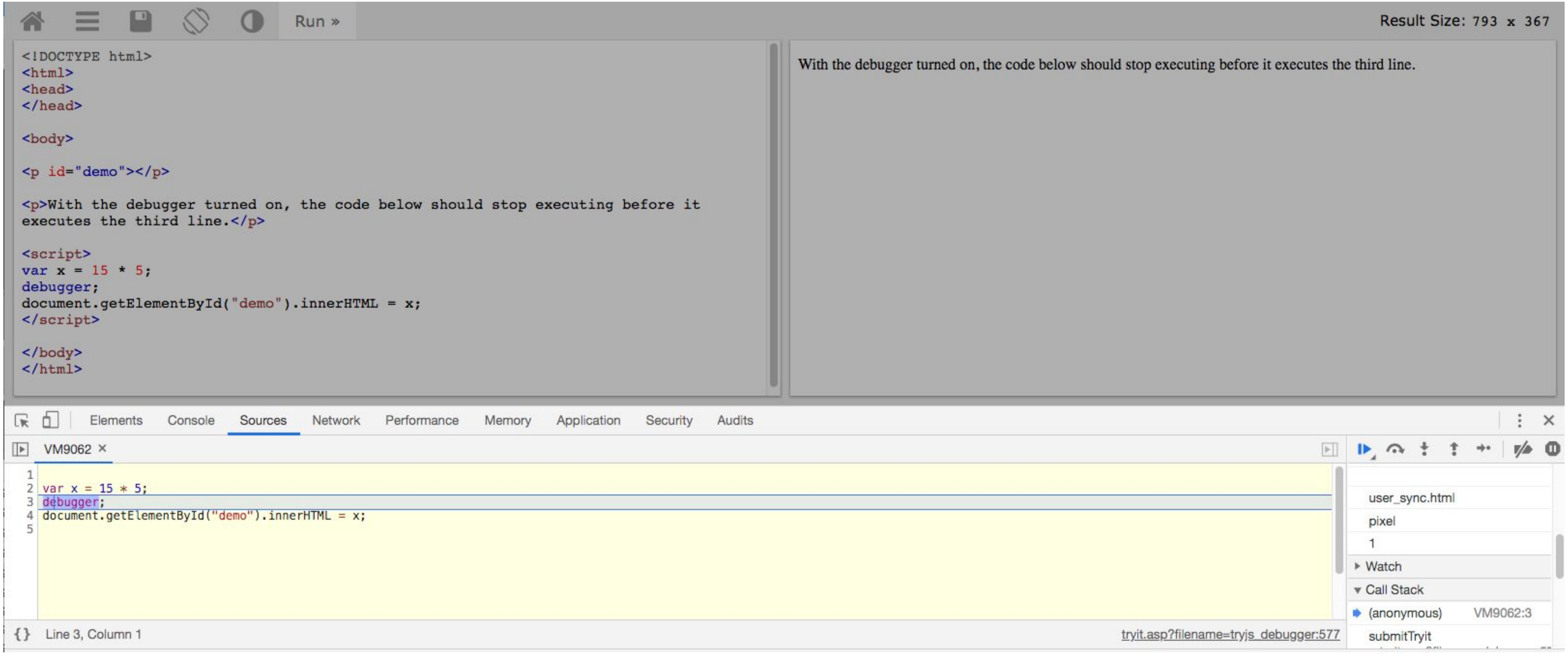


```
console.log('abc', 123, true);
// Output:
// abc 123 true
```



```
console.log('Test: %s %j', 123, 'abc');
// Output:
// Test: 123 "abc"
```

Debugger Example



```
<!DOCTYPE html>
<html>
<head>
</head>

<body>

<p id="demo"></p>

<p>With the debugger turned on, the code below should stop executing before it
executes the third line.</p>

<script>
var x = 15 * 5;
debugger;
document.getElementById("demo").innerHTML = x;
</script>

</body>
</html>
```

With the debugger turned on, the code below should stop executing before it executes the third line.

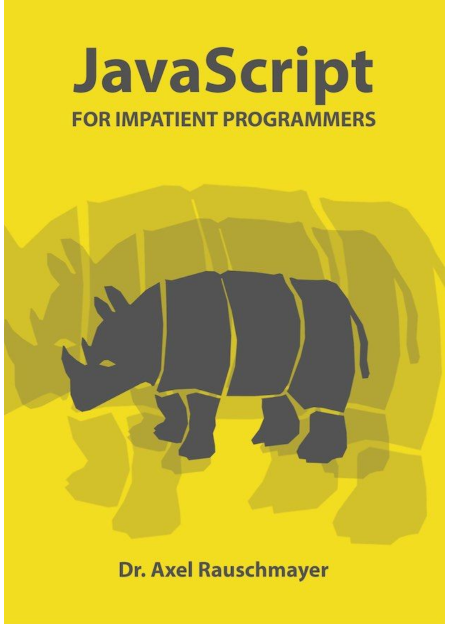
```

var x = 15 * 5;
debugger;
document.getElementById("demo").innerHTML = x;

```

More JS Resources

These JavaScript methods will boost your skills in just a few minutes

Most Common Javascript Methods and Gotchas

MDN documentation on Arrays and Strings



Questions?

Break (5 minutes)

CSS Frameworks

Making accessible, responsive, usable, and visually appealing websites is a lot of work

There are lots of quirks

Not all browsers implement the standards in the same way

Or at all

Popular Frameworks

- PureCSS
- Foundation
- Bootstrap

Bootstrap

- Open source HTML, CSS and JavaScript Framework used for front-end web development
- Originally Twitter Bootstrap
- Created as an internal tool with the aim of eliminating inconsistencies among developers using different tools. Soon, many developers started contributing to the project and then it was released as an open-source project.

Example websites built with Bootstrap

Advantages

- Able to create mobile-first, responsive websites *quickly and easily.*
- Free
- Very popular (large community)
- Easy to get started
- Easily customizable
- Excellent documentation

Disadvantages

- You can use it without actually understanding CSS.
- Without customization, your site may look like other Bootstrap websites
- Large file size

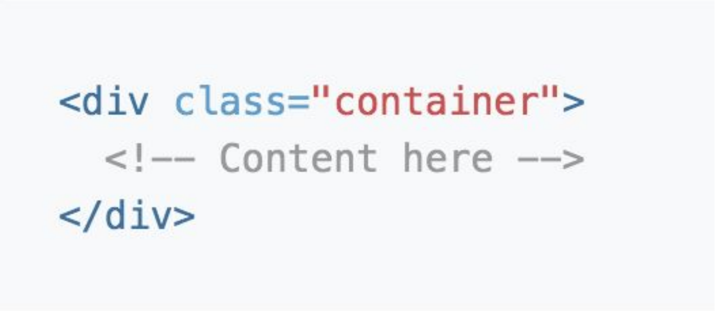
Bootstrap Overview

This is a QUICK introduction to the key features of Bootstrap. You are expected to read the documentation/tutorial to fully understand Bootstrap and complete the activity.

Bootstrap Overview

Containers

- Most basic layout element
- Required
- Responsive, fixed-width or fluid-width



```
<div class="container">
  <!-- Content here -->
</div>
```

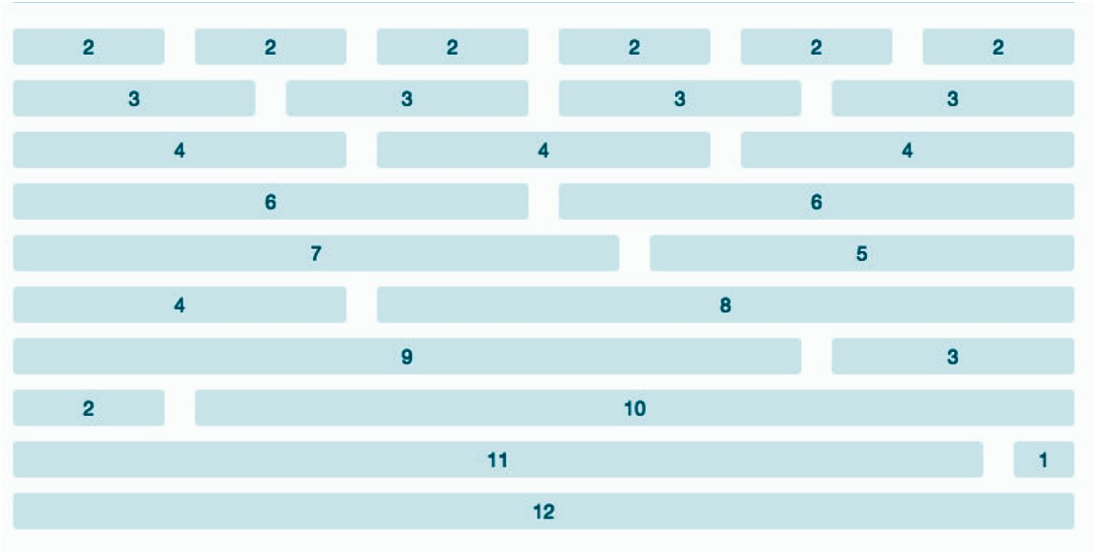
Responsive Breakpoints

- 576px - small devices (phones)
- 768 - medium devices (tablets)
- 992 - large devices (desktop)
- 1200 - extra large devices

https://getbootstrap.com/docs/4.2/layout/overview/

Grid System

- Fully responsive
- Rows - container for columns
- Columns
- Max 12 columns per row



2 2 2 2 2 2

3 3 3 3

4 4 4

6 6

7 5

4 8

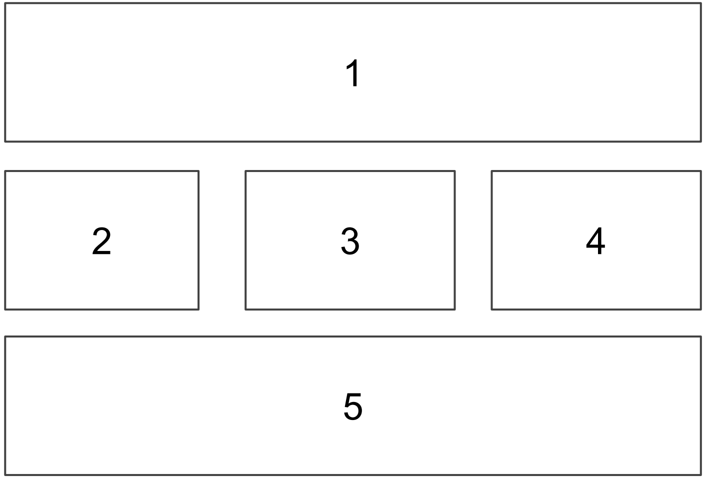
9 3

2 10

11 1

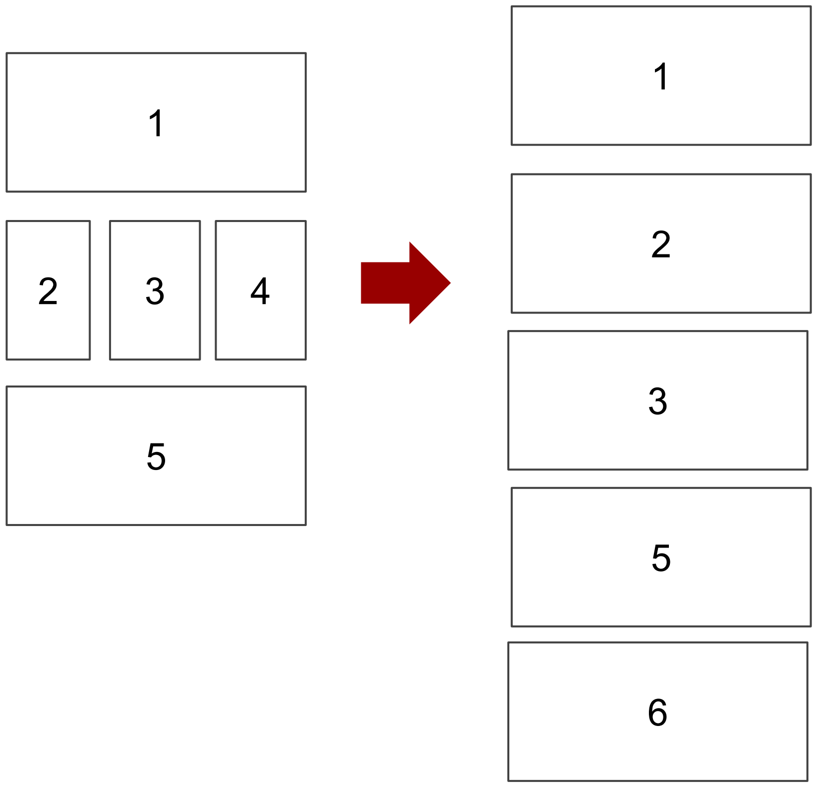
12

Basic Bootstrap Layout



```
<div class =”container”>
     <div class = “row”>
          <div class = “col”>1 </div>
     </div>

     <div class = “row”>
          <div class = “col”>2 </div>
          <div class = “col”>3 </div>
          <div class = “col”>4 </div>
     </div>
     <div class = “row”>
          <div class = “col”>5 </div>
     </div>

</div>
```

Making it responsive



```
<div class =”container”>
     <div class = “row”>
          <div class = “col”>1 </div>
     </div>

     <div class = “row”>
          <div class = “col-sm”>2 </div>
          <div class = “col-sm”>3 </div>
          <div class = “col-sm”>4 </div>
     </div>
     <div class = “row”>
          <div class = “col”>5 </div>
     </div>

</div>
```

Activity 6 - Bootstrapping

In this activity, we are going to use Bootstrap to create a mobile-first, responsive layout.

https://infsci-2560-edmonds.glitch.me/week/seven.html