# Introduction to JavaScript

* Has not correlation with the java language
  + They just named it since Java was the popular language back then
* It is a functional language
  + So not really object oriented but has support for some object-oriented concepts
* Loosely typed
  + Convenient but in a long run very annoying to deal with
* JavaScript is both compiled and interpreted
  + It starts as an interpreter, however, once it notices that you used a function multiple times, it will compile that code (using JIT compiler) into an optimized native machine code.
    - This is extremely important because that makes it so that no other JS implementations that doesn’t have this feature can every beat it in terms of speed.

## ECMAScript

* It is the standardization of JS
* Just like how HTML has HTML 5
* Current version is ES6

## Different Datatypes in JS

* Symbols
* Undefined – It is when you assign a variable and not attach a value to it
* Null – It has no value
* Boolean
* Objects
* Numbers
* Strings

## Scopes in JS

* The scope of a variable describes where it can be used/exist
* Block
  + Cannot be access form outside { }
  + Ex:

{

let x = 2;

}

//Anything cannot use that x variable

* Function
  + Each function you create is a new scope
  + Kinda like methods in C# any variables declared in there can only be used in that method
* Global
  + Can be access anywhere in the JS script regardless of the scope it was initialized in.
* Var keyword will give that variable a global scope
* Let keyword will limit the scope of the variable depending on where you initialized it.

## Other variable declarations

* Const
  + Cannot change the reference value of the variable once it is declared
* {}
  + Declaring an object, you use curly brackets
  + Ex: const person = {Name: “Stephen”, Money: “$10”}
    - Note: we can still change the properties of the person, but you cannot point the person to a complete different object.

## Truthy and Falsey

* In JS, all values have a Boolean equivalent to it
* Meaning you can do “Hello”==96.7 perfectly fine in JS

### What counts as false values?

* FUN0NE
* False
* Undefined
* Null
* 0 (-0 or +0)
* NaN (Not a number)
* Empty

### Anything else will evaluate to true

## Different Function in JS

* Basic Function
* Callback Function
  + A function that has another function in its parameter to be used on its implementation details
* Arrow function
  + Very similar to what we have been doing with () => {}
* IIFE
  + Stands for Immediately Invoked Function Expression
  + Function that will run the moment it is defined

# Encapsulation

* A workaround to try to do OO encapsulation because JS does not have access modifiers
* You must use closure to be able to do encapsulation
* What is a closure?
  + It is when you have a function within a function and a variable defined in the first function but can only be accessed by the inner function.

# Inheritance

* Before ES6, this was harder to implement, and you have to use prototypes inheritance
* Not with ES6, you are able to create classes and method syntax
  + You use the extends keyword to do inheritance

# Introduction to sending and receiving data in JS

## AJAX

* It stands for Asynchronous JavaScript and XML.
* Web dev techniques for updating your page dynamically after receiving your data.
* Used to follow SOAP principles by using XML exclusively but now we can use JSON too.

## XMLHttpRequest

* The main object that AJAX uses to send/receive data.
* XML is in the name because AJAX used to just do XML.

## Fetch API

* Like AJAX but more inline with REST principles
* It uses promises to achieve a synchronicity
  + Promises represents either the completion or failure of an asynchronous operations