# Week 1: Weekly Videos and Curriculum

## 1. JavaScript and Programming

### JavaScript

**JavaScript** is a computer programming language. At a high level, that means we can write commands using the language **JavaScript** to tell a computer what to do. **JavaScript** was initially released in 1995 and has since become the core language that provides functionality to the internet, or rather, to all the websites and web applications on the internet.

**JavaScript** is a lightweight, interpreted, scripting language. FYI: **JavaScript** is not **Java**. **JavaScript** and **Java** are both Oracle languages, but they have **very different** syntax, semantics, and usage.

Here is a brief comparison of **JavaScript** and **Java**:

**JavaScript**:

- JavaScript is a weakly typed, interpreted language.
- JavaScript is a scripting language that allows creation of interactive web pages.
- JavaScript is a dynamic, cross-platform language.
- What Is JavaScript?

**Java**:

- Java is a strongly typed language.
- Java is a compiled, multi-platform, object-oriented programming language
- The Java Tutorials

**JavaScript** is traditionally a front end language - meaning it runs on the user's computer instead of a server - however, in 2009 Node.js came into existence and enabled JavaScript to run outside of the browser, making it possible to use for back end programming.

**In this course, we will be using JavaScript to build front end applications.**

**Hypertext Markup Language** (**HTML**) and **Cascading Style Sheets** (**CSS**) are two languages that are often used hand in hand with **JavaScript**. While we'll learn more about **HTML** and **CSS** later on in this program, know that they focus on building out the look and layout of a website or application while **JavaScript** adds the functionality.

Resources:

- JavaScript Documentation (MDN Web Docs)
- JavaScript Basics
- HTML Basics
- CSS Basics

## What is Programming?

Front End - Week 1: What is Programming

Before we answer the question **What is Programming?** Let's explore **Why** we even have programming.

If we were to start a business before we had computers, we would need to keep track of everything on paper.

- All of our customer information, product or service information, every order or purchase made, every set of instructions and much more would all need to be stored and organized in some way.
- On top of simply storing all this information, we would have manual processes in place for different business actions.

**For example**: Imagine that we run a shipping or freight company and an order came in to ship a bunch of inventory from one state to another. How would we accomplish this manually?

1. Retrieve information: ***weight*** and ***size of the shipment*** and the ***required delivery date***

2. Schedule the shipment: Look at our driver's schedules, look at vehicles available, and figure out how to best satisfy the shipment.

   **Note**: There would be papers/requests/orders that change hands multiple times to get everything finalized and the entire process would take a lot of time and effort.

With computers, however, we know that we could enter values such as the ***weight***, ***size***, and ***required delivery date*** into some application and it would automatically select the best option for shipping. That automated process, that saves a lot of time, effort, and money is the result of programming.

***Programming is writing instructions for a computer that tell it how to move, manipulate, and display data in an automated fashion.***

The definition of Computer Programming in the simplest form is the process of preparing a program for a computer. When presented with a task, or a request to have a computer execute a task, it is the job of a Software Developer or Programmer to write code in a language that is understood by a computer to accomplish that task and to complete the request -- this is programming. As you progress through this course, you will be given a task or set of tasks each week, and asked to craft or "program" the solution. To do so, you will write a computer program to accomplish the requirements of that task.

Before we actually start to program a computer and learn how to write code, we need to answer the following question:

**What it takes to teach a computer to do a task?**

Everything that we see on a website or in an application is a visual representation of some type of data. The data is what drives the program or the experience. Programming is all about data.

Let's take a look at this example. Imagine a bank account. A bank account may contain money. There will be a number of interactions that a person could have with their bank account. If you were teaching a person about a bank account, you might instruct them how to do the following:

- Retrieve the balance in the account.
- Deposit money into the account.
- Withdraw money from the account.

It seems simple enough, right? The idea is to break the problem down into smaller pieces, so that we can instruct a person how to do each task.

What if you are trying to get a computer to execute these requests? The same is true with writing a program to accomplish these tasks. The goal is to break down the problem into smaller pieces to instruct the computer how to do each task.

- The computer would need to have a place to store & manipulate its **data** (**variables**).
- The computer would need to know **what** we want it to do (**tasks, methods**),
- The computer would have to be taught **how** to accomplish those tasks (**operations**).

Stay tuned, we will learn more about all of those in the future.

Here is the example that is used in this video -- notice the following sentence:

**"The cat jumped off the shelf and landed on my head"**

**Question**: How many words are there in this sentence?
**Explanation**: For a person, it seems very simple. How do we know when one word stops and another begins?

**Programming Task**: Write a program to get a computer to count the words in sentence above.
Think about what you could use to teach the computer to do something that seems simple to us.

Breaking down problems into the most specific possible details is how a computer is going to read our commands, and that's how we need to start thinking as a developer.

**Review**: Programming is writing instructions for a computer that tell it how to move, manipulate, and display data in an automated fashion.

**Digging a bit deeper -- Interesting Topics related to Programming**:

1. ***Top-Down*** *vs.* ***Bottom-Up*** *Programming and Design -- look this up in an internet search to see different approaches to programming.*
2. ***Steps to Programming****:*

    - *Decide how to solve a task or request -- remember to break the task into smaller pieces to be solved*

- *Write the code to solve the task by first writing code to solve each smaller piece of the solution*
- *Compile and/or Run the program that has been written.*
- *Test and Debug the program*
    - *Test:  Make sure the code works as designed.*
    - *Debug:  Track Down and Fix any flaws in the code that you wrote.*
- *Document exactly what has been coded and how it works.*