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
```

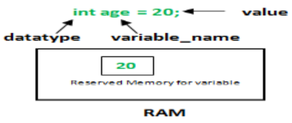
Print with console: console.log() is used to log(print) a message to the console.

- **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-declare 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
```

JavaScript Comment: **Part of code which is not executed.**

```
//Single line comment

/*
1. Multiple line comment
2.
3.
4.
…
*/
```

Operator and Conditional Statements

Operator: JavaScript operators are used to perform different types of mathematical and logical computations.

Types: 1. Arithmetic Operators, 2. Assignment Operators, 3. Comparison Operators, 4. String Operators**, 5.** Logical Operators 6. Bitwise Operators 7. Ternary Operators 8. Type Operators

1. Arithmetic Operators: a. summation (+), Subtraction (-), Multiplication (*), Division (/), Modulus (%), Exponentiation (**)
b. Unary Operator: Increment (++), Decrement (--)

```
// a. summation(+),Substraction (-), Multiplication (*), Division(/), Modulus(%), Exponentiation(**),

let a=5,b=2;

let sum=a+b;

let sub=a-b;

let mul=a*b;

let div=a/b;

let rem=a%b;

let exp=a**b;

console.log("a = ",a," b=",b);

console.log("a + b = ",a,"+",b,"=",sum);

console.log("a - b = ",a,"-",b,"=",sub);

console.log("a * b = ",a,"*",b,"=",mul);

console.log("a / b = ",a,"/",b,"=",div);

console.log("a % b = ",a,"%",b,"=",rem);

console.log("a ** b = ",a,"**",b,"=",exp);

// b. Unary Operator

console.log("\n\n a= ",a); // first done increment then print

console.log("Pre-Increment: ",++a); // first done increment then print

console.log("Post-Increment: ",a++); // first print then  done increment

console.log("Post-Increment: ",a); // first print then  done increment

console.log("\nPre-Decrement: ",--a); // first done decrement then print

console.log("Post-Decrement: ",a--); // first print then  done  decrement

console.log("Post-Decrement: ",a); // first print then  done decrement
```

2. **Assignment Operators:** Assignment operators assign values to JavaScript variable,

- ➢ (=, +=,-=, *=, /=, %=, **=)

```
// 2. Assignment Operator: (=,+=,-+, *=,/=, %=, **=)
let num1=5; // assign the value to variable:  num1=10, use the ooperator: =
num1+=4; //
console.log("Summation: ", num1);
num1-=2;
console.log("Subtraction: ", num1);
num1*=4;
console.log("Multiplication: ", num1);
let num2=6
num2/=3;
console.log("Division: ", num2);
let num3=7;
num3%=2;
console.log("Remainder: ", num3);
num1=5; //update
num1**=2;
console.log("Exponential: ", num1);
```

3. **Comparison Operator:** it compares between two values.

- ➢ Equal to (==)→**check only value**, Equal to & type (===)→ **check value and data types**, Not equal to (! =), Not equal to &type (! ==), greater than (>), Less than(<), Greater than or equal(>=), Less than or equal(<=)

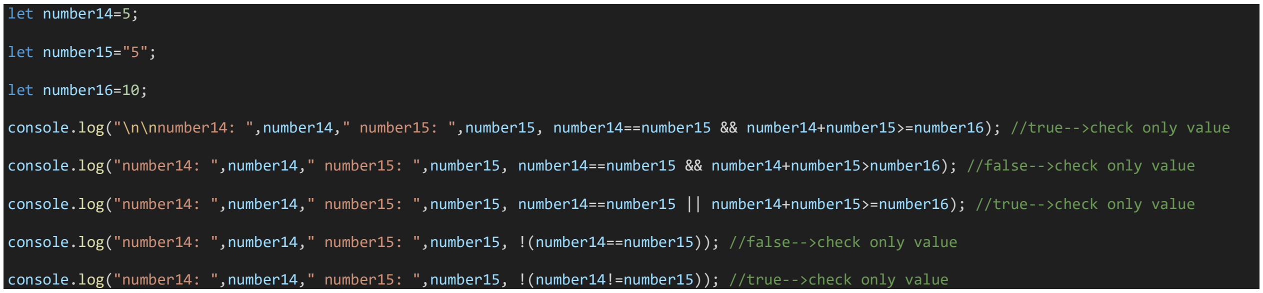
```
let number10=5;
let number11="5";
console.log("\n\nnumber10:",number10," number11:",number11, number10==number11);// true--> check only value
console.log("number10:",number10," number11:",number11, number10!=number11);// false--> check only value

let number12=5;
let number13= "5";
console.log("\n\nnumber12:",number12," number13:",number12, number12===number13);// true--> check value and data types
console.log("number12:",number12," number13:",number12, number12 !==number13);// false--> check value and data types

console.log("\n\nnumber12:",number12," number13:",number12, number12 >number13);// false--> check value
console.log("number12:",number12," number13:",number12, number12 >=number13);// true--> check value
console.log("number12:",number12," number13:",number12, number12 <number13);// false--> check value
console.log("number12:",number12," number13:",number12, number12 <=number13);//  true--> check value
```

4. Logical Operators: Compares between two conditions.

- **logical AND (&&):** if two conditions are true then return true otherwise return false.
- **logical OR (||):** if two conditions are false then return false otherwise return true.
- **logical NOT (!):** it’s made false value to true vice-versa.



```
let number14=5;

let number15="5";

let number16=10;

console.log("\n\nnumber14: ",number14," number15: ",number15, number14==number15 && number14+number15>=number16); //true-->check only value

console.log("number14: ",number14," number15: ",number15, number14==number15 && number14+number15>number16); //false-->check only value

console.log("number14: ",number14," number15: ",number15, number14==number15 || number14+number15>=number16); //true-->check only value

console.log("number14: ",number14," number15: ",number15, !(number14==number15)); //false-->check only value

console.log("number14: ",number14," number15: ",number15, !(number14!=number15)); //true-->check only value
```

5. Ternary Operator:

Syntax: condition? True output: false output

Conditional Statements: It’s controlled the flow of code based on conditions.

1. **if**: Executes a block of code if the specified condition is true.
2. **else**: Executes a block of code if the same condition is false.
3. **else if**: Tests a new condition if the first condition is false.
4. **switch**: Allows you to specify multiple alternative blocks of code to be executed based on different conditions.

```
Syntax:
if (test expression)
    {// code
    }
else if (test expression)
    {// code-- depend on if
    }
else
    {// code-- depend on if
    }
Note:
1. if can run on its own because it checks a condition and executes code if the condition is true.
3. else and else if cannot run without an if because they depend on the if condition to determine what to do if the if condition is false.
```

Example:

```
let number17 = 5;
let number18 = "4";
let number19 = 3;
if (number17 === number19) {
    console.log("number17 is equal to number19.");
} else if (number17 == number18) {
    console.log("number17 is equal to number18, but their types are different.");
} else {
    console.log("number17 is not equal to number18 or number19.");
}
```

Switch Statement: