GYANSABHA
For Better Future

Introduction

JavaScript Programming

Introduction

What is JavaScript?

- It is a verb of the web page that defines all the actions to be performed on a web page.
- An object-oriented programming language that uses a JIT compiler.
- It's everywhere and installed in every web browser.
- JS coverage such as web development, mobile development.
- JS is simple, easy to use, and highly compatible with HTML and CSS.
- JS is used to create client-side dynamic pages.

History

- The originator of JavaScript was Brendan Eich.
- he developed in 1995.
- JavaScript was later renamed to 'mocha' and 'livescript', but he remains JS for trademark reasons.
- He also developed the first his JS engine, Spider Monkey, which is still used in Mozilla Firefox today.

Features

- All major web browsers support JavaScript as they provide a built-in execution environment.
- JavaScript is a weakly typed language, where certain types (depending on the operation) are implicitly cast.
- JavaScript is an object-oriented programming language that uses prototypes instead of classes for inheritance.
- It is a lightweight, interpreted language.
- The language is case-sensitive.
- JavaScript is supported on multiple operating systems including Windows, macOS, and more. This gives the user good control over her web browser.

Applications

- client-side validation,
- dynamic dropdown menu,
- date and time display,
- Display of pop-up windows and dialog boxes (warning dialog boxes, confirmation dialog boxes, prompt dialog boxes, etc.);
- Display the clock, etc.

Tool

You only need two things to work with JavaScript

A. A text editor

- You need a simple editor to write and edit JS code. This can also be Notepad.
- However, there are other powerful editors that offer additional features such as autocomplete, indentation and highlighting.
- Examples: Visual Studio Code, Sublime Text, Atom, etc.

B. Browser

- All browsers have a built-in JS engine.
- These are used for the purpose of running and debugging JS code

Variable

A JavaScript variable is just a location name. JavaScript has two types of variables: local variables and global variables. Variable is a name of a memory location where the data is stored.

Syntax:

var var_name = val;

In JavaScript, variable is define using var keyword.

1. var name = “Anandrao”;
2. var roll_no = 27;

Example:

var name=”Anandrao”;

console.log(name);

There are two types of variables:

1. local
2. global

1. **local:** It is declared inside block or function. It is accessible within the function or block only.
   Example:
   ```
   function local()
   {
       var x=10; //local variable
   }
   ```

2. **global:** It is accessible from any function. A variable i.e., declared outside the function or declared with window object is know a global variable.
   Example:
   ```
   Var data=200;        //global variable
   Function a()
   {
   ```

```
        Console.log(data);
}
a();
```

Data Type

JavaScript is a dynamically typed language and does not need to specify data type explicitly of the variable.

There are 2 types of data type:

1. **Primitive data type**

   There are 5 primitive data type in JavaScript

   - **Number:**
     Number is represents integer and floating numbers.
     Example:
     const a = 4;
     const b = 4.1;
     console.log(a,"\n",b);

     Output:
     4
     4.1

   - **String:**
     String is used to store text. The two way to declare the string.
     - Double: “Hello”
     - Single: ‘Hello’

     Example:
     const a="Gyansabha";
     const b='For Better future';

     console.log(a,"\n",b);

     Output:
     Gyansabha
     For Better future

   - **Boolean:**
     Boolean represents either true or false.
     Example:
     const a= true;

```
const b=false;

console.log(a,"\n",b);
```

Output:
true
false

- **Undefined:**
  Represent the undefined value. A variable has an undefined value if it is declared but not assigned a value.

  Example:
  ```
  let a;

  console.log(a);
  ```

  Output:
  Undefined

- **Null:**
  Represents null i.e. no value at all

  Example:
  ```
  const a=null;

  console.log(a);
  ```

  Output:
  Null

2. **Non-primitive data type**

   There are 3 type of non-primitive data type.

   - **Object**

     Represents an instance that has access to members.

   - **Array**

Represents a group of similar values

- **RegExp**
  Represents regular expression

Operators

In JavaScript, operators are special symbols used to perform operations on their operands. For example:

4+10; //14

Var a=4+10; //a=14

Here, + is the arithmetic operator and = is the assignment operator.

There are following type of operators

1. Arithmetic Operators
2. Comparison Operators
3. Assignment Operators
4. Logical Operators
5. Bitwise Operators
6. Other Operators

1. Arithmetic Operators:

It is perform the mathematical operations (like addition, subtraction, multiplication, division, etc.).

Operator	Meaning of Operator
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder

Example:

```
let x = 4;
let y = 10;

console.log('Addition of x + y = ', x + y);
console.log('Subtraction of x - y = ', x - y);
console.log('Multiplication of x * y = ', x * y);
```

```
console.log('Division of x / y = ', x / y);
console.log('Remainder of x % y = ', x % y);
```

Output:

```
Addition of x + y =  14
Subtraction of x - y =  -6
Multiplication of x * y =  40
Division of x / y =  0.4
Remainder of x % y =  4
```

2. Comparison Operators

This operator checks the relationship between two operands. Relation true then return 1 and Relation false return 0.

Operator	Meaning of Operator
==	Is equal to
===	Strict equal
!=	Not equal to
!==	Not strict equal
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

Example:

```
console.log(2 == 2);
console.log(2 != 2);
console.log(2 === 2);
console.log(2 !== '2');
console.log(2 > '2');
console.log(2 < '2');
console.log(2 <= '2');
console.log(2 >= '2');
```

Output:

```
true
false
```

true
true
false
false
true
true

3. Assignment Operators

Assignment operators are used to assign values to variables.

Operator	Meaning of Operator
=	Assignment operator
+=	Addition assignment
-=	Subtraction Assignment
*=	Multiplication Assignment
/=	Division Assignment
%=	Remainder Assignment

Example:

```
var a=4;
console.log(a);

a+=10;
console.log(a);

a-=10;
console.log(a);

a*=10;
console.log(a);

a/=10;
console.log(a);

a%=10;
console.log(a);
```

Output:

4
14
4
40
4
4
1048576

4. Logical Operators

logical operator returns either 0 or 1 depending upon whether expression results true or false.

Operator	Meaning of Operator
&&	Logical AND
||	Logical OR
!	Logical NOT

Example:

```
console.log(true && false); //false
console.log(4 && 10);   //10

console.log(true || false); //true
console.log(4 || 10);   //4

console.log(!true); //false
console.log(!4);   //false
```

Output:

false
10
true
4
false
false

5. Bitwise Operators

Arithmetic operation are converted to bit-level operation.

Operator	Meaning of Operator
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise complement
<<	Shift left
>>	Shift right

Example:

```
var a=4, b=10;

res = a & b;
console.log(res);

res = a | b;
console.log(res);

res = a ^ b;
console.log(res);

res = a << b;
console.log(res);

res = a >> b;
console.log(res);

res = a >>> b;
console.log(res);
```

Output:

```
0
14
14
4096
0
```

0

6. Other Operators

Operator	Meaning of Operator
?:	Conditional Operator returns value based on the condition. It is like if-else.
,	Comma Operator allows multiple expressions to be evaluated as single statement.
delete	Delete Operator deletes a property from the object.
in	In Operator checks if object has the given property
instanceof	checks if the object is an instance of given type
new	creates an instance (object)
typeof	checks the type of object.
void	it discards the expression's return value.
yield	checks what is returned in a generator by the generator's iterator.

Comments

JavaScript comments are hints that programmers can add to make their code easier to read and understand. They are completely ignored by the JavaScript engine. It is used to add information, warnings and suggestion of code.

There are two types of comments:

1. Single line comment
2. Multi-line comment

1. **Single line comment:**
   It is start with double forward slashes [//]. It is used before and after statement. For example,

```
name = "Gyansabha";

// printing name on the console
console.log("Hello " +name);
```

2. **Multi-line comment:**
   Any text between /* and */ is a multi-line comment. It can be used single line or multi line comments. For example,

```
/* print gyansabha
using mulit line comment
*/
name = "Gyansabha";
// printing name on the console
console.log("Hello " +name);
```