**FINAL ASSIGNMENT**

Name: Mineshkumar Dayalbhai Tandel

Student ID: 2110050

**SOLOLEARN JAVASCRIPT COURSE**



Sololearn JAVASCRIPT course consists of below mentioned concepts:

1. Overview
2. Basic Concepts
3. Conditional and Loops
4. Functions
5. Objects
6. Core Objects
7. DOM and Event
8. ECMAScript 6
9. **Overview:**

Overview section of the course is consists of:

**a). Data Types:** types of information in various forms such as integer, string, boolean; stored in a variable for data manipulation purpose.

**b). Comments:** A comment is a text note added to source code to provide explanatory information,

A single line comment - //

A multiple line comment - /\* \*/

**c). Variables:** Variable is just a name associated with values of various data types. It is used for storing data values. ​Javascript has **var, const and let** keywords to initiate variable name

Conventions to be followed while declaration of variables.

1. **Basic Concepts:**

Basic concepts such as:

**a). Math Operators:** Arithmetic Operators, such as **+, -, \*, /, %, ++, –** are applied to number type operands

**b). Assignment Operators:** Assignment Operators such as **=, +=, -=, %=, \*=, /=** are used to assign values to declared variables.

**c). Comparison Operators:** When it comes to compare two data types of the same kind, comparison operators are used to find out if the variables or values are different. The output is either true or false. ==, ===, !=, !==, >, >=, <, <=.

**d). Logical or Boolean Operators:** Logical Operators, like Comparison operators, return a true (1) or false (0) value when processed. Logical operators combine two comparisons and return the true (1) or false (0) value depending on the results of the comparisons. Logical operators are &&, ||, !.

**e).String Operators:** String operator ‘+’ is used to concat two strings.

1. **Conditional and Loops**:

**Conditional statements such as:**

**a). The if statement:** The block of code within the curly bracket only gets executed if the condition in parentheses is evaluated to be true.

Syntax: **if(condition) {block of code}**

**b). The else statement:** Execution of Second block of code will be carried out which is, if the specified condition is false.

Syntax: **if(condition) {**

**First block of code}**

**else {**

**Second block of code**

**}**

**c). The else if statement:** If we have different conditions, based upon which expected results are achieved using the else if statement. Note: only one block of code will be executed depending upon condition validation.

Syntax: **if(condition1) {**

**First block of code}**

**else if (condition2) {**

**Second block of code}**

**else {**

**Third block of code}**

**d). The switch statement:** This is used to perform different actions based on the different conditions.

**Loops such as:**

**e). The for loop:** for(statement1; statement2; statement3), where, statement1 is optional and it sets a variable before the loop starts.

Statement2 defines the condition of the loop to be run.

Statement3 increases a value each time a code block in the loop is executed.

**f). The while loop:** It repeats through the block of code, as long the condition is true.

**g). The do-while loop:** The loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

**h) Break and continue statement:** The break is used to jump out of the loop and continue executing the code after the loop, while the continue statement breaks only one iteration and continues with the next iteration.

1. **Functions:** It is used to avoid code multiplications and it enables us to use a single code as many times as needed.

**a). Function definition:**

Syntax: **function name() { block of code}**

To define a JavaScript function, use the **function** keyword, followed by a **name**, followed by a set of **parentheses ()** and code to be executed lies inside **brackets {}.**

**b). Function parameters:** Multiple parameters can be used with commas in between them as:

Syntax: **function name(param1, param2, param3){ block of code}.**

**Other inbuilt functions:**

**a). Alert function:** It takes a single parameter, which is the text displayed in the pop-up box.

Syntax: **alert(‘Message’)**

**b). Prompt function:** Prompt is used to get input from a user after displaying a message.

Syntax: **var Name = prompt(‘Enter your name: ‘)**

Entered input by the user will be stored in the **Name** variable and it will display a message inside the parenthesis.

**c). Confirm function:** Confirm function renders two options, “OK” and “Cancel”, returns true - if the user clicks ‘Ok’ and it returns false - if the user clicks Cancel.

1. **Objects:** Objects are key:value pair

**a). Object-Initialization:**

**Syntax : const person = {**

**Name: ‘Java’,**

**Age: 25**

**}**

Using Function we can create objects having different Value:

**Function person(firstName, lastName, age) {**

**this.firstName = firstName**

**}**

**Const Person1 = new Person(‘Kaka’, ‘Maha’, 34)**

1. **Core Objects:**

**a). Array:** It is an object which stores multiple values in a single variable. It can be accessed by index number within square brackets. The array has a length property and concat() method.

b). **Math Object:** It allows to perform several math tasks, and has different properties like, Math.E, Math.PI.

c). **Date Object:** It enables us to work with dates. JavaScript counts months from 0 to 11, January as 0 and December as 11.

1. **DOM:** Dom is an interface which can be used by Javascript to manipulate document

It is a tree-like structure, which has parent and child structures.

1. **ECMASCRIPT 6:**

To standardize JavaScript, **ECMAScript** (ES) is created, which is a scripting language.

The Sixth Edition, initially known as **ECMAScript 6** (ES6) and later renamed to **ECMAScript 2015**, adds significant new syntax for writing complex applications, including classes and modules, iterators and for/of loops, generators, arrow functions, binary data, typed arrays, collections (maps, sets and weak maps), promises, number and math enhancements, reflection, and proxies.

**a). ES6 variables and strings:** We can declare variables in three different ways in ES6.

var name = ‘Minesh’;

let age = 25;

const height = 173

**var :** var keyword defines a variable globally or locally to entire function disregarding

entire block scope

**let :** let is used to declare variable limited to block scope, expression and statements only

**const :** const has the same scope as let and with const it does not allow reassignment

**Template Literal :** Template literals are a way to output variables in the string using

backtick character.

Var name = ‘Minesh’

Syntax: `Hello! ${name}`

**b). Loops and Functions in ES6:**

**for … of loop :** until ES5 use of **for….in** loop is used but due to integer concatenation

instead of addition has raised an issue, which brought up for… of loop

Syntax : **for (let i of array) { body block}**

**Function: ES6** introduced arrow functions and default parameters

Syntax : **const add = (parameters) => { body block of the function}**

**c). Rest parameter and spread operator:**

**… -> spread operator**

**…nums -> Rest Parameter**

**d). Destructuring:**

**Array:** const arr = [1, 2, 3];

Let [one, two, three] = arr

console.log(one) // 1

**e). Map and Set:**

**Map:** A Map object can be used to hold key/value pairs. A key or value in a map can

be anything (objects and primitive values).

**let map = new Map([[‘name’, ‘Minesh’], [‘age’, ‘25’]])**

**set(key, value):**  Adds a specified key/value pair to the map. If the specified key already exists, value corresponding to it is replaced with the specified value.

**map.set(‘name’, ‘Gimple Elliot’)**

**GIT REPOSITORY LINK - https://github.com/Minesh6684/Javascript/tree/main/MIDTERM**