

Introduction to Bootstrap



Responsive Web Design

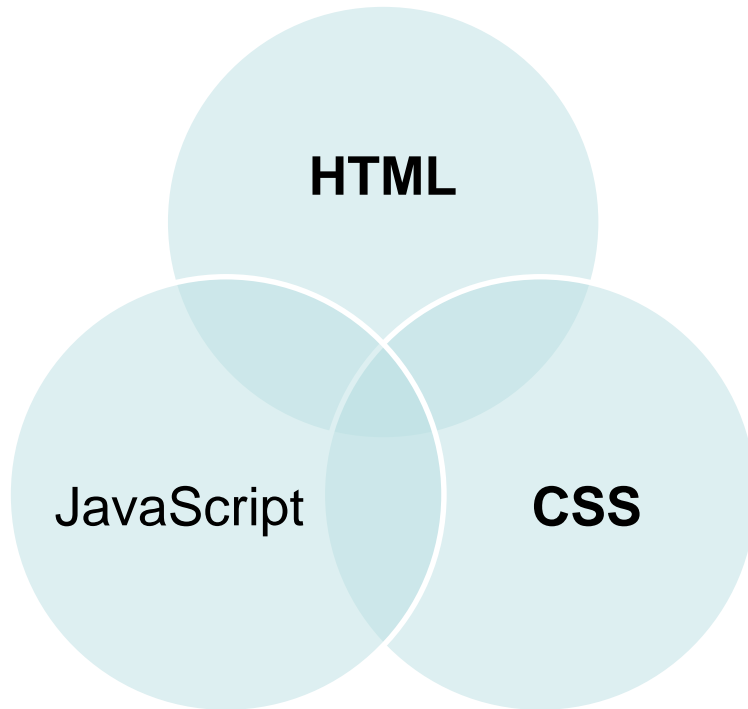
What is RWD?

- The idea of responsive web design has been called many things: “flexible”, “fluid”, “elastic”, etc.
- Bottom line: a responsive web site (or web interface) is an interface that adapts to browser dimensions
- Responsive Web Design is important because ...



... we live in a world full of different devices.

Where this topic fits...



HTML and CSS are the basic tools used to implement a Responsive Web Design.

Today's material **compliments** JavaScript, but does not **require** JavaScript

RWD defined

Ethan Marcotte is a web designer, author, and public speaker. See

<http://ethanmarcotte.com>

He described the theory and practice of responsive web design in his brief 2011 book titled *Responsive Web Design*.

Ethan Marcotte, defines RWD as these three things:

1. A Flexible Grid Layout
2. Flexible Images
3. Media Queries

Also in 2011, something called Bootstrap was released by a development team at Twitter.

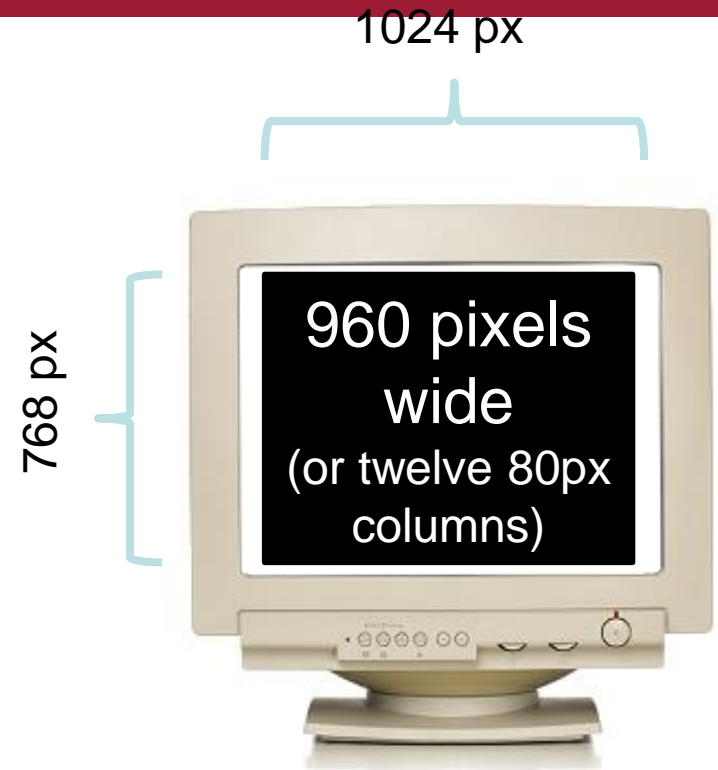
1. Bootstrap is a CSS framework. (A framework is a collection of existing technologies, bundled together in a new, novel, and consistent way.)
2. Bootstrap is a convenient way to get a flexible grid layout.
3. Bootstrap is free to use, Open Source, and remains popular today.

Bootstrap's flexible grid layout

Bootstrap layout is oriented around the notion of ***spans***. Conceptually, a typical Bootstrap layout slices a monitor into 12 columns called ***spans***.

Twelve might seem like an awkward number to choose, but it made sense back in 2011. Web designers were attempting to accommodate 1024x768 monitors.

Designers found it convenient to assume a 960 pixel width screen. 960 pixels is *close* to 1024 and conveniently divisible by 1, 2, 3, 4, 5, 6, 8, 10, **12**, 15, and 16



Look at index01.html and index02.html to see the benefit of a flexible grid.



What is Bootstrap?

1. Bootstrap is a *framework*.
2. A *framework* is often nothing more than a collection of existing technologies, bundled together in a new, novel, and consistent way.
3. In today's class we'll be using Bootstrap 4, the latest major version of this framework.
4. So.... to be more precise.... Bootstrap is a CSS framework.
5. Bootstrap is free to use and Open Source.
6. It was developed by Twitter (and it is still maintained by Twitter.)
7. Arguably, Bootstrap is more popular than Twitter itself. 😊

How to get bootstrap...



The viewport meta tag is important! See:

https://www.w3schools.com/css/css_rwd_viewport.asp

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
</head>
```



Pro tip! Knowing what a CDN URL is, and how to add one to your solution, is a powerful way to extend the options available to you, the programmer.

So, *just like we got the jQuery library* from a Content Delivery Network (CDN) reference, we get the bootstrap framework from yet another CDN.

Next step...

Once you have linked to a bootstrap.min.css file, you get a whole bunch of classes set up for you, for free!

1. The “container” class is essential. Everything needs to be inside of it.
2. The “row” class is almost as important. Use it to specify groupings of columns.
3. Finally, there are a number of classes named with the pattern *col-grid class-?*. (huh?)

```
<div class="col-md-10">Some Content here.</div>
```



Grid Class

Where to begin...

- 1.What do I need in the HTML head tag?
- 2.Bootstrap's responsive grid layout
- 3.Conventions for styling text
- 4.Conventions for icons (Glyphs)
- 5.Bootstrap themes

A return to the flexible grid...

Once you have linked to a bootstrap.min.css file, you get a whole bunch of classes set up for you, for free!

1. The “container” class is essential. Everything needs to be inside of it.
2. The “row” class is almost as important. Use it to specify groupings of columns.
3. Finally, there are a number of classes named with the pattern col-?-?. (huh?)

```
<div class="col-md-12">Some Content here.</div>
```

Remember this?

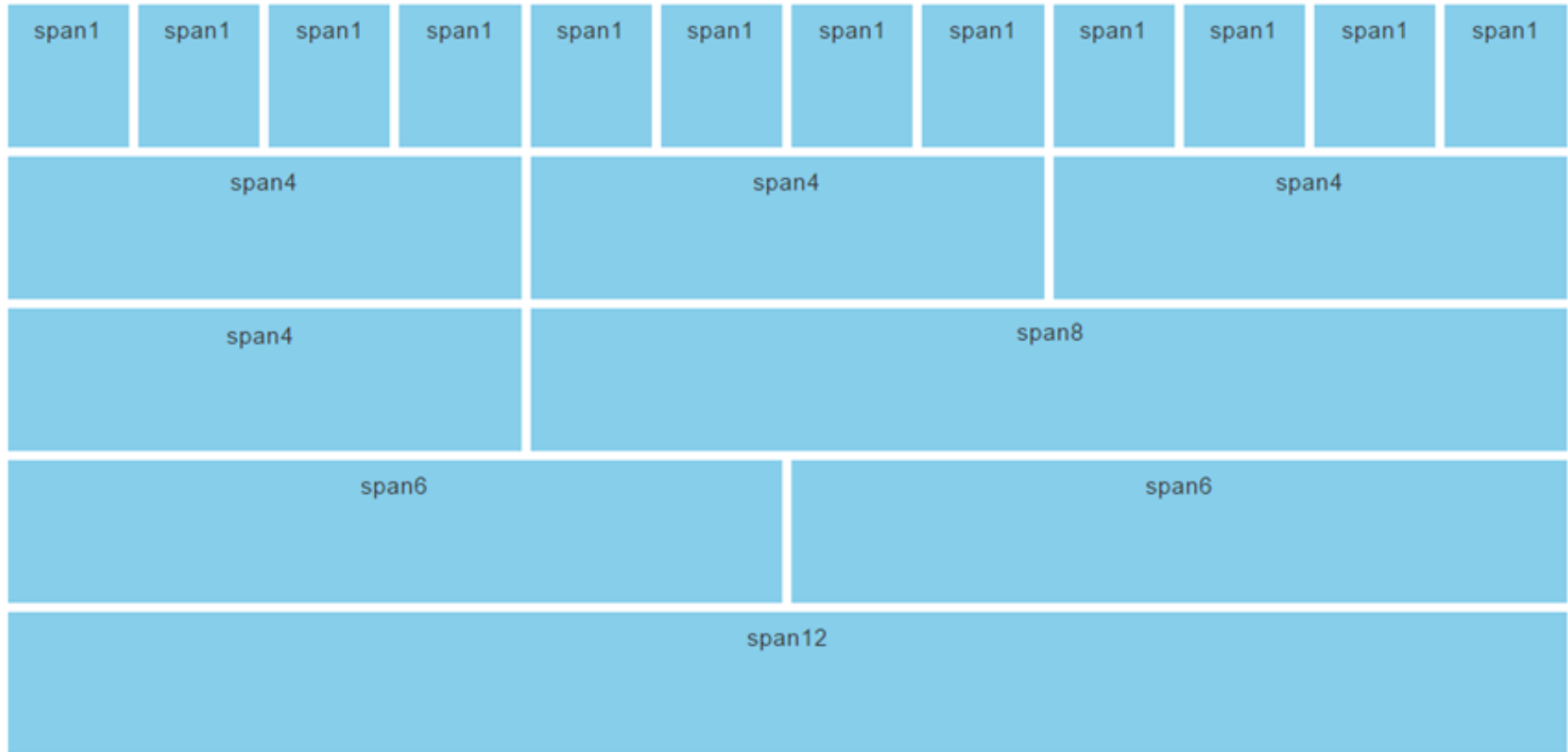
80px 80px 80px 80px 80px 80px 80px 80px 80px 80px 80px 80px
8.333% 8.333% 8.333% 8.333% 8.333% 8.333% 8.333% 8.333% 8.333% 8.333% 8.333% 8.333%

--	--	--	--	--	--	--	--	--	--	--	--

This allows us to break the page into columns or “spans”

160px	800px										

Spans are indicated in units of 12ths



In bootstrap, the column spans always need to add up to 12

Grid Classes

To specify a particular column's width, we specify both the grid class, and the span of the column.

The Bootstrap 4 grid system has five classes:

- (Extra small - for phones / viewport width less than 576px)
- sm (Small - for tablets / viewport width of 576px or greater)
- md (Medium - for typical laptops / width of 768px or greater)
- lg (Large - for larger desktops / width of 992px or greater)
- xl (eXtra Large - width of 1200px or greater)

Tip: Each class scales up! So a best practice is to design around the smallest anticipated viewport size.

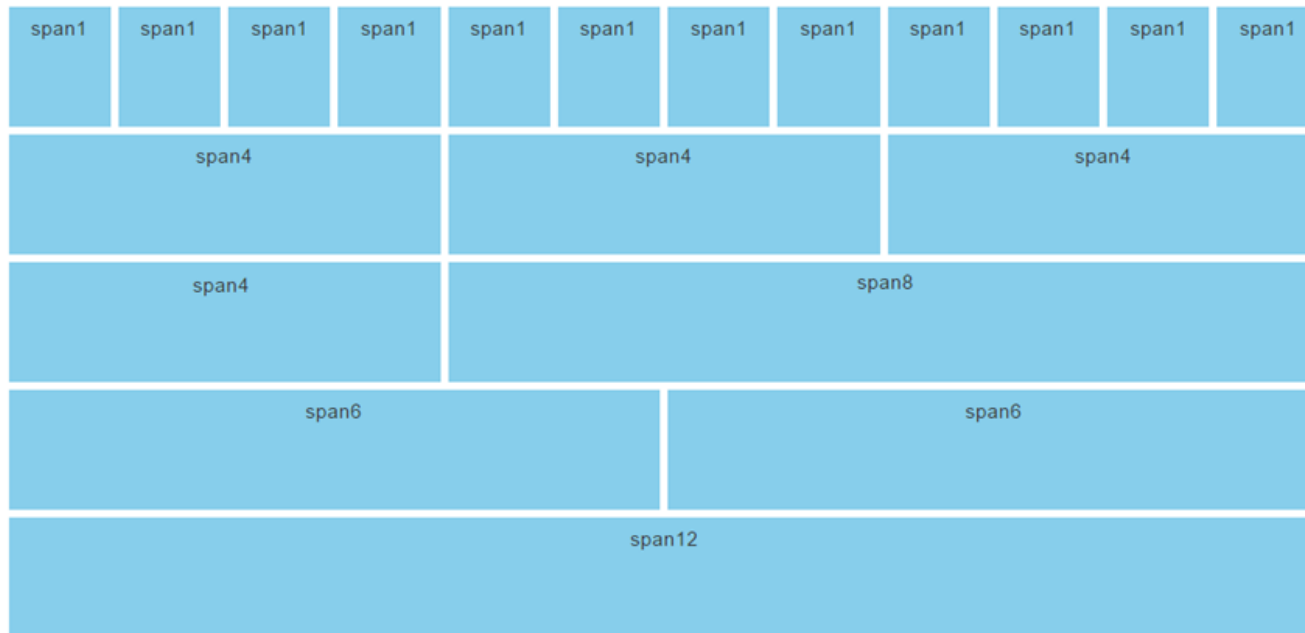
Spans should add up to 12

After you specify the grid class, indicate the width of the column in spans.

```
<div class="col-md-10">Some Content here.</div>
```



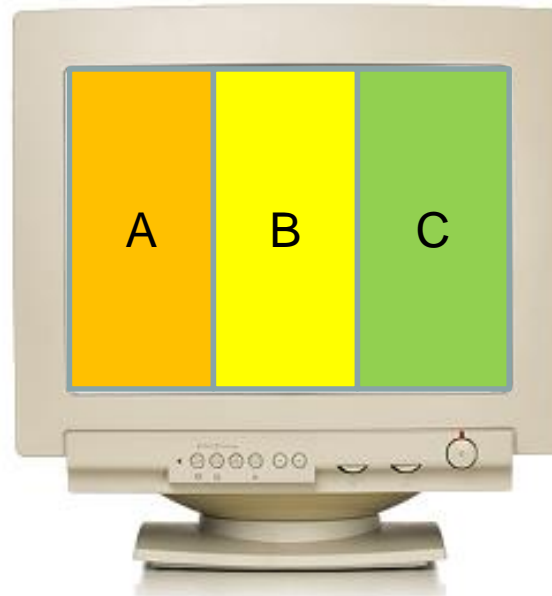
Width in spans



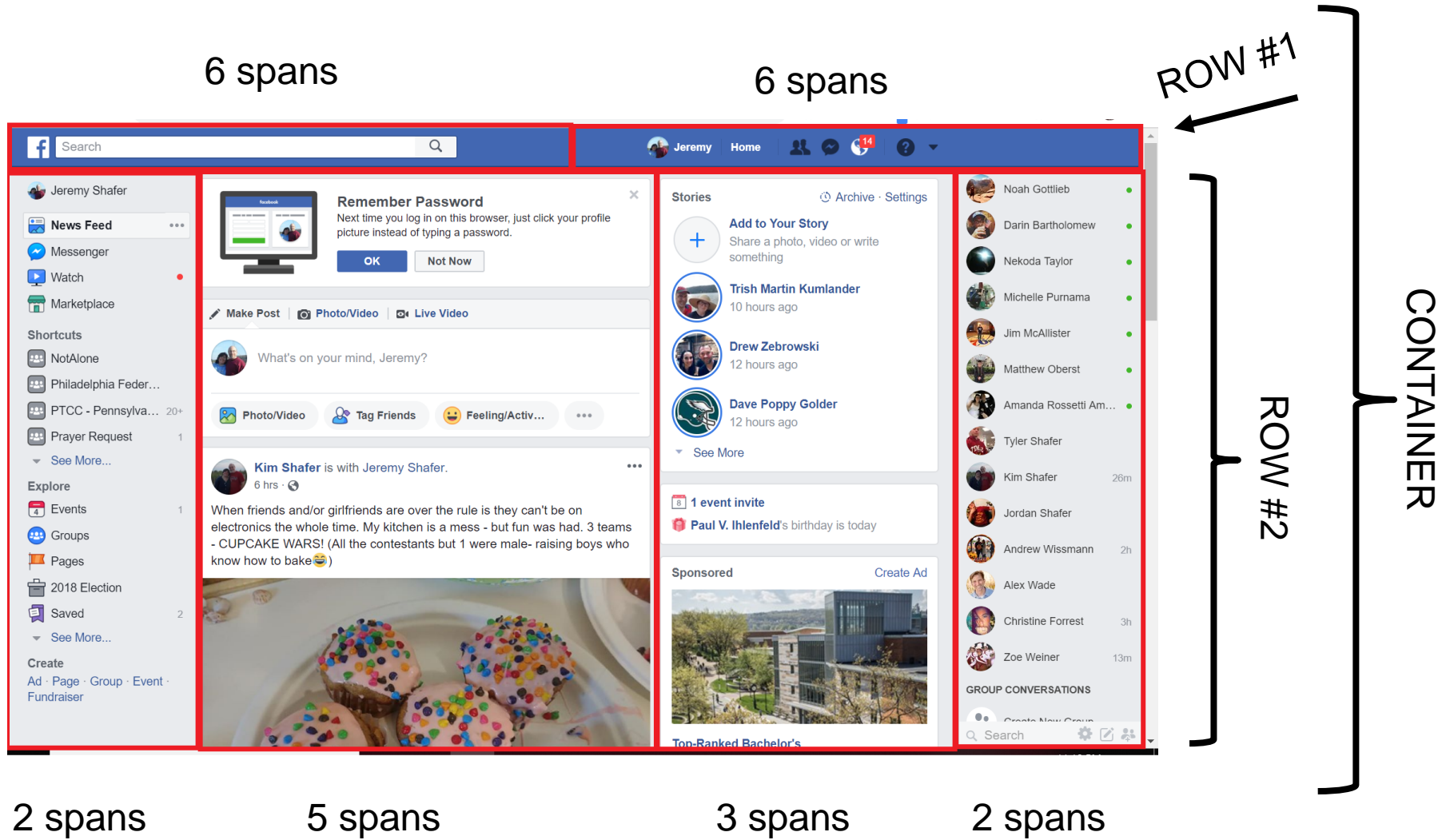
- In bootstrap, the column 12 spans equate to the full width of the viewport.
- The bootstrap grid system is responsive, and the columns will re-arrange automatically depending on the screen size.
- A best practice is to make sure that the column widths add up to 12 spans. (There are occasional exceptions to that best practice.)

Some sample code for a 3 column layout

```
<div class="container">  
  <div class="row">  
    <div class="col-md-4">A</div>  
    <div class="col-md-4">B</div>  
    <div class="col-md-4">C</div>  
  </div>  
</div>
```



Another example



Contextual Classes

Bootstrap also gives us contextual classes. For example, there are CSS classes for background colors: `.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, and `.bg-danger`

```
<p class="bg-primary">This text is important.</p>
```

```
<p class="bg-success">This text indicates success.</p>
```

```
<p class="bg-info">This text represents some information.</p>
```

```
<p class="bg-warning">This text represents a warning.</p>
```

```
<p class="bg-danger">This text represents danger.</p>
```

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

The class names are “contextual” because they convey the ***meaning*** of the style, not its ***appearance***.

Contextual Classes - buttons

```
<input type='button' class='btn btn-primary' value='Primary'>
```

Primary

```
<input type='button' class='btn btn-success' value='Success'>
```

Success

```
<input type='button' class='btn btn-info' value='Information'>
```

Information

```
<input type='button' class='btn btn-warning' value='Warning'>
```

Warning

```
<input type='button' class='btn btn-danger' value='Danger'>
```

Danger

Text styling

<code>.font-weight-bold</code>	Bold text
<code>.font-weight-bolder</code>	Bolder text
<code>.font-italic</code>	Italic text
<code>.font-weight-light</code>	Light weight text
<code>.font-weight-lighter</code>	Lighter weight text
<code>.font-weight-normal</code>	Normal text
<code>.lead</code>	Makes a paragraph stand out
<code>.small</code>	Indicates smaller text (set to 80% of the size of the parent)
<code>.text-left</code>	Indicates left-aligned text
<code>.text-*-left</code>	Indicates left-aligned text on small, medium, large or xlarge screens
<code>.text-break</code>	Prevents long text from breaking layout
<code>.text-center</code>	Indicates center-aligned text
<code>.text-*-center</code>	Indicates center-aligned text on small, medium, large or xlarge screens
<code>.text-decoration-none</code>	Removes the underline from a link

Text styling

<code>.text-right</code>	Indicates right-aligned text
<code>.text-*-right</code>	Indicates right-aligned text on small, medium, large or xlarge screens
<code>.text-justify</code>	Indicates justified text
<code>.text-monospace</code>	Monospaced text
<code>.text-nowrap</code>	Indicates no wrap text
<code>.text-lowercase</code>	Indicates lowercased text
<code>.text-reset</code>	Resets the color of a text or a link (inherits the color from its parent)
<code>.text-uppercase</code>	Indicates uppercased text
<code>.text-capitalize</code>	Indicates capitalized text
<code>.initialism</code>	Displays the text inside an <code><abbr></code> element in a slightly smaller font size
<code>.list-unstyled</code>	Removes the default list-style and left margin on list items (works on both <code></code> and <code></code>). This class only applies to immediate children list items (to remove the default list-style from any nested lists, apply this class to any nested lists as well)
<code>.list-inline</code>	Places all list items on a single line (used together with <code>.list-inline-item</code> on each <code></code> elements)
<code>.pre-scrollable</code>	Makes a <code><pre></code> element scrollable

Table styling

Class	Description
<code>.table-primary</code>	Blue: Indicates an important action
<code>.table-success</code>	Green: Indicates a successful or positive action
<code>.table-danger</code>	Red: Indicates a dangerous or potentially negative action
<code>.table-info</code>	Light blue: Indicates a neutral informative change or action
<code>.table-warning</code>	Orange: Indicates a warning that might need attention
<code>.table-active</code>	Grey: Applies the hover color to the table row or table cell
<code>.table-secondary</code>	Grey: Indicates a slightly less important action
<code>.table-light</code>	Light grey table or table row background
<code>.table-dark</code>	Dark grey table or table row background

Glyphicons

Bootstrap provides 260 glyphicons from the “Halflings” set at <http://glyphicons.com/>

Glyphicons can be used in text, buttons, toolbars, navigation, forms, etc.

Here are some examples of glyphicons:

Envelope glyphicon: 

Print glyphicon: 

Search glyphicon: 

Download glyphicon: 

Conventions for icons (Glyphs)

A glyphicon is inserted with the following syntax:

```
<span class="glyphicon glyphicon-name"></span>
```

Unfortunately, glyphicons have some problems. Personally, I prefer to use a different css library.

Adding this link to the head:

```
<link rel="stylesheet"  
href="https://maxcdn.bootstrapcdn.com/font-  
awesome/4.5.0/css/font-awesome.min.css">
```

Allows me to do this:

```
<i class="fa fa-name"></i>
```



Flexible Images

Flexible Images are necessary to scale images fluidly.

The CSS attribute that allows us to do this is:

```
max-width: 100%;
```

That is, if we apply that style to an image, then it will expand (or shrink) to the size of its containing element.

Note: It's important to have a containing element!

There are a number of extra steps that are necessary to implement this effect in legacy browsers... like IE9. Assuming that you are using a browser released in the last few **years** you don't need to worry about that!

See
[index04.html](#)



Media Queries

Media queries allow us to override previously declared style sheet attributes using an `@media` rule.

The `@media` rules allow us to customize styles based on the size of the current viewport.

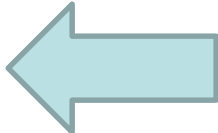
The `@media` rules allow us to deal with exceptions that can't be gracefully addressed by our flexible grid.



Media Queries (2)

For example:

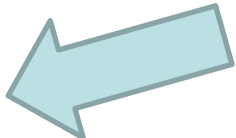
```
@media (max-width: 576px) {  
  img {  
    display: none;  
  }  
}
```



This says ... when the viewport is narrow, don't display any of the images.

Another example:

```
@media (max-width: 576px) {  
  #header{  
    min-height: 20px;  
  }  
}
```



This says ... when the viewport is narrow, style the tag with the id of "header"

See
[index_5.html](#)



Thanks for your attention!

Get to liftin'

