JavaScript Beginning to Mastery Syllabus

* JavaScript vs EcmaScript
* Install vscode and run first program

# Basics of Programming in JavaScript

* Hello world program
* Declare variable using var
* More about variable
* Let
* Const
* String Indexing
* Useful string methods
* Template Strings
* Null, undefined, BigInt, typeof
* Booleans and Comparison Operator
* Truthy and Falsy Values
* If else statement
* Ternary Operator
* && || operator
* Nested if else
* If elseif else
* Switch statement
* While loop
* While loop examples
* For loop
* For loop examples
* Break and continue keyword
* Do while loop

# Arrays in JavaScript

* Intro to arrays
* Push pop shift unshift
* Primitive vs reference data types
* Clone array & spread operator
* For loop
* use const for creating arrays
* While loop in array
* For of loop
* For in loop
* Array destructuring

# Objects in JavaScript

* Intro to objects
* Dot vs Bracket Notation
* Iterate objects
* Computed properties
* Spread operator in objects
* Object Destructuring
* Objects inside Array
* Nested Destructuring

# Functions in JavaScript

* Function declaration
* Function Expression
* Arrow Functions
* Function declarations are hoisted (covered in great detail , later in this course)
* Function inside function
* Lexical Scope
* Block Scope Vs Function Scope
* Default Parameters
* Rest Parameters
* Parameter Destructuring
* Very brief intro to callback functions(covered in great detail , later in the course)
* Functions returning Functions

# Very Important Array Methods

* Foreach method
* Map method
* Filter
* Reduce
* Sort
* Find
* Every
* Some
* Fill method
* Splice method

# More useful things

* Iterables
* Sets
* Maps
* Object.assign
* Optional chaining

# Object Oriented JavaScript / Prototypal Inheritance

* Methods
* This keyword, Window object
* Call, apply and bind method
* Some warnings
* This inside arrow functions
* Short syntax for methods
* Factory functions & discuss some memory related problems
* First solution to that problem
* Why that solution isn’t that great
* What is \_\_proto\_\_ , [[prototype]]
* What is prototype
* Use prototype
* New keyword
* Constructor function with new keyword
* More discussion about proto and prototype
* Class keyword
* Example using class keyword
* Super keyword
* Method overriding
* Getters and setters
* Static methods and properties

# How JavaScript Works

* Global Execution context
* This and window in global execution context
* Hoisting
* Are let and const are hoisted ? What is a reference Error ?
* Function execution context
* Scope chain and lexical environment
* Intro to closures
* Closure example 1
* Closure Example 2
* Closure Example 3

# DOM Tutorial

* HTML and CSS Crash course ( Around 30-40 minutes)
* Async vs defer
* Select elements using id
* querySelector
* textContent & innerText
* Change the styles of elements using js
* Get and set attributes
* Select multiple elements and loop through them
* innerHTML
* Deeply understand dom tree, root node , element nodes, text nodes
* classList
* Add new elements to page
* Create elements
* Insert adjacent elements
* Clone nodes
* More methods to add elements on page
* How to get the dimensions of the element
* Intro to events
* This keyword inside eventListener callback
* Add events on multiple elements
* Event object
* How event listener works
* Practice with events
* Create demo project
* More events
* Event bubbling
* Event Capturing
* Event delegation
* Create Project using event delegation

# Asynchronous JavaScript

* Is Javascript a synchronous or asynchronous programming language ?
* SetTimeout()
* SetTimeout() with 0 millisecond
* Callback Queue
* SetInterval and create little project with setInterval
* Understand callbacks in general
* Callbacks in asynchronous programming
* Callback Hell and Pyramid of doom
* Intro to promises
* Microtask Queue
* Function that returns promise
* Promise and settimeout
* Promise.resolve and more about then method
* Convert nested Callbacks to flat code using promises
* Intro to Ajax, HTTP Request
* XHR requests
* Error handling in XHR requests
* XHR request Chaining
* Promisifying XHR requests and chaining using then method
* Fetch API
* Error Handling in Fetch API
* Consume Promises with async and Await
* Split code into multiple files using ES6 modules.
* Congratulations
* Now you know javascript in Great Details
* What next ?