JAVASCRIPT

Name of jS was livescript

What is JS?

* It is the lightweight, interpreted programming langauge
* Design for creating network centrick applications
* Complementory and integrated with java and HTML
* Open and cross platform

Limitation of the javascript

* Client side JS does not allow the reading and writing of files.
* JS is not available for networking applications
* JS does not have a multi threading or multiprocessor capability

Things need to know about JS

JS will ignore the spaces, tab and new lines

JS is case sensitive langauge

Comments can be written as // or /\* \*/

JS placement in HTML file

on head, on body, both and on a file

1. On head section

If you want script run on some event, such as when user click somewhere then place script in head section

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <script type = "text/javascript">

      function sayHello() {

        alert("Hello World")

      }

    </script>

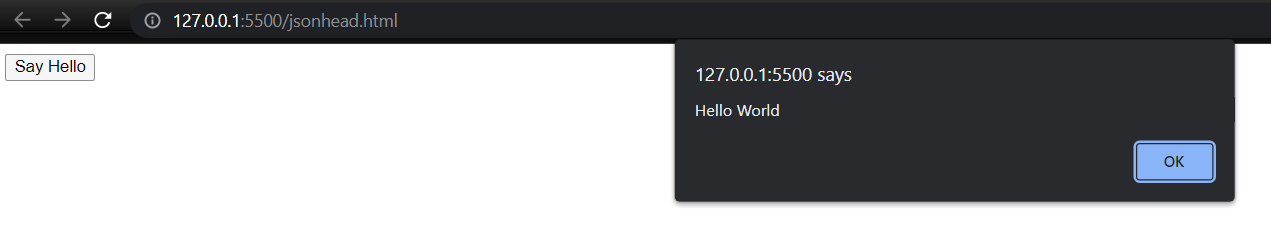
</head>

<body>

    <input type="button" onclick="sayHello()" value="Say Hello">

</body>

</html>



Hello world using a javascript

<html>

   <body>

      <script language = "javascript" type = "text/javascript">

        // <!--

            document.write("Hello World!")

         //-->

      </script>

   </body>

</html>

//Hello World!

//http://127.0.0.1:5500/helloworld.html

1. On body section

If script runs as page loads so that script generates content in the page.

1. JavaScript code needs to access the elements in the HTML document. By placing the code in the body section, it ensures that all elements in the document have been loaded and can be manipulated by the JavaScript code.
2. JavaScript can modify the content and layout of the HTML page. Placing the code in the body section makes it possible to execute the code after the content of the page has been loaded and displayed to the user.
3. Better performance: By placing JavaScript at the end of the body section, the browser can display the content of the page first, improving the loading performance of the page.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        document.write("Hello World")

    </script>

        <p> This is a body section </p>

</body>

</html>

/\*

jsonbody.html

Hello World

This is a body section

\*/

Applications of the JS

1. Client side validation
2. Manipulate HTML pages
3. User notifications
4. Backend data loading
5. Presentations
6. Server applications

JS datatypes

* 1. Numbers
  2. String
  3. Boolean

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        var money

        var name = "Yagnik"

        var value = true

        money = 10

        document.write(money, name, value)

    </script>

</body>

</html>

<!--

10Yagniktrue

-->

Scope of the variables in JS

Local variable :- Variable declared inside the function and is valid only inside function.

Globle variable :- Variable declared outside of the function.

Local variable inside the function has highest priority inside the function.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body onload="checkscope()">

    <script>

        var myvar = "globle"

function checkscope(){

var myvar = "local"

document.write(myvar)

}

    </script>

</body>

</html>

<!--

local

-->

Reserved keywords in JS

|  |  |  |  |
| --- | --- | --- | --- |
| abstract | else | instanceof | switch |
| boolean | enum | int | synchronized |
| break | export | interface | this |
| byte | extends | long | throw |
| case | false | native | throws |
| catch | final | new | transient |
| char | finally | null | true |
| class | float | package | try |
| const | for | private | typeof |
| continue | function | protected | var |
| debugger | goto | public | void |
| default | if | return | volatile |
| delete | implements | short | while |
| do | import | static | with |
| double | in | super |  |

Operators in JS

|  |  |
| --- | --- |
| **Sr.No.** | **Operator & Description** |
| 1 | **+ (Addition)**  Adds two operands  **Ex:** A + B will give 30 |
| 2 | **- (Subtraction)**  Subtracts the second operand from the first  **Ex:** A - B will give -10 |
| 3 | **\* (Multiplication)**  Multiply both operands  **Ex:** A \* B will give 200 |
| 4 | **/ (Division)**  Divide the numerator by the denominator  **Ex:** B / A will give 2 |
| 5 | **% (Modulus)**  Outputs the remainder of an integer division  **Ex:** B % A will give 0 |
| 6 | **++ (Increment)**  Increases an integer value by one  **Ex:** A++ will give 11 |
| 7 | **-- (Decrement)**  Decreases an integer value by one  **Ex:** A-- will give 9 |

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

            var myvar1 = 20;

            var myvar2 = 10;

            var myvar3 = "test";

            var linebreak = "<br />";

            document.write(myvar1+myvar2);

            document.write ("<br>");

            document.write(myvar1+myvar2+myvar3);

            document.write ("<br>");

            document.write(myvar1-myvar2);

            document.write ("<br>");

            document.write(myvar1/myvar2);

            document.write ("<br>");

            document.write(myvar1\*myvar2);

            document.write ("<br>");

            document.write(myvar1++);

            document.write(linebreak);

            document.write(myvar1--);

            document.write(linebreak);

            document.write(33 % 10);

    </script>

</body>

</html>

<!--

30

30test

10

2

200

20

21

3

-->

How to get into new line in JS?

             document.write(linebreak);

or

document.write ("<br>");

Comparison operators

|  |  |
| --- | --- |
| **Sr.No.** | **Operator & Description** |
| 1 | **= = (Equal)**  Checks if the value of two operands are equal or not, if yes, then the condition becomes true.  **Ex:** (A == B) is not true. |
| 2 | **!= (Not Equal)**  Checks if the value of two operands are equal or not, if the values are not equal, then the condition becomes true.  **Ex:** (A != B) is true. |
| 3 | **> (Greater than)**  Checks if the value of the left operand is greater than the value of the right operand, if yes, then the condition becomes true.  **Ex:** (A > B) is not true. |
| 4 | **< (Less than)**  Checks if the value of the left operand is less than the value of the right operand, if yes, then the condition becomes true.  **Ex:** (A < B) is true. |
| 5 | **>= (Greater than or Equal to)**  Checks if the value of the left operand is greater than or equal to the value of the right operand, if yes, then the condition becomes true.  **Ex:** (A >= B) is not true. |
| 6 | **<= (Less than or Equal to)**  Checks if the value of the left operand is less than or equal to the value of the right operand, if yes, then the condition becomes true.  **Ex:** (A <= B) is true. |

Logical Operators

|  |  |
| --- | --- |
| **Sr.No.** | **Operator & Description** |
| 1 | **&& (Logical AND)**  If both the operands are non-zero, then the condition becomes true.  **Ex:** (A && B) is true. |
| 2 | **|| (Logical OR)**  If any of the two operands are non-zero, then the condition becomes true.  **Ex:** (A || B) is true. |
| 3 | **! (Logical NOT)**  Reverses the logical state of its operand. If a condition is true, then the Logical NOT operator will make it false.  **Ex:** ! (A && B) is false. |

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        a = true

        b = false

        document.write(a&&b)

        document.write("<br>")

        document.write(a||b)

        document.write("<br>")

        document.write(!(a||b))

    </script>

</body>

</html>

<!--

false

true

false

-->

Bit wise operators

|  |  |
| --- | --- |
| **Sr.No.** | **Operator & Description** |
| 1 | **& (Bitwise AND)**  It performs a Boolean AND operation on each bit of its integer arguments.  **Ex:** (A & B) is 2. |
| 2 | **| (BitWise OR)**  It performs a Boolean OR operation on each bit of its integer arguments.  **Ex:** (A | B) is 3. |
| 3 | **^ (Bitwise XOR)**  It performs a Boolean exclusive OR operation on each bit of its integer arguments. Exclusive OR means that either operand one is true or operand two is true, but not both.  **Ex:** (A ^ B) is 1. |
| 4 | **~ (Bitwise Not)**  It is a unary operator and operates by reversing all the bits in the operand.  **Ex:** (~B) is -4. |
| 5 | **<< (Left Shift)**  It moves all the bits in its first operand to the left by the number of places specified in the second operand. New bits are filled with zeros. Shifting a value left by one position is equivalent to multiplying it by 2, shifting two positions is equivalent to multiplying by 4, and so on.  **Ex:** (A << 1) is 4. |
| 6 | **>> (Right Shift)**  Binary Right Shift Operator. The left operand’s value is moved right by the number of bits specified by the right operand.  **Ex:** (A >> 1) is 1. |
| 7 | **>>> (Right shift with Zero)**  This operator is just like the >> operator, except that the bits shifted in on the left are always zero.  **Ex:** (A >>> 1) is 1. |

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

            var a = 2; // Bit presentation 10

            var b = 3; // Bit presentation 11

            document.write(a|b)

            document.write("<br>")

            document.write(a&b)

            document.write("<br>")

            document.write(a^b)

            document.write("<br>")

            document.write(~b)

            document.write("<br>")

            document.write(a<<b)

            document.write("<br>")

            document.write(a>>b)

            document.write("<br>")

            document.write(a>>>b)

               </script>

</body>

</html>

<!--

3

2

1

-4

16

0

0

-->

Conditional operators

|  |  |
| --- | --- |
| **Sr.No.** | **Operator and Description** |
| 1 | **? : (Conditional )**  If Condition is true? Then value X : Otherwise value Y |

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        a = 10

        b = 20

        document.write((a>b)?100:200)

        document.write("<br>")

        document.write((a>b)?200:"100 is high")

    </script>

</body>

</html>

<!--

200

100 is high

-->

**Type of Operator**

The typeof operator is a unary operator that is placed before its single operand, which can be of any type. Its value is a string indicating the data type of the operand.

The typeof operator evaluates to "number", "string", or "boolean" if its operand is a number, string, or boolean value and returns true or false based on the evaluation.

Here is a list of the return values for the typeof Operator.

|  |  |
| --- | --- |
| **Type** | **String Returned by type of** |
| Number | "number" |
| String | "string" |
| Boolean | "boolean" |
| Object | "object" |
| Function | "function" |
| Undefined | "undefined" |
| Null | "object" |

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        a = 10

        b = "String"

document.write(typeof a)

document.write("<br>")

document.write(typeof a == "string" ? "a is string": "a is number")

document.write("<br>")

document.write(typeof b == "string" ? "a is string": "a is number")

    </script>

</body>

</html>

<!--

number

a is number

a is string

-->

Consitions in JS

If conditions

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        var age = 18

        if (age > 18) {

            document.write("Age is above 18")

        } else if (age == 18){

            document.write("Age is 18")

        } else {

            document.write("Age is above 18")

        }

    </script>

</body>

</html>

<!--

    Age is 18

-->

Switch conditions

Switch condition with break

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        var grade = "A"

        switch (grade){

            case "A" : document.write("Excellent")

            break

            case "B" : document.write("Very good")

            break

            case "C" : document.write("good")

            default : document.write("No grade")

        }

    </script>

</body>

</html>

Switch conditions without break

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        var grade = "A"

        switch (grade){

            case "A": document.write("Excellent<br>");

            case "B": document.write("Very good<br>");

            case "C": document.write("good<br>");

            case "D": document.write("Average<br>");

            default: document.write("No grade<br>")

        }

    </script>

</body>

</html>

<!--

Excellent

Very good

good

Average

No grade

Here in a switch statement break will plays a vital role. If not used then aswitch will check al the conditions.

-->

While loop in JS

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        var a = 0

        while(a<10){

            document.write(a)

            document.write("<br>")

            a++

        }

    </script>

</body>

</html>

<!--

0

1

2

3

4

5

6

7

8

9

-->

Do while loop in JS

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        a = 0

        do{

document.write(a+"<br>")

a++

        }while(a<5)

    </script>

</body>

</html>

<!--

0

1

2

3

4

-->

For loop in JS

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        var count

        for (count=0;count<=10;count++) {

            document.write(count+"<br>")

        }

    </script>

</body>

</html>

<!--

    0

1

2

3

4

5

6

7

8

9

10

-->

For in loop,

For in loop is similar to range in golang

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

                const person = {fname:"John", lname:"Doe", age:25};

                let text = "";

                for (let x in person) {

                text += person[x];

                }

                document.write(text)

    </script>

</body>

</html>

<!--

    JohnDoe25

-->

Loop control using break

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        var a = 0

        while (a<20){

                if (a==10){

                    break;

                }

                document.write(a+"<br>")

                a++

        }

    </script>

</body>

</html>

<!--

    0

1

2

3

4

5

6

7

8

9

-->

Continue statement loop control

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        var a = 1

        while(a<7){

            a++

            if (a==5){

                continue

            }

            document.write(a+"<br>")

        }

    </script>

</body>

</html>

<!--

    2

3

4

6

7

-->

Break label

or

go to statement with break

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

gotolableinner:

        for (let j=0;j<6;j++){

        gotolable:

        for (let i=0;i<10;i++){

            document.write(i+"<br>")

            if (i==2)  break gotolable;

            if (i==4) break gotolableinner;

        }

    }

        document.write("Exit from loop")

    </script>

</body>

</html>

<!--

0

1

2

0

1

2

0

1

2

0

1

2

0

1

2

0

1

2

Exit from loop

-->

Continue label

or

go to statement with continue

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

gotolableinner:

        for (let j=0;j<6;j++){

        gotolable:

        for (let i=0;i<10;i++){

            document.write(i+"<br>")

            if (i==2)  continue gotolable;

            if (i==4) continue gotolableinner;

        }

    }

        document.write("Exit from loop")

    </script>

</body>

</html>

<!--

0

1

2

3

4

0

1

2

3

4

0

1

2

3

4

0

1

2

3

4

0

1

2

3

4

0

1

2

3

4

Exit from loop

-->

Function that will show a button and when pressed then show some text.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <input type = "button" onclick="sayhello()" value="sayhello">

    <script>

function sayhello(){

document.write("Hello beautifull world")

}

    </script>

</body>

</html>

<!--

    buttonsayhello.html

    When button is click then button will gone and display

    Hello beautifull world

-->

Function which shows the age and height

or

function with when button pressed then shows the age and height

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <input type = "button" onclick="age(15,7)" value = "Say Hello">

    <script>

function age(age, height){

document.write("your age is "+age, " your height is " + height)

}

    </script>

</body>

</html>

<!--

    Function with parameter height and age

    Initially shows button when clicked then shows below text

    your age is 15 your height is 7

-->

Return a value from the function

or

calling the function

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <input type ="button" onclick="callfunction()" value="Calling the function">

    <script>

function callfunction() {

name = concatename("yagnik", "pokal")

document.write(name)

}

function concatename(fname, lname) {

    myname = fname +" "+ lname

    return myname

}

    </script>

</body>

</html>

<!--

    Return a value from the function or calling the function

When button is pressed then call another function

Another function gives the name and print on screen as below

yagnik pokal

-->

Remaining topics

* [JavaScript Nested Functions](https://www.tutorialspoint.com/javascript/javascript_nested_functions.htm" \o "JavaScript Nested Functions)
* [JavaScript Function( ) Constructor](https://www.tutorialspoint.com/javascript/javascript_function_constructors.htm" \o "JavaScript Function Constructors)
* [JavaScript Function Literals](https://www.tutorialspoint.com/javascript/javascript_function_literals.htm" \o "JavaScript Function Literals)

Event in JS

or

alert in js

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <input type="button" onclick="showalert()" value="showalert">

    <script>

function showalert(){

    alert("Heyy how are you")

}

    </script>

</body>

</html>

