



Point to an external javascript file:

```
<script type="text/javascript" src="xxx.js"></script>
```

Comment:

```
/*Some  
Comment*/
```

or

```
Javascript Content Here//Some Comment
```

Basic Javascript Tag:

```
<script type="text/javascript">  
  <!--  
    Javascript Content Here;  
  //-->  
</script>
```

Alert Box:

```
alert("Muffins!");
```

Confirm Box:

```
function askForMuffins(){  
  var reply=confirm("Can I have a Muffin?");  
  if (reply==true){  
    alert("Thanks for the Muffin!");  
  }  
  else{  
    alert("What? No Muffins?");  
  }  
}
```

Prompt Box:

```
function nameMuffins(){  
  var name=prompt("Please enter the Muffins name","Mr.Muffin");  
  if (name!=null && name!=""){  
    document.write("Hello " + name + "!");  
  }  
}
```



Special Characters:

Code	Purpose
\'	Single Quote
\"	Double Quote
\\	Backslash
\n	New Line
\r	Carriage Return
\t	Tab
\b	Backspace
\f	Form Feed

Replace text in element:

```
<html>
  <body>
    <h1>My First Web Page</h1>
    <p id="demo"></p>
    <script type="text/javascript">
      <!--
        document.getElementById("demo").innerHTML="Muffins?";
      //-->
    </script>
  </body>
</html>
```

Defining a Function:

```
function functionname(var1,var2,...,varX){
  Javascript Content Here;
}
```

Calling a Function:

```
<html>
  <head>
    <script type="text/javascript">
      function someMuffins(){
        Javascript Content Here;
      }
    </script>
  </head>
  <body>
    <button type="button" onclick="someMuffins()">Some Muffins Button</button>
```



```
</body>  
</html>
```

Returning a value from a function:

```
function product(a,b){  
    return a*b;  
}
```

Declaring Local Variables:

Any declared with var in a function

```
var x;  
var x="Muffin";
```

Declaring Global Variables:

Any declared without var

```
x="Muffin";
```

Setting Variables:

```
x=5;  
x="Muffin";  
x=y+5;
```

Set Arrays:

```
var muffin=new Array();  
muffin[0]="Banana";  
muffin[1]="All Bran";  
muffin[2]="Plain";
```

or

```
var muffin=new Array("Banana","All Bran","Plain");
```

or

```
var muffin=["Banana","All Bran","Plain"];
```

Access Arrays:

```
document.write(muffin[0]);
```

Modifying Arrays:



```
muffin[0]="Chocolate Chip";
```

Adding to Arrays:

```
muffin.push("Lemon","Fruit");
```

Create Boolean Object:

```
var myBoolean=new Boolean();
```

Operators:

Code	Purpose
=	Assigns Values
+	Add Values
-	Subtract Values
*	Multiply Values
/	Divide Values
%	Division Remainder
++	Increment By 1
--	Decrement By 1

Comparison Operators:

Code	Purpose
==	Equal To
===	Exactly Equal To (Value And Type)
!=	Not Equal
>	Greater Than
<	Less Than
>=	Greater Than Or Equal To
<=	Less Than Equal To

Logical Operators:

Code	Purpose
&&	And
	Or
!	Not

Conditional Operator:

```
message=(muffins=="YES")?"Yay muffins!":"No Muffins?";
```



Adding String Values:

String values concatenate

```
txt1="I like";  
txt2="Muffins!";  
txt3=txt1+" "+txt2;  
txt3="I like Muffins!";
```

If Statement:

```
if (condition){  
    Javascript Content Here;  
}
```

Else Statement:

```
if (condition){  
    Javascript Content Here;  
}  
else{  
    Other Javascript Content Here;  
}
```

Else If Statement:

```
if (condition1){  
    Javascript Content Here;  
}  
else if (condition2){  
    Javascript Content Here;  
}  
else{  
    Javascript Content Here;  
}
```

Switch Statement:

```
switch (numberOfMuffins){  
    case 1:  
        document.write("You have 1 Muffins!");  
        break;  
    case 2:  
        document.write("You have 2 Muffins!");  
        break;  
    case 3:  
        document.write("You have 3 Muffins!");  
}
```



```
        break;
    default:
        document.write("Yum, Muffins!");
}
```

For Loop:

```
for (variable=startvalue;variable<=endvalue;variable=variable+increment){
    code to be executed
}
```

While Loop:

```
while (variable<=endvalue){
    Javascript Content Here;
}
```

Do While Loop:

```
do{
    Javascript Content Here;
}while (variable<=endvalue);
```

Infinite Loop:

```
var t;
function foreverLoop(){
    t=setTimeout("foreverLoop()",1000);
}
foreverLoop();
```

To Stop the Infinite Loop

```
clearTimeout(t);
```

ForIn Statement:

Loops through the properties of an object

```
for (variable in object){
    Javascript Content Here;
}
```



Break Statement:

Exits loop at that point

```
break;
```

Continue Statement:

break the current loop and continue with the next value

```
continue;
```

Acting on Event:

```
<button type="button" onclick="displayDate()">Display Date</button>
```

onSubmit:

```
<form method="post" action="xxx.htm" onsubmit="return checkForm()">
```

Navigator Object can provide information on the browser.

Set Cookies:

```
c_name = cookie name  
value = varuble to be stored  
exdays = expiry of cookie  
  
function setCookie(c_name,value,exdays){  
    var exdate=new Date();  
    exdate.setDate(exdate.getDate() + exdays);  
    var c_value=escape(value) + ((exdays==null) ? "" : "; expires="+exdate.toUTCString());  
    document.cookie=c_name + "=" + c_value;  
}
```

Returns Cookie:

```
function getCookie(c_name){  
    var i,x,y,ARRcookies=document.cookie.split(";");  
    for (i=0;i<ARRcookies.length;i++){  
        x=ARRcookies[i].substr(0,ARRcookies[i].indexOf("="));  
        y=ARRcookies[i].substr(ARRcookies[i].indexOf("=")+1);  
        x=x.replace(/^\\s+|\\s+$/g,"");  
        if (x==c_name){  
            return unescape(y);  
        }  
    }  
}
```



Checking for the Cookie:

```
function checkCookie(){
    var username=getCookie("username");
    if (username!=null && username!=""){
        alert("Welcome again " + username);
    }
    else{
        username=prompt("Please enter your name:","");
        if (username!=null && username!=""){
            setCookie("username",username,365);
        }
    }
}
```

TimeOut:

Executes code sometime in the future

```
var t=setTimeout("Javascript Content Here;",milliseconds);
```

Creating Objects:

```
muffinObj=new Object();
muffinObj.name="Mr.Muffin";
muffinObj.age=1;
muffinObj.kind="Banana";
```

or

```
muffinObj={name:"Mr.Muffin",age:1,kind:"Banana"};
```

Adding Methods:

```
muffinObj.eat=eat;
```

Print Command:

```
window.print();
```