**Javascript Handbook**

**History:**

* Javascript is a scripting language that was first developed in 1995 by Brendan Eich for making websites at the Netscape navigator browser interactive. It was initially named mocha but associating its name with then popular language Java made it more popular. Although Java and Javascript are very different.
* Soon after that Jscript was launched by Internet explorer with features of javascript along with additional features. But the problem arose when there was no standardization due to which, in order to make interactive websites interactive one had to learn particular scripting language for that browser.
* So, in order to standardize scripting languages, ECMA international generated guidelines to be followed by all scripting languages so that they could be used across multiple browsers without thinking of creating websites in different languages for different browsers.
* ECMA launched ES1(ECMAScript1) in 1997. All scripting languages could then be regarded as the versions providing same functionalities but having different syntax. So even if we use Javascript in reference of ECMAScript its one and the same thing.
* The boost came in 2015 when ES6 was updated with the modern scripting functionalities and after that the TC39 (Technical Community 39) of ECMA International decided to launch new versions every year and the versions would be thereafter be designated by the year: ES2015, ES2016, so on.
* Since Javascript had to follow the standards and was already popular, it gained more users thus highlighting Javascript among all the other scripting languages.

**Features of JavaScript:**

* JavaScript is backward compatible, which means that it can support the older commands in the updated browsers.
* JavaScript is not forward compatible, which means you cannot use latest commands if your browser is not updated. However tools like Babel can be used that transpile the new commands into older ones that can run on outdated versions of browsers.
* JS1.js contains:

Commit1:

printing in console, declaring variables using var, let (things related to their usage), declaring constants, string indexing and string length

commit2:

string methods like trim, toUpperCase, toLowerCase, string slicing, list of primitive datatypes, typeof, conversion of string to number and vice-versa, string concatenation, template strings, undefined datatype

commit 3:

Null datatype, max safe integer, big int , Boolean datatypes, comparision operators, if else, truthy and falsy values, ternary operator, &&, || operators

Commit 4:

Prompt(), nested if-else, else if ladder, switch statement

* JS2.js contains:

Commit 4:

While loop, for loop, scope of variable using let and var declarations, break, continue, do while loop

Commit 5:

Arrays, push , pop, shift, unshift, primitive vs reference types, array cloning methods, creating arrays out of already existing arrays with additional items, array concatenation

Commit 6:

For loop in array, using const to create array, while loop in array

Commit 7:

For in loop, for of loop, array destructuring

* JS3.js contains:

Commit 7:

Object, methods of accessing data in objects, adding a key value pair in object, difference between dot and bracket notation and uses of bracket notation

Commit 8:

Iterating object using for in loop, Object.keys(), computed properties, spread operator in objects.

Commit 9:

Object destructuring, objects inside array, nested destructuring

Commit 10:

Arrow functions, hoisting, nested functions, lexical scope, block scope vs function scope

Commit 11:

Default parameters, rest parameters, parameter destructuring, callback functions, functions returning functions

* JS5.js contains:

Commit 12:

Array methods - forEach(), map(), filter(), reduce()

Commit 13:

Array methods – sort(), find(), every(), some(), fill(), splice()

* JS6.js contains:

Commit 14:

Sets, has(), for of loop in sets, finding length of set, map objects, map iterator

Commit 15:

set(), get(), keys(), adding key value pair using arrays, extending key value pair in an object using map, Object.assign(), optional chaining

Commit 16:

Methods, this keyword, global object

Commit 17:

call(), arrow function and this, apply(), bind(), short syntax alternate for methods

* JS7.js contains:

Commit 18:

Using function to create multiple objects (constructor function), store methods in different object, Object.create(), \_\_proto\_\_

Commit 19:

Functions as objects, prototype, new keyword, hasOwnProperty()

Commit 20:

Prototype in array, Object.getPrototypeOf(), Array.prototype, changing function prototype from object to array, classes, extends ,inheritance

Commit 21:

Super (), same name method in base class, getters and setters, static methods and properties

Commit 22:

Created JS8.js

* JS8.js contains:

Commit 23:

How JS works, how JS executes code, Global execution context

Commit 24:

Function declarations and GEC, hoisting explained, function expression and GEC, let and const and GEC, temporal dead zone

Commit 25:

Function execution context, snippet analysis for understanding

Commit 26:

Scope chain, closures

Commit 27:

Application of closure

* JS9.js contains:

Commit 28: