JavaScript Syllabus

Introduction

Introduction

- What is programming language?
- What is front-end?
- What is back-end?
- Introduction of HTML
- Introduction of CSS
- Introduction of JavaScript
- Role of HTML, CSS, and JavaScript?

History of JavaScript

- JavaScript history
- ECMAScript
- Versions of JavaScript

Project Setup

Visual Studio Code

- Installing visual studio code
- File vs Folder vs Workspace
- VSCode shortcuts
- Installing extensions
    - Prettier
    - Live Server
    - Monokai Pro
- Applying Settings
- Applying Color Theme: Monokai Pro
- Default Formatter
- Prettier.rc file and it’s configurations

Project setup

- Separation of concern
- Creating index.html
- Linking JavaScript file

JavaScript Core

JavaScript Features

- High Level Language
- Garbage Collected
- Interpreted Language (JIT)
- Multi Paradigm
- Prototype based functions
- First Class Function
- Dynamically Type/ Dynamic
- Single Threaded
- Non-Blocking Event Loop

Value, Variable and Data types

- What is value
- What is variable
- What is data type
- Different types of data types
  - Difference between primitive and non-primitive data types
  - Primitive Data types (In built data types)
    - Number
    - String
    - Undefined
    - Boolean
    - Symbol (new in ECMAScript 2015)
    - BigInt (new in ECMAScript 2020)
  - Non-primitive Data types
    - Object
    - Array

Identifiers

- What is identifier
- Rules for creating identifier

Comments

- What is comment
- Single line comment
- Multi line comment
- Comment rules

use strict

- What is the significance of use strict
- Without use strict

Statement

- What is a statement in programming
- How to write a single line of statement
- How to write a multi-line statement
- Semi colon in statement
- Whitespace in a statement
- What is a code block

let, const and var

- let
- const
- var
- Difference between let, const and var
- When to use let, const and var

JavaScript operators

- Assignment operator
- Arithmetic operator
- Comparison operator
- Logical operators
- Type operators
- Operator precedence
- Truth table of &&, || and !

Conditional statements

- if
- else if
- else
- Grouping multiple conditions using logical operator

JavaScript Output

- console.log
- document.write()
- window.alert()
- innerHTML

JavaScript String

- What is a string
- Uses of single quote and double quotes in string
- How to create a String
- String Literal
- String Object
- String Literal vs String Object
- String length
- String to Array
- String Template Literal
- String functions
  - slice
  - substring
  - substr
  - replace
  - repeat
  - toUpperCase
  - toLowerCase
  - concat
  - trim
  - padStart
  - padEnd

    - charAt
    - split
    - indexOf
    - lastIndexOf
    - startsWith
    - endsWith
    - search
    - match
    - includes

Type Conversion

- Implicit type conversion
- Explicit type conversion
- Automatic Type conversion (Coercion)
- Manual Type conversion
    - Number
    - String
    - Boolean

JavaScript Popup Boxes

- Alert Box
- Confirm Box
- Prompt Box

Truthy and Falsy Values

- What are the truthy and falsy values in JavaScript
- Falsy values
    - undefined, 0, null, ‘’, false, NaN
- Falsy and Truthy values in conditional statements

Other Operators

- Loose equality operator
- Strict equality operator
- Typeof operator
- Ternary operator

Looping and Switch

- For Loop
- While Loop
- Do while loop
- Loop inside loop
- Backwards Loop
- For of loop
- For in loop
- Switch
    - Cases in switch
    - Default case
    - Break

- Break and continue

Scope

- Scoping
- Different types of scopes in JavaScript
    - Global Scope
    - Functional scope
    - Block scope

Functions

- Function declaration
- Function expression
- Arrow function
- Difference between function declaration and function expression
- Difference between function expression and arrow function
- Anonymous function
- Function invoking/calling
- Function calling from other function
- Function as values
- Parameters
- Arguments
- Arguments Object in functions

More on functions

- Default parameters
- Passing arguments: value vs reference
- First Class function/Citizen
- High Order function
- Callback function
- setTimeOut
- setInterval
- Function returning function
- The call and apply methods
- The bind method
- Immediately invoked function expression
- Closures

Hoisting

Temporal Dead Zone

DRY Principle

Debugging

Debugging

- Overview of Google chrome developer tools
- Debugging points, adding a breakpoint
- Fixing errors
    - console.log
    - console.warn
    - console.error
    - console.table
- How to fix a bug, different steps:
    - Identifying bug
    - finding bug
    - fixing bug
    - Not repeat bugs
- Different type of errors
    - Syntax Error
    - Reference Error
    - Type Error
    - Other Errors
        - Eval Error
        - Internal Error
        - Range Error
        - URI Error

Numbers and Dates

Number

- Converting numbers
- NaN
- Infinity
- Number System
    - Binary
    - Octal
    - Decimal
    - HexaDecimal
- Checking numbers
- Hoisting in numbers
- Math and Rounding
- The Reminder operator
- Numeric Separators
- Working with BigInt
    - Exceptions in BigInt
- Number class functions
    - toFixed
    - toString
    - valueOf
    - Number()
    - parseInt
    - parseFloat

    - isNaN
- Number Properties
    - MAX_VALUE
    - MIN_VALUE
    - POSITIVE_INFINITY
    - NEGATIVE_INFINITY

Date

- Creating Dates and different ways of creating Date object
- Understanding milliseconds and other units of time
- Operations with Dates
    - Date setter methods
    - Date getter methods
- Internationalization Dates
- Internationalization Numbers
- setTimeOut and setInterval

JavaScript DOM and BOM

- DOM (Document Object Model)
    - Introduction
    - DOM functions
        - getElementById
        - getElementsByTagName
        - getElementsByClassName
        - querySelector
        - querySelectorAll
        - write()
    - Properties
        - innerHTML
        - attribute
        - style.property
        - textContent
    - Forms
        - Forms validation
        - Properties
            - Disabled
            - Max
            - Min
            - Pattern
            - Required
    - Type of Events
        - Onclick
        - Onchange
    - Mouse events
        - Onmousedown
        - Onmouseup
    - Event Listener

        - addEventListener
    - Navigation
        - parentNode
        - childNodes
        - firstChild
        - lastChild
        - nextSibling
        - previousSibling
    - DOM Nodes
        - createElement
        - createTextNode
        - appendChild

- JavaScript BOM
    - Window object
    - History object
    - Navigator Object
    - Screen Object
    - Location Object
    - Timing
    - Cookies
    - LocalStorage

JavaScript Behind The Scene

- JavaScript behind the scene
    - JavaScript Engine
    - Call Stack
    - Execution Context
    - Memory/Heap
    - Compiler
    - Interpreter
    - Compiler Vs Interpreter
    - Event Loop
- Execution Context consists of 3 things:
    - Variable Environment
        - let, const and var declarations
        - functions
        - Arguments Objects
    - Scope Chain
    - this keyword
- Execution Context divides in two parts
    - Type of execution context
        - Global
        - Functional
    - Creation Phase
    - Code Phase
- Scope Chain:
    - Scoping: How our programs variables are organized and accessed

    - 3 types:
        - Global Scope
    - Local/Function Scope
    - Block Scope
- this key word
    - this in global scope
    - this in function
    - this in object
    - this in arrow function
    - this in inside function inside object
- Primitive vs Object
    - Understanding of how primitive and non-primitives are stored in memory
    - Copying object
    - Copy first level properties
        - Shallow copy
        - Deep copy

Modern Features

- **Destructuring Arrays**
    - What is destructuring
    - Reverse values using destructuring
    - Return two values from function
    - Destructuring of nested array
    - Setting default values
- **Destructuring Objects**
    - Extract value
    - Different property name
    - Default values
    - Nested Object
    - In Function
- **The Spread Operator**
    - Assigning values
    - Copy Array
    - Join 2 Arrays
    - String to array using spread
    - Passing arguments in function
    - Shallow copy
- **The Rest Parameter**
    - Assign values
    - Rest element last element
    - Assign values in object
    - Variable arguments in function
- **Short Circuiting**
    - Use of ||
    - Replace with ternary operator
    - With non nullish values
    - Use of &&
    - Calling function using &&
- The Nullish Coalescing Operator ??

- **Logical Assignment Operator**
  - ||=
  - &&=
  - ??=
- **Enhanced Object literals**
  - Exactly same name
  - Function in object
  - Computer property name
- **Optional Chaining**
  - Multiple condition in if condition
  - Work for nullish
  - Checking if method exist
  - Checking array is empty

Array

- What is an Array
- Need of Array
- How to create an Array
  - Array Literal
  - Array Object
- Index in Array
- Array length property
- Array Declaration
- Looping Array
- Array functions
  - sort
  - push
  - pop
  - unshift
  - shift
  - toString
  - join
  - concat
  - splice
  - slice
  - sort
  - reverse
  - forEach
  - at
  - map
  - filter
  - reduce
  - find
  - findIndex
  - some
  - every
  - flat

    - flatMap

Object

- What is an object
- Object literal syntax
- Object creation using new keyword
- Annotation
    - Dot
    - Bracket
- Object properties
    - Key
    - Value
    - Array in Object
    - Function in Object
    - Uses of this in Object
- Object methods
    - Keys
    - Values
    - Entries

Set

- What is a Set
- Creating set
- Elements order in Set
- Set size
- Set.has function
- Set.values function
- Set.delete function
- Index in set
- Printing set values using for of loop
- Creating set to array, different ways
- forEach method
- Adding object in set

Map

- What is a Map
- Creating new map
- Adding value in map
- Chaining in map
- .get function
- .has function
- .size function
- .clear function
- Array as key
- Iteration of Map
- Object to map
- Map to array
- forEach function on map

JavaScript OOPs

OOPs

- OOP in JavaScript
- OOPs fundamental concepts:
  - Object
  - Class
  - Encapsulation
  - Abstraction
  - Inheritance
  - Polymorphism
- Constructor functions and new operator
- Prototypes
- Prototypal inheritance and prototype chain
- Prototypal inheritance on Built-in objects
- ES6 classes
- Setters and Getters
- Static methods
- Object.create
- Inheritance between classes
  - Using constructor functions
  - Using ES6 classes
  - Using object.create
- Encapsulation: Protected Properties and Methods
- Encapsulation: Private Class Fields and Methods
- Chaining methods

Asynchronous JavaScript

Asynchronous JavaScript

- Ajax
- What is an API
- XMLHttpRequest
- How the web works
  - Server
  - Client
  - Request
  - Response
- Callback
- Promise and Fetch API
- Consuming Promises
- Chaining Promises
- Handling Rejected Promises
- Asynchronous Behind the Scene: The Event Loop
- Building a Simple Promise
- Consuming Promise with Async/Await

- Error Handling with Try catch
- Returning values from Async functions
- Running promises in Parallel
- Promise Combinators: race, allSettled and any

Modern JavaScript Development

- An Overview of Modern JavaScript Development
- An Overview of Modules in JavaScript
- Exporting and importing in ES6 Modules
- Top-Level await (ES2022)
- The Module Pattern
- Bundling With Parcel and NPM Scripts
- Configuring Babel and Polyfilling
- Transpiling
- Transpiling vs Polyfilling