

JAVASCRIPT



What is JavaScript

- ✓ **JavaScript** was designed to add interactivity to **HTML** pages
- ✓ **JavaScript** is a scripting language
- ✓ A scripting language is a lightweight programming language
- ✓ **JavaScript** is usually embedded directly into **HTML** pages
- ✓ **JavaScript** is an interpreted language (means that scripts execute without preliminary compilation)
- ✓ Everyone can use **JavaScript** without purchasing a license
- ✓ The Real Name of **JavaScript** is **ECMAScript**

Are Java and JavaScript the same

- ✓ NO!
- ✓ **Java** and **JavaScript** are two completely different languages in both concept and design!
- ✓ **Java** (*Developed by Sun Microsystems*) is a powerful and much more complex programming language - in the same category as **C** and **C++**.

Put a JavaScript into an HTML page

- ✓ You have to always write the JavaScript code under the **<script> ... </script>** tag
- ✓ The example below shows how to use JavaScript to write text on a web page:

```
<html>
<body>
<script type="text/javascript">
document.write("Hello World!");
</script>
</body>
</html>
```

- ✓ The example below shows how to add **HTML** tags to the JavaScript:

```
<html>
<body>
<script type="text/javascript">
document.write("<h1>Hello World!</h1>");
</script>
</body>
</html>
```

- ✓ The **Document.Write** command is a standard **JavaScript** command for writing output to a page.

Where to Put the JavaScript

- ✓ You can place an unlimited number of scripts in your document, so you can have scripts in both the body and the head section.

Using an External JavaScript

- ✓ The external script cannot contain the **<script></script>** tags!
- ✓ To use the external script, point to the **.js** file in the **"src"** attribute of the **<script>** tag:

Example

```
<html>
<head>
```



```
<script type="text/javascript" src="ycsd.js"></script>
</head>
<body>
</body>
</html>
```

JavaScript Statements

JavaScript is Case Sensitive

- ✓ Unlike **HTML**, **JavaScript** is case sensitive - therefore watch your capitalization closely when you write **JavaScript** statements, create or call variables, objects and functions.

JavaScript Statements

- ✓ A **JavaScript** statement is a command to a browser. The purpose of the command is to tell the browser what to do.
- ✓ This **JavaScript** statement tells the browser to write "Hello YCS D" to the web page:


```
document.write("Hello YCS D");
```
- ✓ It is normal to add a semicolon at the end of each executable statement. Most people think this is a good programming practice, and most often you will see this in JavaScript examples on the web.
- ✓ **The Semicolon Is Optional** (according to the **JavaScript** standard), and the browser is supposed to interpret the end of the line as the end of the statement. Because of this you will often see examples without the semicolon at the end.
- ✓ **Using Semicolons** makes it possible to write multiple statements on one line.

JavaScript Code

- ✓ **JavaScript** code (or just **JavaScript**) is a sequence of **JavaScript** statements.
- ✓ Each statement is executed by the browser in the sequence they are written.
- ✓ This example will write a heading and two paragraphs to a web page:

```
<script type="text/javascript" >
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
```

JavaScript Blocks

- ✓ **JavaScript** statements can be grouped together in blocks.
- ✓ Blocks start with a left curly bracket {, and ends with a right curly bracket }.
- ✓ The purpose of a block is to make the sequence of statements execute together.
- ✓ This example will write a heading and two paragraphs to a web page:

```
<script type="text/javascript" >
{
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
}
</script>
```

- ✓ The example above is not very useful. It just demonstrates the use of a block. Normally a block is used to group statements together in a function or in a condition (where a group of statements should be executed if a condition is met).

JavaScript Comments

- ✓ **Comments** can be added to explain the JavaScript, or to make the code more readable.
- ✓ **Single Line** comments start with `//`.
- ✓ **Multi Line** comments start with `/*` and end with `*/`.

Example

```
<script>
// The code below will write Hello YCSD

/*
The code below will
write Hello YCSD
*/
document.write("<h1>Hello YCSD</h1>");
</script>
```

- ✓ The `type = "text/javascript"` is optional.

JavaScript Variables

- ✓ Do you remember algebra from school? $x=5$, $y=6$, $z=x+y$
- ✓ Do you remember that a letter (like x) could be used to hold a value (like 5), and that you could use the information above to calculate the value of z to be 11?
- ✓ These letters are called variables, and variables can be used to hold values ($x=5$) or expressions ($z=x+y$).

JavaScript Variables

- ✓ As with algebra, **JavaScript** variables are used to hold values or expressions.
- ✓ A variable can have a short name, like x , or a more descriptive name, like **carname**.
- ✓ Rules for **JavaScript** variable names:
 - Variable names are case sensitive (y and Y are two different variables)
 - Variable names must begin with a letter or the underscore character

Declaring (Creating) JavaScript Variables

- ✓ You can declare **JavaScript** variables with the **var Statement**:

```
var x=5;
var carname="Volvo";
```

- ✓ When you assign a text value to a variable, use **quotes** around the value.
- ✓ No need to declare data type `int` or `char` or `string` like `c / c++`, **JavaScript** detects it automatically.

JavaScript Operators

JavaScript Arithmetic Operators

- ✓ Arithmetic operators are used to perform arithmetic between variables and/or values.
- ✓ Given that $y=5$, the table below explains the arithmetic operators:

Operator	Description	Example	Result
+	Addition	$x=y+2$	$x=7$
-	Subtraction	$x=y-2$	$x=3$
*	Multiplication	$x=y*2$	$x=10$
/	Division	$x=y/2$	$x=2.5$
%	Modulus (division remainder)	$x=y\%2$	$x=1$
++	Increment	$x=++y$	$x=6$
--	Decrement	$x=--y$	$x=4$

JavaScript Assignment Operators

- ✓ Assignment operators are used to assign values to JavaScript variables.
- ✓ Given that $x=10$ and $y=5$, the table below explains the assignment operators:

Operator	Example	Same As	Result
=	$x=y$		$x=5$
+=	$x+=y$	$x=x+y$	$x=15$
-=	$x-=y$	$x=x-y$	$x=5$
=	$x=y$	$x=x*y$	$x=50$
/=	$x/=y$	$x=x/y$	$x=2$
%=	$x%=y$	$x=x\%y$	$x=0$

The + Operator Used on Strings

- ✓ The **+** operator can also be used to add string variables or text values together.
- ✓ To add two or more string variables together, use the **+** operator.

```
txt1="Hello";
txt2="YCSD";
txt3=txt1+txt2;
```

- ✓ After the execution of the statements above, the variable txt3 contains **"HelloYCSD"**.
- ✓ To add a space between the two strings, insert a space into one of the strings:

```
txt1="Hello ";
txt2="YCSD";
txt3=txt1+txt2;
```

- ✓ or insert a space into the expression:

```
txt1="Hello";
txt2="YCSD";
txt3=txt1+" "+txt2;
```

- ✓ Now after the execution of the statements above will output **"Hello YCSD"**

Adding Strings and Numbers

- ✓ **The Rule Is:** If you add a number and a string, the result will be a string!

```
x=5+5;
document.write(x);
x="5"+"5";
document.write(x);
x=5+"5";
document.write(x);
x="5"+5;
document.write(x);
```

JavaScript Comparison and Logical Operators

Comparison Operators

- ✓ Comparison operators are used in logical statements to determine equality or difference between variables or values.
- ✓ Given that $x=5$, the table below explains the comparison operators:

Operator	Description	Example
==	is equal to	x==8 is false
===	is exactly equal to (value and type)	x===5 is true x===5 is false
!=	is not equal	x!=8 is true
>	is greater than	x>8 is false
<	is less than	x<8 is true
>=	is greater than or equal to	x>=8 is false
<=	is less than or equal to	x<=8 is true

How Can it be Used

- ✓ Comparison operators can be used in conditional statements to compare values and take action depending on the result:

```
if (age<18) document.write("Too young");
```

Logical Operators

- ✓ Logical operators are used to determine the logic between variables or values.
- ✓ Given that x=6 and y=3, the table below explains the logical operators:

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
	or	(x==5 y==5) is false
!	not	!(x==y) is true

Conditional Operator

- ✓ JavaScript also contains a conditional operator that assigns a value to a variable based on some condition.

Syntax

```
variablename=(condition)?value1:value2
```

Example

```
age = prompt();
party = (age<18)?"You are Not allow":"You are Welcome";
document.write(party);
```

JavaScript If...Else and Switch Statements

Conditional Statements

- ✓ Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.
- ✓ In JavaScript we have the following conditional statements:
 - **if statement** - use this statement to execute some code only if a specified condition is true
 - **if...else statement** - use this statement to execute some code if the condition is true and another code if the condition is false
 - **if...else if...else statement** - use this statement to select one of many blocks of code to be executed
 - **switch statement** - use this statement to select one of many blocks of code to be executed

Welcome In YCSD

For Access Entire Study Materials

For Student

Please Ensure Your Enrollment Verification

ENROLLMENT VERIFICATION

Student Registration No.

Date Of Birth

Check Now

For Franchise

Please Login Via Franchise Id & Password

CENTER LOGIN

User Name

Password

Login

Thank You for Visiting Us

YCSD

 Youth Computer Skill Development

 YCSD

 www.ycsd.in

 info@ycsd.in



www.ycsd.in