Introduction To Javascript

JavaScript is a very powerful client-side scripting language. It is used mainly for enhancing the webpage by making it more lively and interactive. It is one of the most popular programming languages now a days and used to create various mobile apps, desktop apps, games and many more…

Client-side Programming : It is the program that runs on the client machine (browser) and deals with the user interface/display and any other processing that can happen on client machine

Write First JavaScript Code

There are so many ways to write a JavaScript code. The first is writing with in an HTML document using SCRIPT tag. The SCRIPT tag can be placed with in HEAD tag or BODY tag.

Remember that, If we are placing the script in the head section, Then it will be executed before the <body> is rendered. So the page load time will be more. If we want to speed up the page load time then we should place our script at the end of the body tag.

Refer : Example1.html and Example2.html

document.write

alert()

// (Single Line Comment) : Ignores rest of the code on same line

/*…*/ (Multi Line Comment) : Ignores all the text with in the block

What is ECMAScript or ES ?

ECMAScript (or ES) is a general-purpose programming language, standardized by ECMA International according to the document ECMA-262. It is a JavaScript standard meant to ensure the interoperability of Web pages across different Web browsers.

Variables in JavaScript

Variable is a named memory location to hold value. The value can be changed at any time. var keyword can be used to declare a variable name and = (assignment operator) can be used to assign value to a variable. Declaration and assignment can be done at same time. Remember, a variable declared with out any value, have the value *undefined*. If we redeclare a variable, it will not loose its value.

Refer : Example5.html

Variable Naming Rules in JavaScript

- JavaScript variable names are case sensitive so x and X are two different names.
- JS Variable names can contain letters, digits, underscores, or dollar signs but first character can not be a digit.
- JS Variable names can not contain reserved keywords. Remember, Reserved words are in lower case so to avoid mistake start the first letter of the variable name with upper case letter.

we should use var Var

Data Types in JavaScript

To properly operate data we must know it's data type. JavaScript provides eight different data types which are undefined, null, Boolean, string, symbol, bigint, number, and object.

Remember :

- Same variable can hold different data type if assigned
- typeof operator can be used to get the data type of variable
- JS Numbers are stored as double precision floating point numbers
- Strings are mentioned with in single quotes or double quotes. Escape characters (\") can be used to put double quotes inside double quotes or single quotes inside single quotes.
- Boolean data type has two values only true or false. These are very helpful when we have to store only two values like yes or no.

Refer : Example7.html

More About Double Precision Floating Point Number

It's called a floating point number because the decimal point can "float" around (move left and right).

The "precision" of a number refers to how many digits it can have.

A “Single precision floating point number” can store only 32 bits!

A "double precision floating point number" is a decimal-type number that can store 64 bits (binary digits) of information.

Arithmetic Operators

+ (addition) : Adds both the operands

- (Subtraction) : Subtracts the right operand from left

* (Multiplication) : Multiplies both the operands

/ (Division) : Divides left operand with right

% (Modulus) : Performs integer division and gives the reminder

Refer : Example8.html

Increment & Decrement Operators in JavaScript

var++ (Post Increment) : Uses the original value or var in expression then increases the value by 1

++var (Pre Increment) : Increases the value by 1 Then uses the modified new value in the expression.

Var-- (Post Decrement) : Uses the original value or var in expression then decreases the value by 1

--var (Pre Decrement) : Decreases the value by 1 Then uses the modified new value in the expression.

Refer : Example9.html

```
<title>Increment & Decrement Operators in JavaScript</
<script type="text/javascript">
    a = 5;b = 6
    document.write(a++ + b+"<br/>");
    a = 5;b = 6
    document.write(++a + b+"<br/>");
    a = 5;b = 6
    document.write(a-- + b+"<br/>");
    a = 5;b = 6
    document.write(--a + b+"<br/>");
</script>
```

Assignment Operators

= : Assigns the value of right operand to left

+= : adds both the operands and assigns value to left operand

-= : subtracts right operand from left and assigns value to left operand

*= : product of both the operands assigns to left operand

/= : Divides left operand with right and assigns result to left operand

%= : Finds the modulus and assigns to left operand

Refer : Example10.html

```
//a*=b // a = a * b
//document.write(" ",a)
//a /=b // a = a / b
//document.write(" ",a)
a%=b // a = a % b
document.write(" ",a)
```

Comparison Operators in JavaScript

== : (Equal to) Returns true if the value of both the operands are same

=== : (Identical to) Returns true if the value and data type of both the operands are same

!= : (Not Equal to) Returns true if both are not equal

!== : (Not Identical) Returns true if both are not identical

> : (Greater Than) Returns true if left operand is greater then right

>= : (Greater Than or Equal To) Returns true if left operand is greater than or equal to right

< : (Less Than) Returns true if left operand is less than right

<= : (Less Than or Equal To) Returns true if left operand is less than or equal to right

Refer : Example11.html

```
//  document.write(5==3)
//  document.write(5==5)
//  document.write(5=='5')
//  document.write(5==='5')
    document.write(5===5)
//  document.write(5!='5')
//  document.write(5!=3)
//  document.write(5!=='5')
//  document.write(5 > 3)
```

Logical Operators in JavaScript

&& (Logical And) : Returns true if both the operands are true

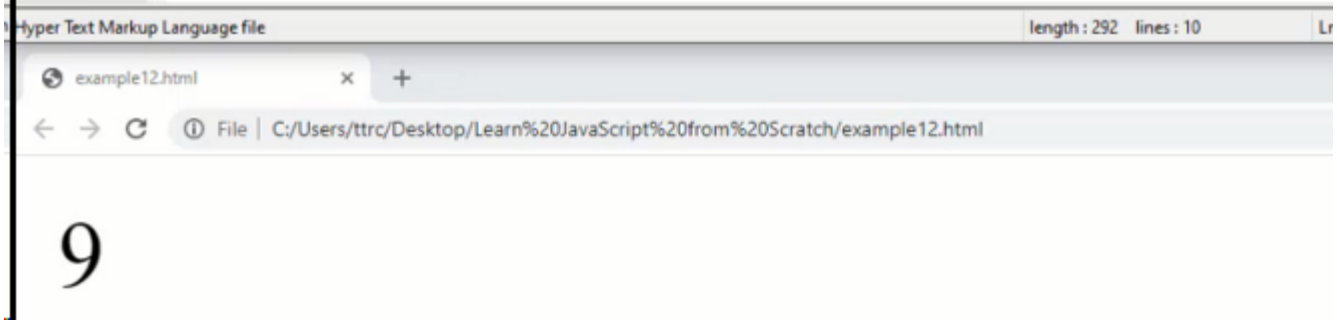
|| (Logical Or) : Returns true if one of the operands is true

! (Not Operator) : Returns true if the operand is false and returns false if the operand is true.

Refer : Example12.html

```
//document.write((5>3)&&(5>4))
//document.write((5>3)&&(5>6))
document.write((5>3)||(5>6))
//document.write(!(5>6))
```

```
6     document.write((5>3)&&(5+4))
7     //document.write((5>3)&&(5>6))
8     //document.write((5>3)||(5>6))
9     //document.write(!(5>6))
10 </script>
```




```
//document.write((5>3)||(5>6))
//document.write(!(5>6))
//document.write((5>3)&&(5+4))
//document.write((5<3)&&(5+4))
//document.write((5>3)||(5+6))
document.write((5<3)||(5+6))
```

Ternary Operators in JavaScript

Syntax :

Variable = (conditional expression) ? Value if true : value if false

If the conditional expression is evaluated true, value if true will be assigned to variable other wise value if false will be assigned to the variable.

Remember, Nested ternary operators can also be used if necessary.

Refer : Example13.html

```
<script type="text/javascript">
    age=17
    document.write(age>=18?"Major":"Minor")
</script>
```