Intro to Bootstrap 4

# 66. What is Bootstrap?

## Front-end development vs back-end development

**Front-End Development**: what the user sees

* The face of your website/mobile app
* The user interface. Buttons, animations, etc.
* Languages: HTML, CSS, Javascript

**Back-End Development**: what the user can’t see. Behind-the-scenes.

* The functionality of your website/mobile app
* The server, the databases, etc.
* Languages: C#, Ruby, PHP, Java, NodeJS, SQL, .NET

## What is Bootstrap?

**Bootstrap** is a bunch of code that was written to be reusable and to help with front-end development.

It is useful for making a website more responsive. A **responsive** website is a website that can respond to the viewport. This means that it can be easily viewed on any device (computer, phone, tablet, etc.)

Bootstrap also allows you can access a bunch of pre-styled elements that you can simply drop into your code with very minimal effort.

# 67. Installing Bootstrap

## CDN

**CDN** (Content Delivery Network): Instead of hosting your website in a single location, you have a whole bunch of points where that website can be accessed.

* This cuts down on the **latency** or how long it takes for your website to load up.

Bootstrap can be accessed using a CDN. This downloads a copy of Bootstrap to the user’s browser if they don’t already have it. The good thing about bootstrap is that it is so popular and used on many websites, which means that its highly likely a user has already visited a website with bootstrap on it and therefore already has it downloaded it to their browser.

We can quickly install bootstrap by linking to it in the <head> of the html as seen below:

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">

# 68. Web Design 101 – Wireframing

**Wireframing**: a low fidelity representation of your website/app design.

* Make a design by simply using pencil and paper
* takes little to no effort to make it

**Mock-up**: a hide fidelity representation of your website/app design.

* Should look like a screenshot of what it should look like while its in action

**Prototype**: an animated version of a mock-up.

* Not necessary

Good Workflow

1. Look at other people’s websites for ideas about layout, structure
2. Look at UI patterns (ex. ui-patterns.com)
3. Create a simple sketched wireframe to plan the layout and see where everything should be
4. Create a mockup using sketch app, photoshop, illustrator
5. Create a prototype

Inspiration:

* ui-patterns.com
* dribbble.com
* sneakpeekit.com
* balsamiq.cloud

# 91. Code Refactoring

To become a better programmer, you must remember **DRY**: **D**on’t **R**epeat **Y**ourself

Code Refactoring (in order of importance)

1. Readability - make sure your code is easy to understand
   1. If you come back to the code in the future, will you be able to understand it?
   2. Is everything organized in a logical way?
   3. Do the variables have understandable names?
   4. Comments to indicate what the code means and how it’s organized
2. Modularity – how easy is it to reuse bits of code
   1. If something breaks, will you be able to narrow down which part of the code is responsible for it?
3. Efficiency - How fast does code run?
4. Length – How long is the code? Usually better to keep it shorter (unless this will cut back on website features)