JavaScript

JavaScript: JavaScript is a programming language that's used to give instructions to a computer and display the output.

Input (code or set of instructions) → **Computer** → Display the Output.

Note: JavaScript code extension: hello.js

```
//1.print with console.log();
console.log("Hello");// with double qoutes
console.log('Hello');// with single qoutes
```

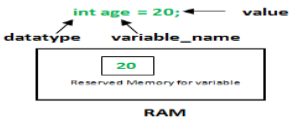
Print with console: console.log() is used to log(print) a message to the console.

- **Syntax: console.log ();** → **console.log("Hello world!");**
  - I. **console:** A built-in JavaScript object that provides access to the browser's debugging console.
  - II. **`Log()`:** A method of the `console` object, that's used to print messages to the console, for **testing purposes.**

```
//1. print with console.log(): --> for testing purpose
console.log("Hello JavaScript");// with double qoutes
console.log('Hello JavaScript');// with double qoutes
console.log("Mohammad Al-Amin"); // with double qoutes
console.assert.log('Mohammad Al-Amin'); // with double qoutes
```

Variable and Data Types

Variables: Variables are Containers for Storing Data, that can be declared in 4 ways:



Note: JavaScript is dynamically typed language

1. **Automatically: Global Scope** (can be re-declare and update)

```
// 1.declare Variables with automatically: Global Scope (can be re-decalre and update)
userName ="Mohammad Al-Amin";
userId=101;
userAddress="Dhaka, bangladesh";
userConatact="01745157083";
userDegree="B.Sc in CSE";
userSalary=50000.00;
console.log("userName: ",userName);
console.log("userId: ",userId);
console.log("userAddress: ",userAddress);
console.log("userConatact: ",userConatact);
console.log("userDegree: ",userDegree);
console.log("userSalary: ",userSalary);
userName ="Mohammad Yasin"; // re-declare is possible
console.log("userName: ",userName);
userId=102; //update
console.log("userId: ",userId);
```

2. **Using-var: Global Scope** (can be re-declare and update).

```
// 2.declare Variables with keyword - var: Global Scope (can be re-decalre and update)

var Number1=100;
var Number2=100.112131323;
console.log("Number1: ",Number1);
console.log("Number2: ",Number2);

var Number1=200; //re-decalre
var Number2=200.222131323; //re-decalre
console.log("Number1: ",Number1);
console.log("Number2: ",Number2);

Number1=300; //update
Number2=300.332131323; //update
console.log("Number1: ",Number1);
console.log("Number2: ",Number2);
```

3. **Using let: Block Scope** (can't be re-declare but it can be update).

```
// 3.declare Variables with keyword - let: Block Scope (can't be re-decalre but it
canbe  update)
let Number3=400;
let Number4=500.552131323;
console.log("Number3: ",Number3);
console.log("Number4: ",Number4);

// let Number3=400; //cant' be re-derclare
// let Number4=500.552131323; //cant' be re-derclare
// console.log("Number3: ",Number3);
// console.log("Number4: ",Number4.toFixed(2));

Number3=600; //update
Number4=700.772131323; //update
console.log("Number3: ",Number3);
console.log("Number4: ",Number4);
```

4. **Using const: Block Scope** (can't be re-declare and update).

```
// 4.declare Variables with keyword - const: Block Scope (can't be re-decalre and update)
const Number5=800;
const Number6=900.992131323;
console.log("Number5: ",Number5);
console.log("Number6: ",Number6);

// const Number5=800;//cant' be re-derclare
// const Number6=900.992131323;//cant' be re-derclare
// console.log("Number5: ",Number5);
// console.log("Number6: ",Number6);

// Number5=1000; // can't be update
// Number6=1000.1992131323; // can't be update
// console.log("Number5: ",Number5);
// console.log("Number6: ",Number6);
```

Variable naming Rules:

1) Variable names are case sensitive; "a" & "A" is different.

```
//1. Variable names are case sensitive; "a" & "A" is different.
let fullName="Tahsen Ahmed Al-Amin";
let FullName="Mohammad Imran";
console.log("Full Name: ",fullName );
console.log("Full Name: ",FullName );
```

2) Only letters (a-z / A-Z), digits (1…), underscore(_) and $ is allowed. (not even space).

```
//2. Only letters (a-z / A-Z), digits (1…), underscore( _ ) and $ is allowed. (not even space)
let numbeR=1000.12323;
console.log("Number: ", numbeR);
let number200=2000.12323;
console.log("Number: ", number200);
let _number=3000.3123312;
console.log("Number: ", _number);
let $number=4000.3243458947;
console.log("Number: ", $number);
```

3) Only a letter, underscore (_) or $ should be 1st character.

4) Reserved words and digit cannot be variable names.

```

// 3)Reserved words and  digit  cannot be variable names.
// 12number=5000.3432434; // digit can't be the 1st character
// console.log("Number: ", 12number);
// let new=15000.3432434; // digit can't be the 1st character
// console.log("Number: ", new);
```

5) Variable names can't contain spaces.

```
// 4). Variable names can't contain spaces.
// let full name="Mohammad Yasin- Imran"; //cant't contain space
// console.log("Name: ", full name );
```

Different type of variables name cases:

1) **fullName**: Camel Case (Generally max. time use it)
2) **fullname**: Lowercase
3) **FULLNAME**: Uppercase (or CONSTANT Case)
4) **full_name**: Snake Case
5) **full-name**: Kebab Case (or Hyphen Case)
6) **FullName**: Pascal Case

Datatypes: 1. primitive (7 types), 2. non-primitive (objects→ Arrays, Functions etc.)

1. primitive-data types: number, string, Boolean, undefined, null, symbol, BigInt

```
// 1. primitive-data types: string, number, boolean, undefined, null,  Bigint,symbol
let str="Amin";//string
let num1=100.12331; //number
let bool=true; // boolean
let num2; //undefinde
let num3=null; //null--> it's a object, null means absence of a objects
let Bigint=BigInt("123"); // BinInt() ->>1 to n
let symbol=Symbol("AlAmin95");
console.log("str: ", str, typeof(str));
console.log("num1: ",num1, typeof(num1));
console.log("bool: ",bool, typeof(bool));
console.log("num2: ",num2, typeof(num2));
console.log("num3: ",num3, typeof(num3));
console.log("Bigint: ", Bigint, typeof(Bigint));
console.log("symbol:  ", symbol, typeof(symbol));
```

2. non-primitive data types: (objects→ Arrays, Functions etc.)

- **Objects: it's a collection of values , that is the key value pairs.**

Example: create object name: students→ name; string, age: number; marks: numbers, isPass: Boolean.

```
const student={ // object
    fullName: "Al-Amin", //key
    age:20,
    cgpa: 3.09,
    isPAss: true
}
console.log(student, typeof(student))
console.log(student["fullName"], student['age']); // access individual keys of student
console.log(student.fullName, student.age, student.cgpa); // access individual keys of student
// update name and age
student.fullName="Tahsin";
student.age=25;
console.log(student["fullName"], student['age']); // access individual keys of student
```