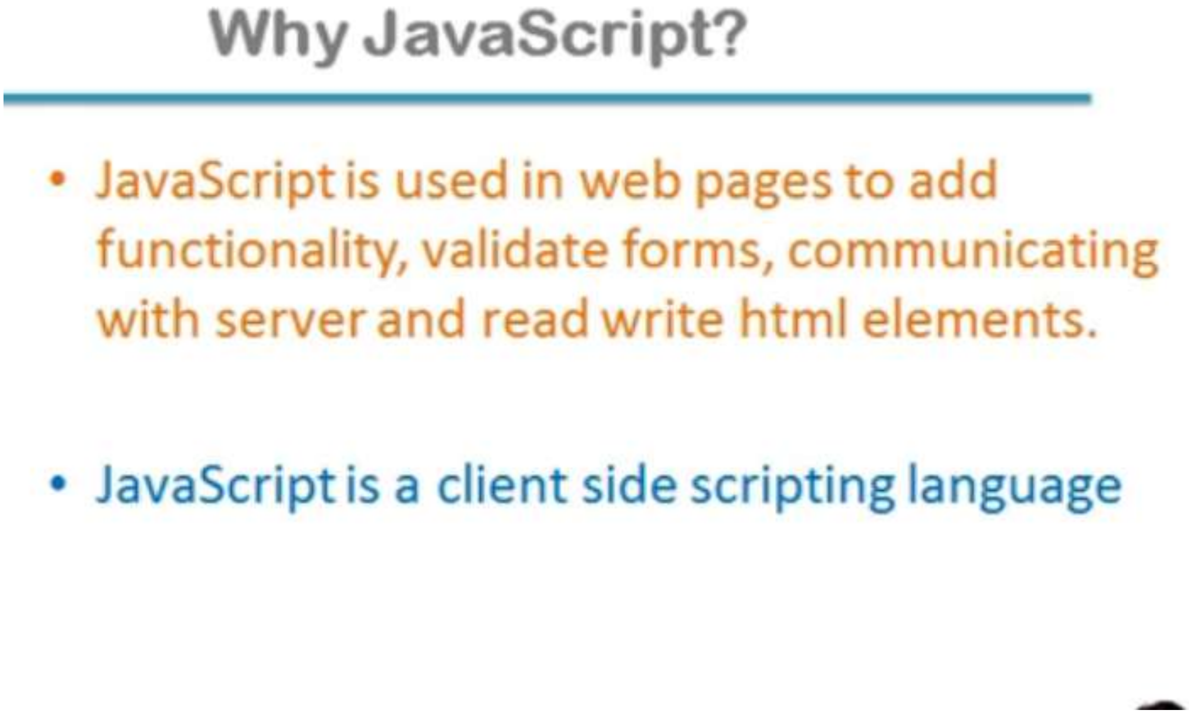
- JavaScript is a high level, un-typed and interpreted programming language

- Despite some naming, syntactic, and standard library similarities, JavaScript and Java are otherwise unrelated

- JavaScript is a high level, un-typed and interpreted programming language

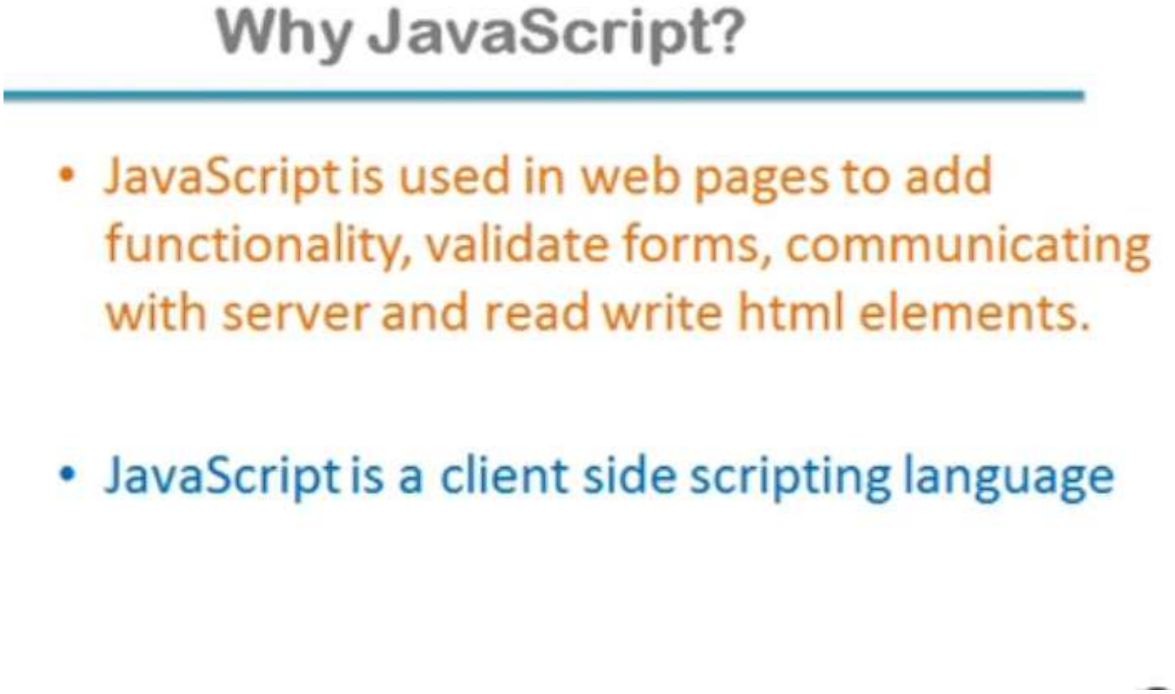
- Despite some naming, syntactic, and standard library similarities, JavaScript and Java are otherwise unrelated



Why JavaScript?

- JavaScript is used in web pages to add functionality, validate forms, communicating with server and read write html elements.
- JavaScript is a client side scripting language

Css and hmtl both are not programming language. Programming language is one which take decision by its self



Why JavaScript?

- JavaScript is used in web pages to add functionality, validate forms, communicating with server and read write html elements.
- JavaScript is a client side scripting language

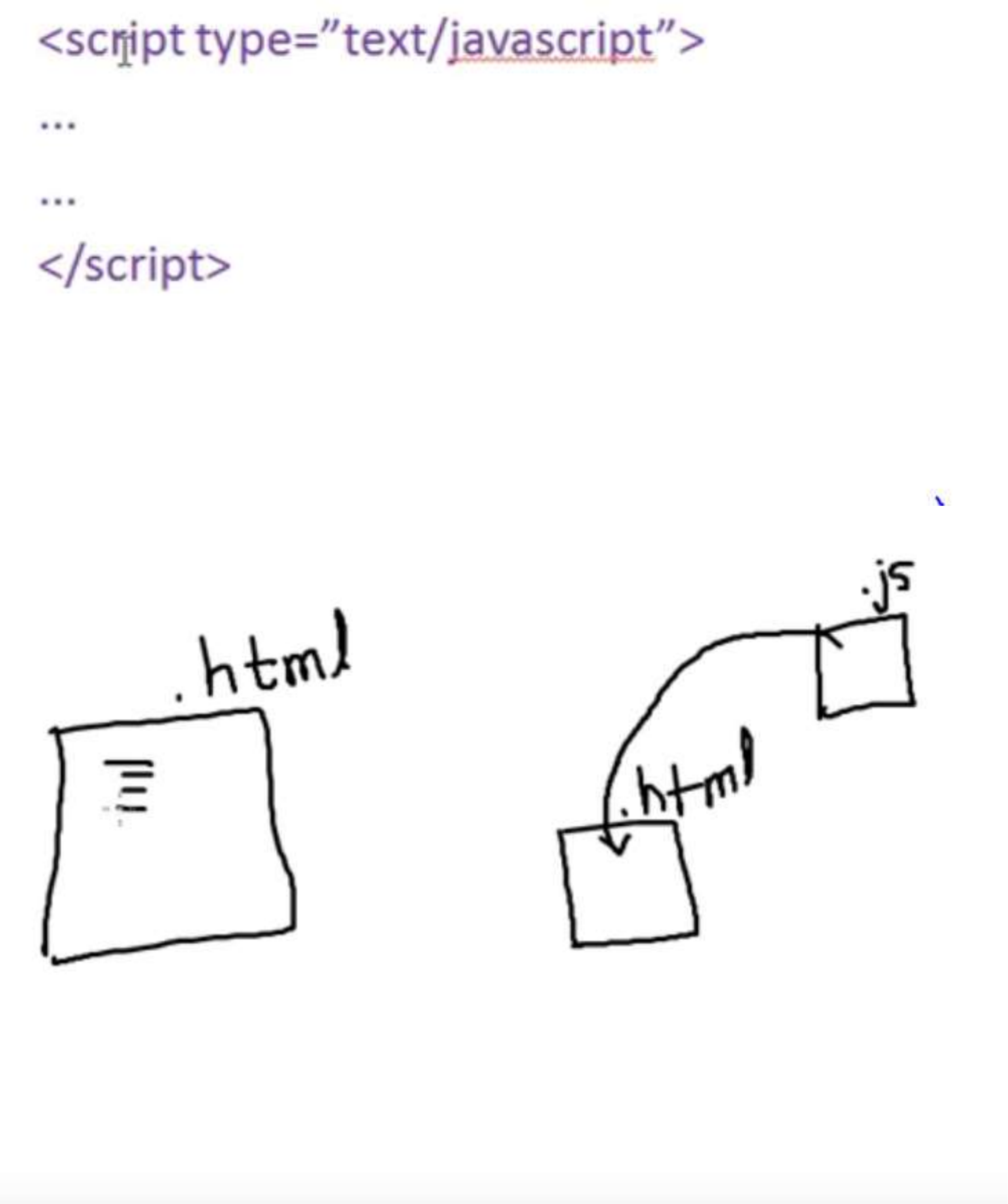
JavaScript Interpreter

- As the browser goes through the JavaScript, it passes it to a special program called interpreter, which converts the JavaScript to the machine code your computer understands

- Browsers come equipped with JavaScript interpreters.

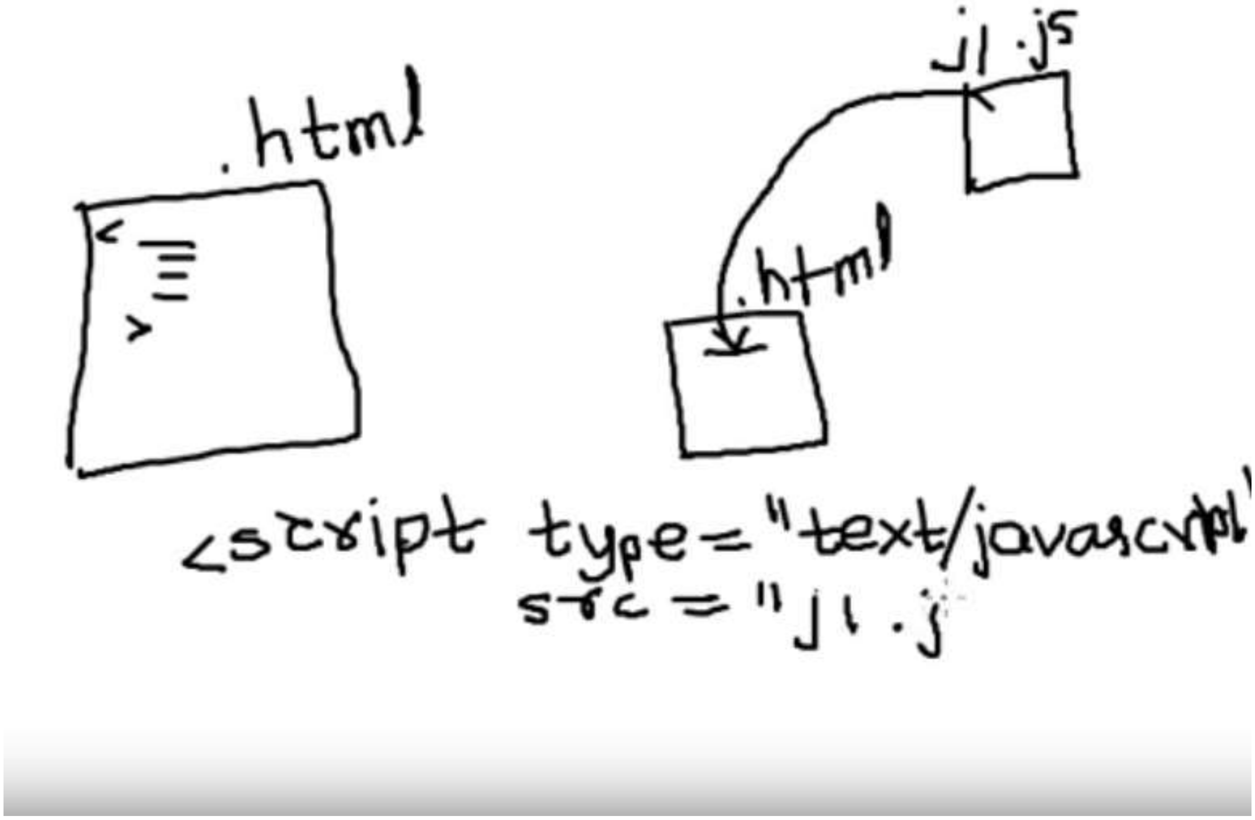
Where to write code?

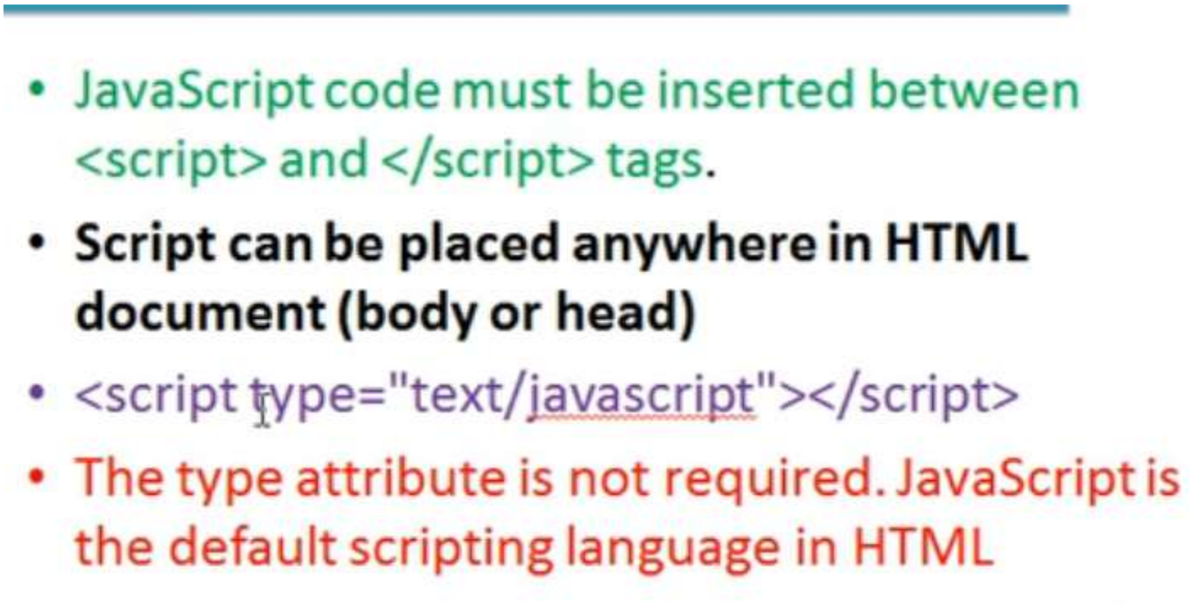
- In the same HTML file
- In a separate file, which can be embedded into HTML file



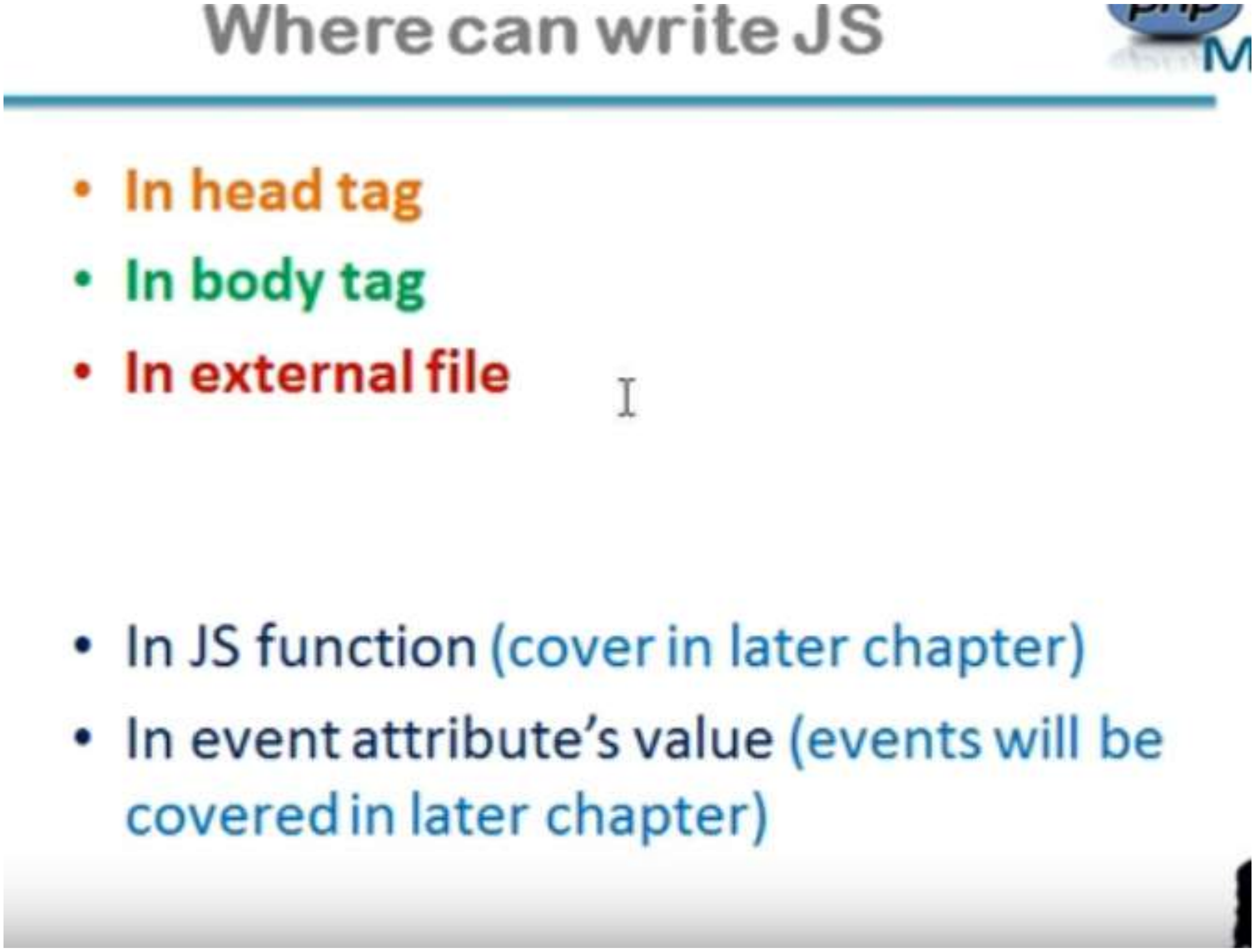
```
<script type="text/javascript">
...
...
</script>
```

.html
.js
.html





- JavaScript code must be inserted between <script> and </script> tags.
- **Script can be placed anywhere in HTML document (body or head)**
- <script type="text/javascript"></script>
- The type attribute is not required. JavaScript is the default scripting language in HTML



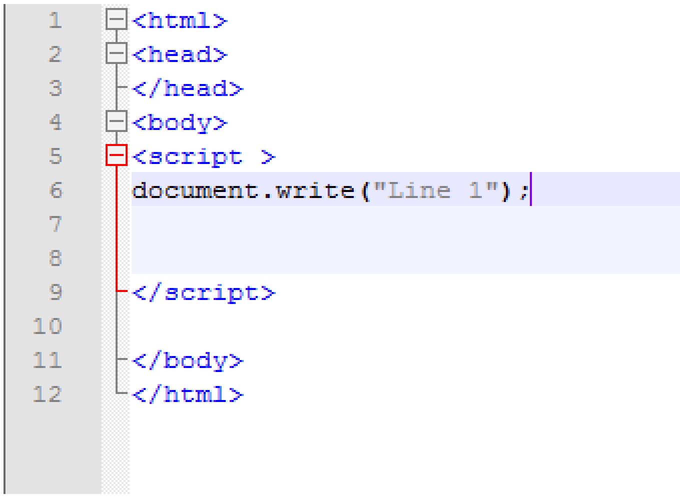
Where can write JS

- In head tag
- In body tag
- In external file

- In JS function (cover in later chapter)
- In event attribute's value (events will be covered in later chapter)

Document.write("");

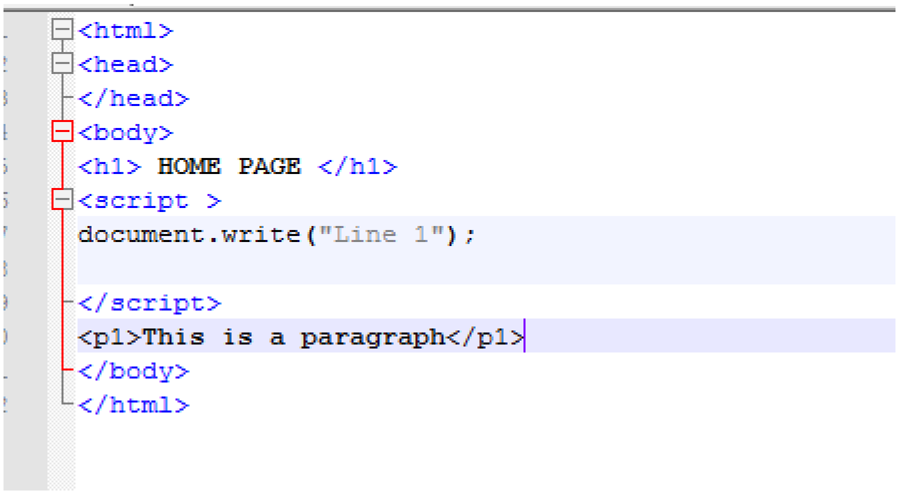
This function works same as we do in c++ as cout



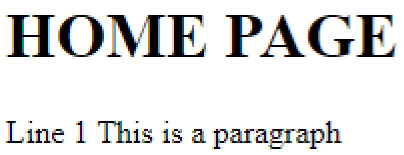
```
<html>
<head>
</head>
<body>
<script >
document.write("Line 1");


</script>

</body>
</html>
```

Output:

Line 1

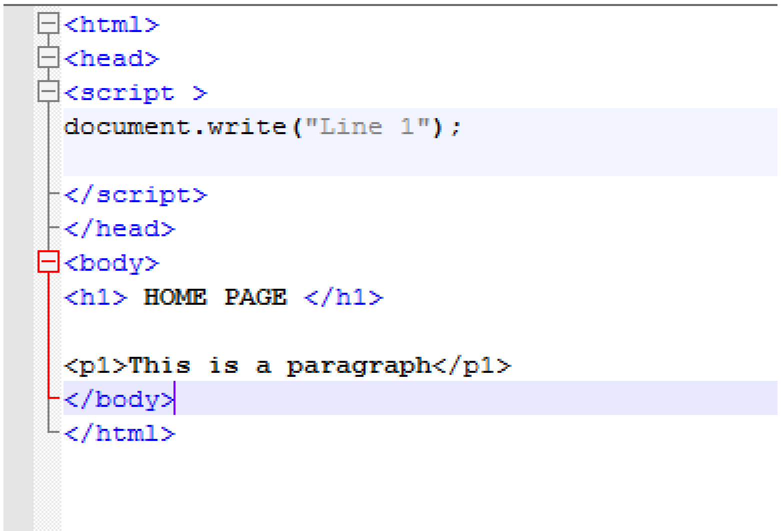


```
<html>
<head>
</head>
<body>
<h1> HOME PAGE </h1>
<script >
document.write("Line 1");

</script>
<p1>This is a paragraph</p1>
</body>
</html>
```

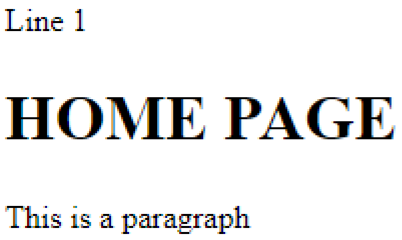


```
HOME PAGE

Line 1 This is a paragraph
```

In body this code run sequentically but if a palcew javascript in head so jaavascriptcode run first like

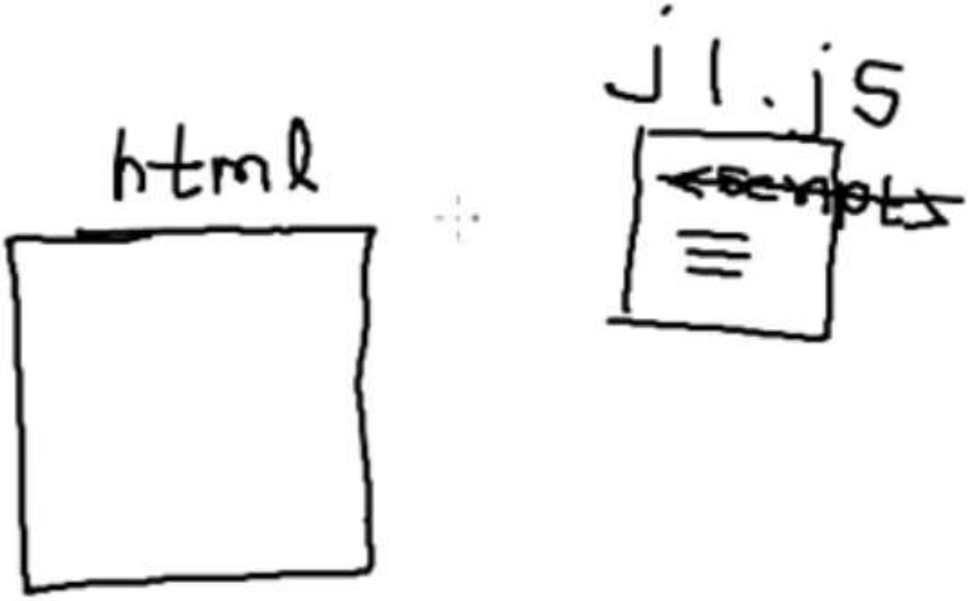


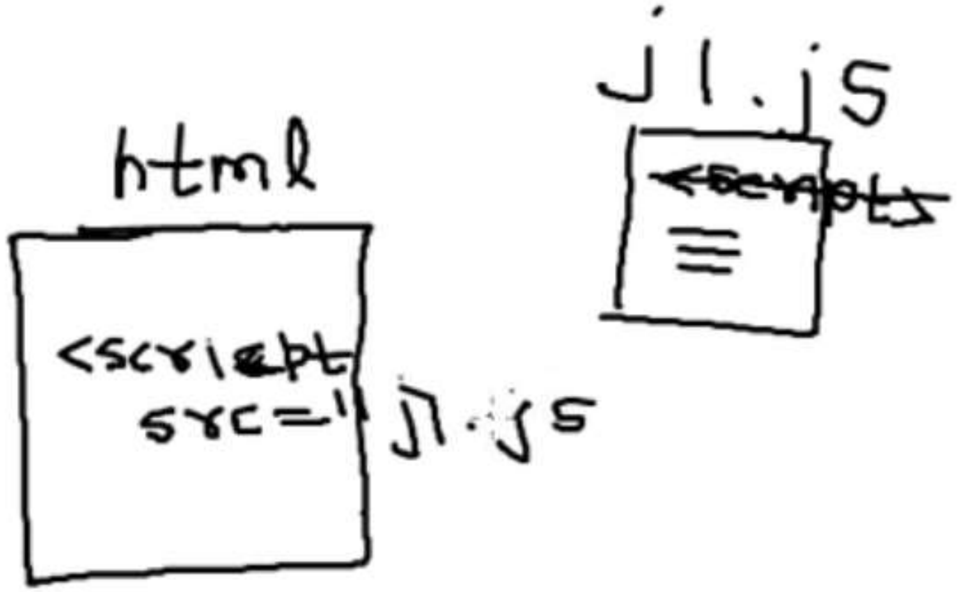
```
<html>
<head>
<script >
document.write("Line 1");

</script>
</head>
<body>
<h1> HOME PAGE </h1>

<p1>This is a paragraph</p1>
</body>
</html>
```

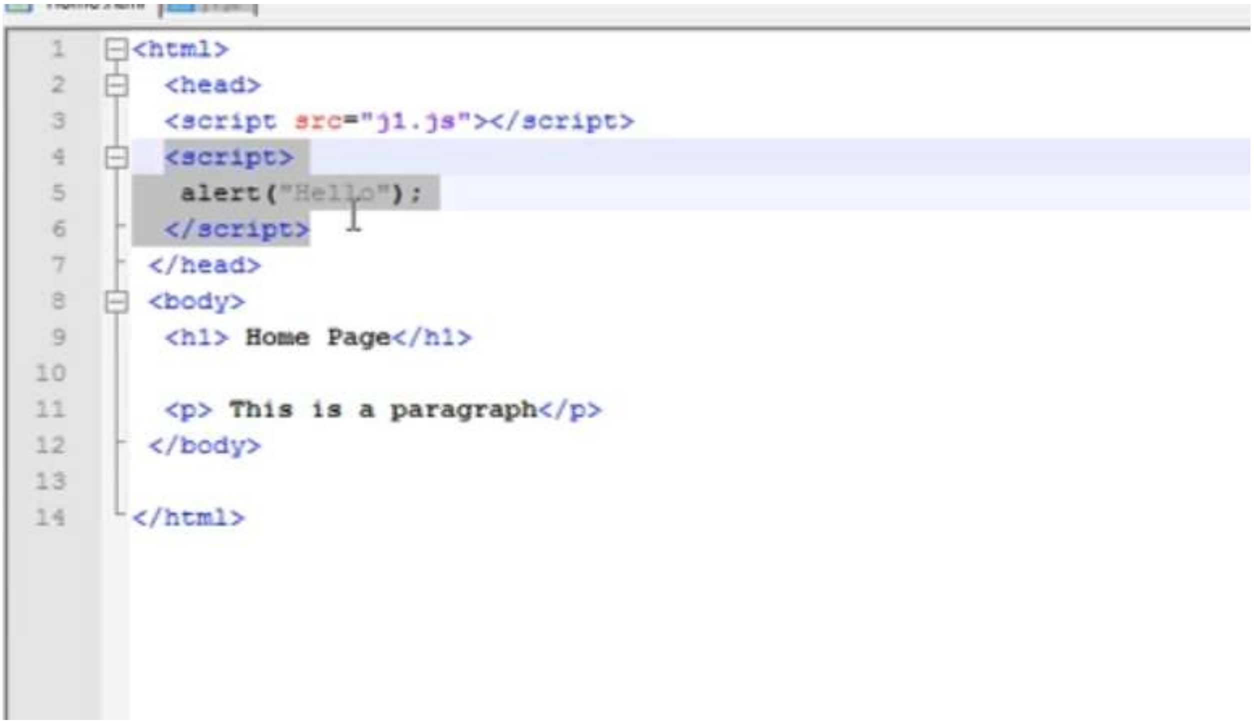


```
Line 1

HOME PAGE

This is a paragraph
```

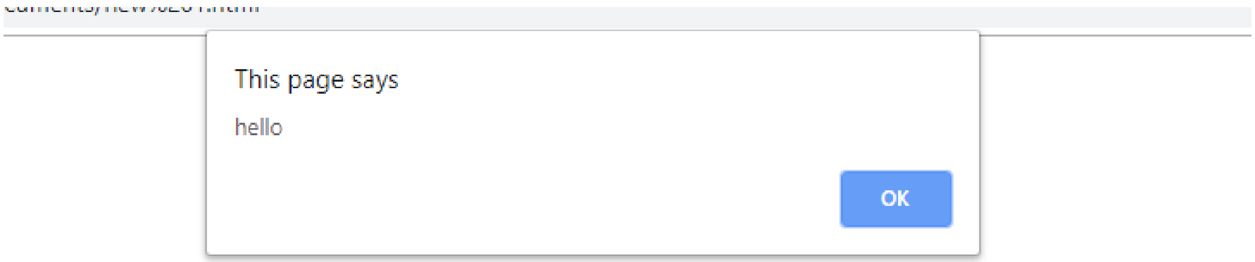
This means that head tag commands run first in every execution


html
j1.js
~~<script>~~


html
j1.js
~~<script>~~
<scriept
src="j1.js



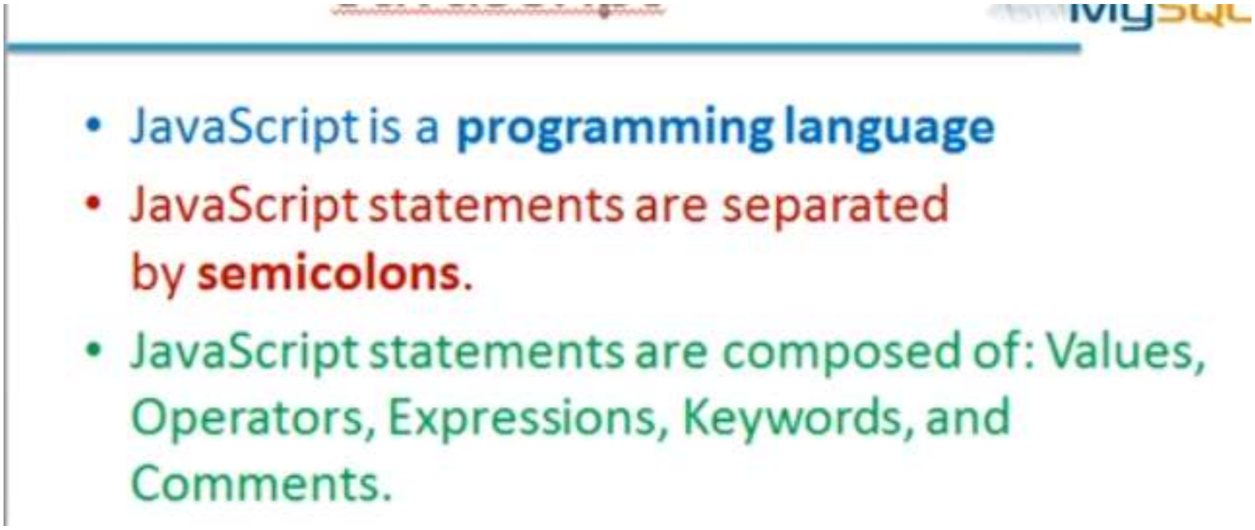
```
<html>
  <head>
  <script src="j1.js"></script>
  <script>
   alert("Hello");
  </script>
 </head>
 <body>
  <h1> Home Page</h1>

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

</html>
```



This page says

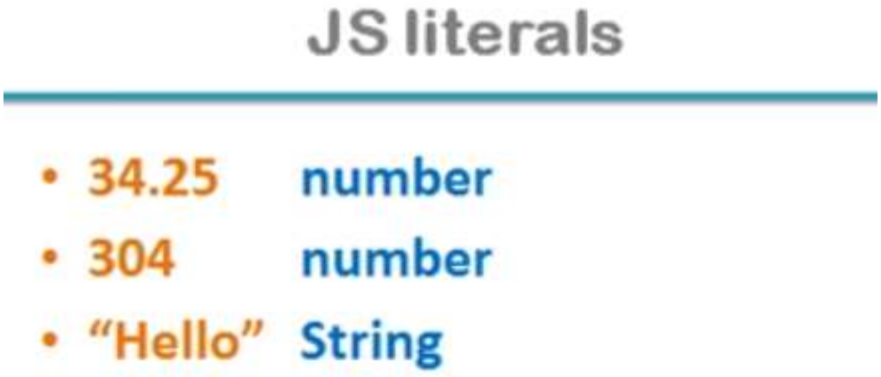
hello

Ise ko html boltey h html decision nh le skta ..is m koi control statement nh mtlb if else or loops nh h but in javascript we have all these features



- JavaScript is a **programming language**
- JavaScript statements are separated by **semicolons**.
- JavaScript statements are composed of: Values, Operators, Expressions, Keywords, and Comments.

Java is influence to c.statement k end m semicolon lagna hota h

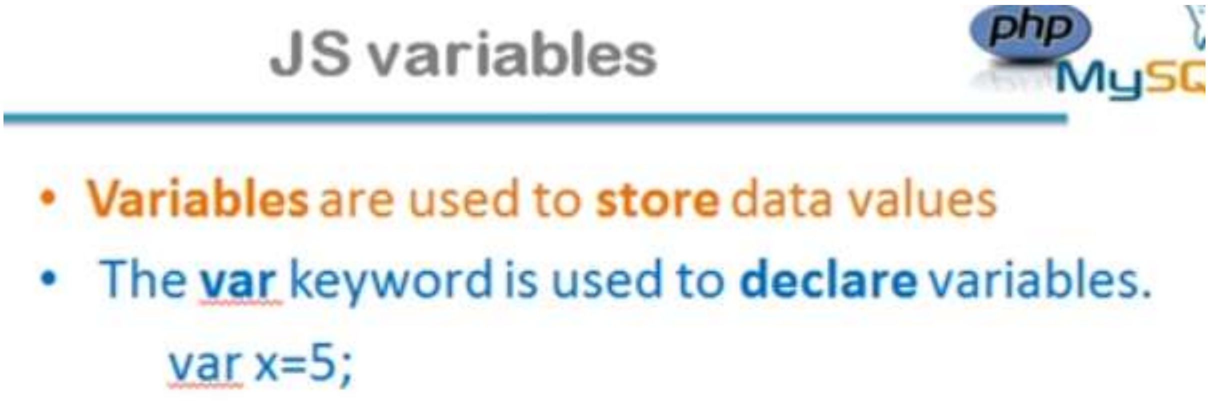


JS literals

• 34.25	number
• 304	number
• “Hello”	String

Constant literal hotey h …number or string dono hosktey h literal is different from variable

Varaible:DATA Sustain krta h jab tk uska scope hota h



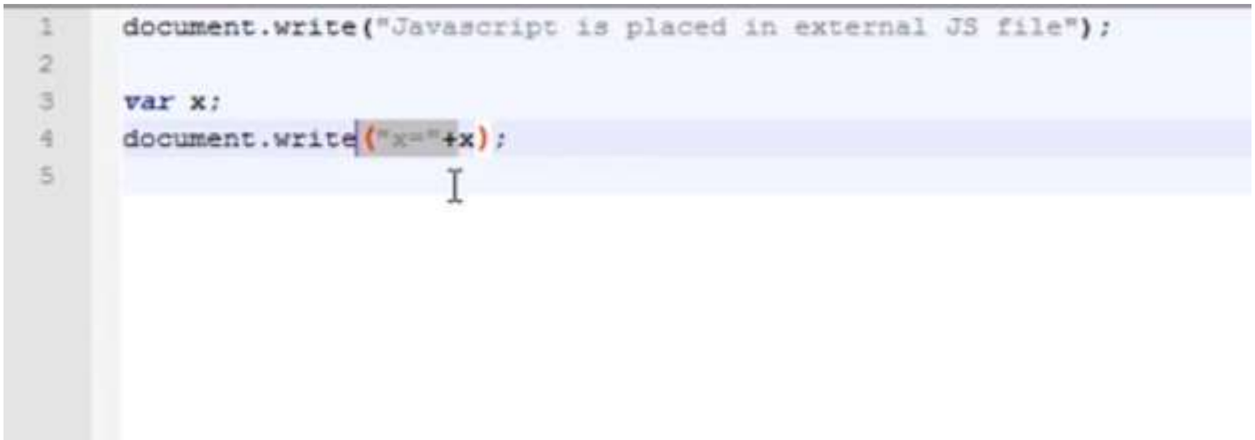
JS variables

- **Variables** are used to **store** data values
- The **var** keyword is used to **declare** variables.

```
var x=5;
```

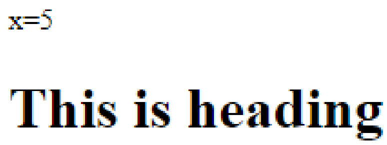
Variable garbage value ka concept nh hota agar intilize nh toh blank value print krega

:



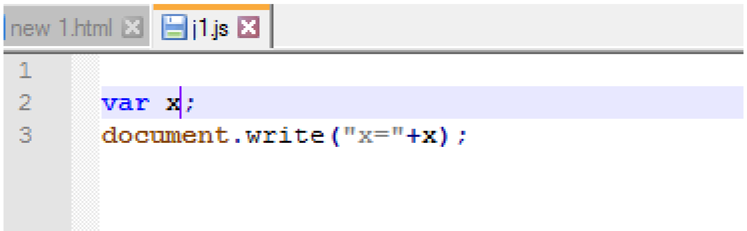
```
document.write("Javascript is placed in external JS file");

var x;
document.write("x="+x);

```

Output:



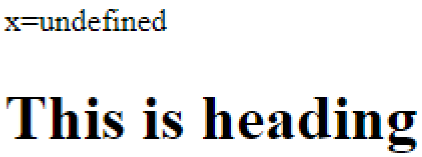
```
x=5
This is heading
```

If we had intilize x so;



```

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

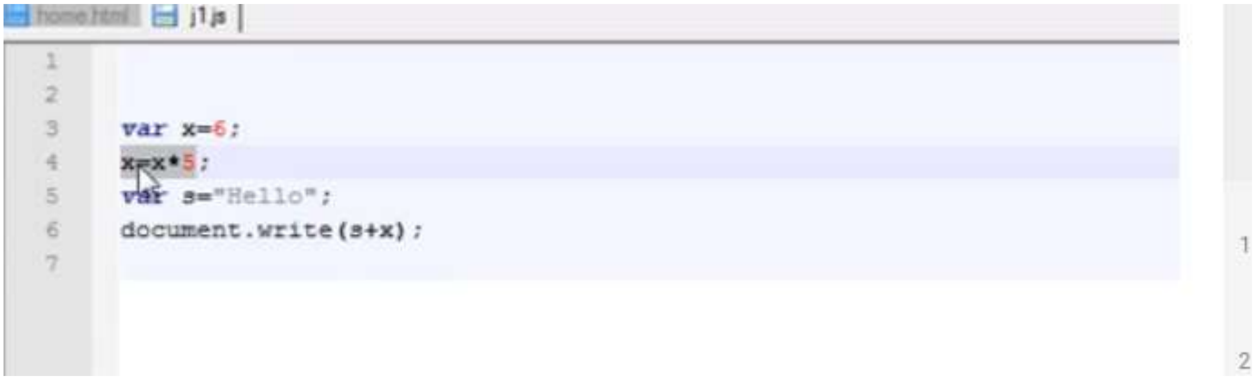
Output



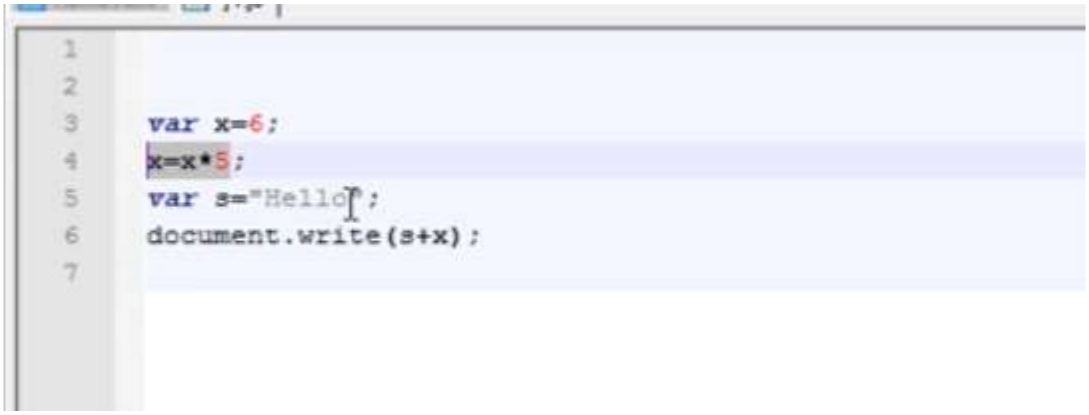
x=undefined

This is heading

Unlike c and c++ hum khud variable ki type nh detey uski value decide krti h

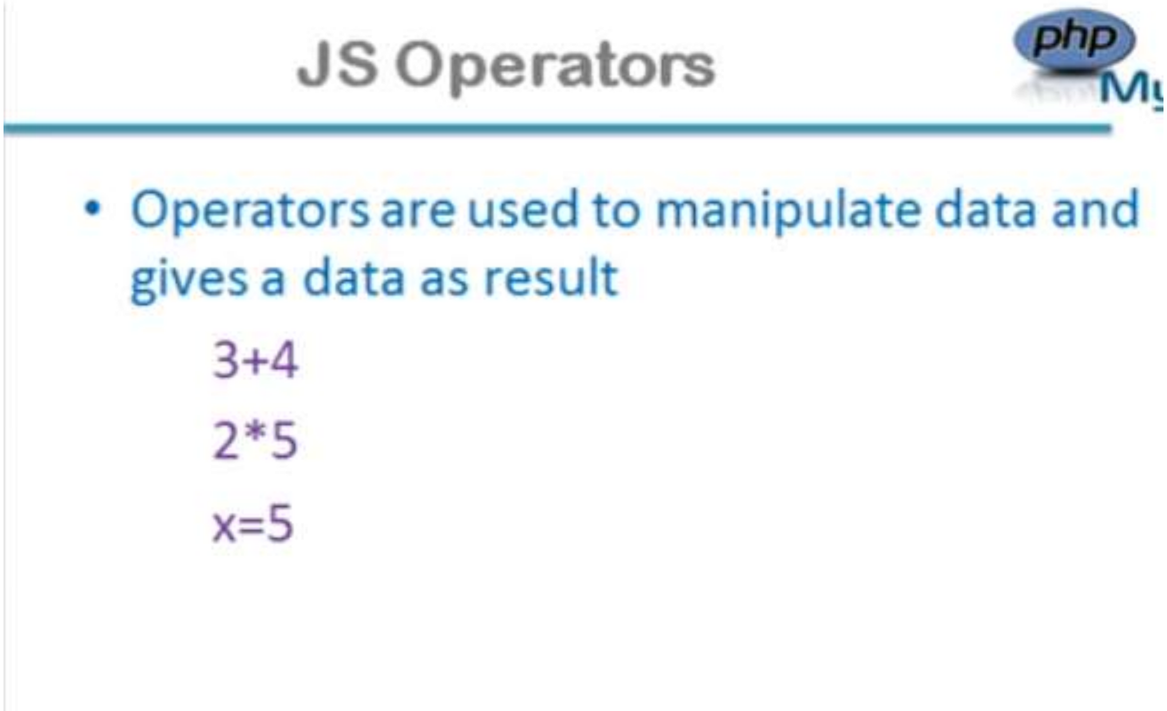


```
var x=6;
x=x*5;
var s="Hello";
document.write(s+x);
```



```
var x=6;
x=x*5;
var s="Hello";
document.write(s+x);
```

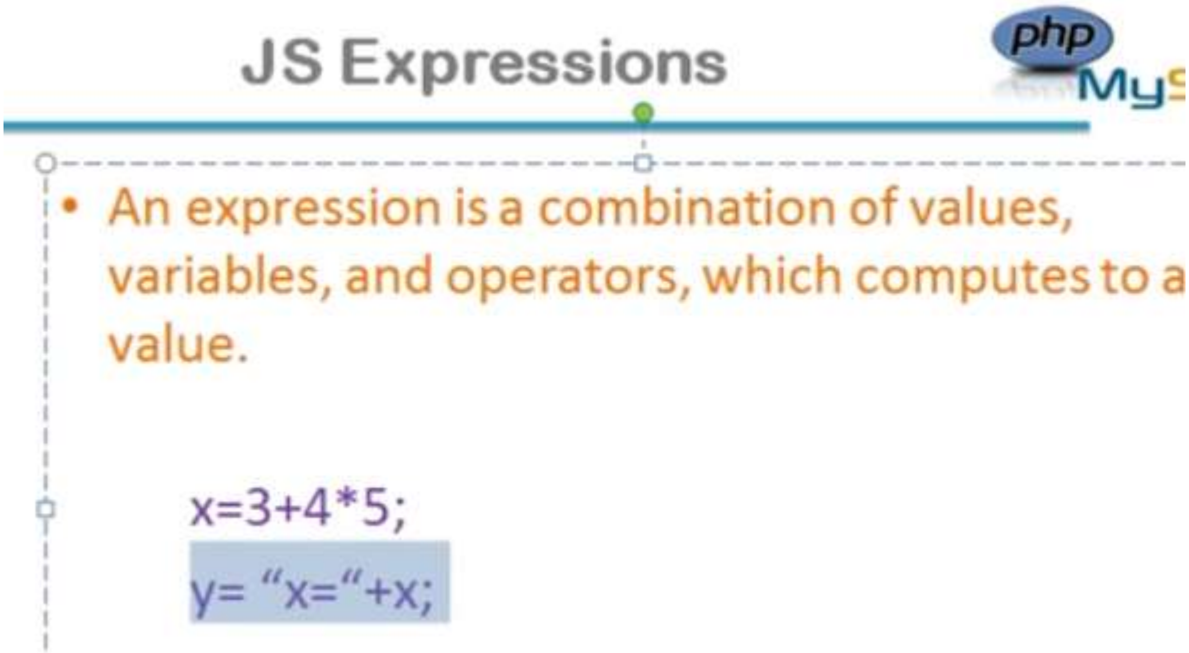
Operator



JS Operators

- Operators are used to manipulate data and gives a data as result

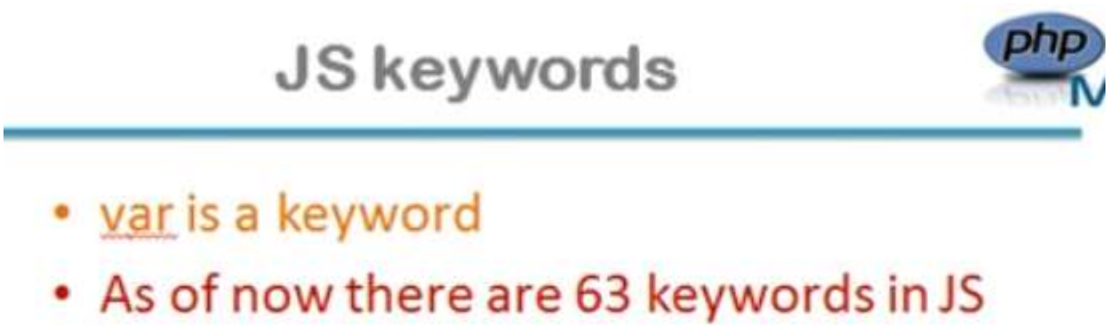
```
3+4
2*5
x=5
```



JS Expressions

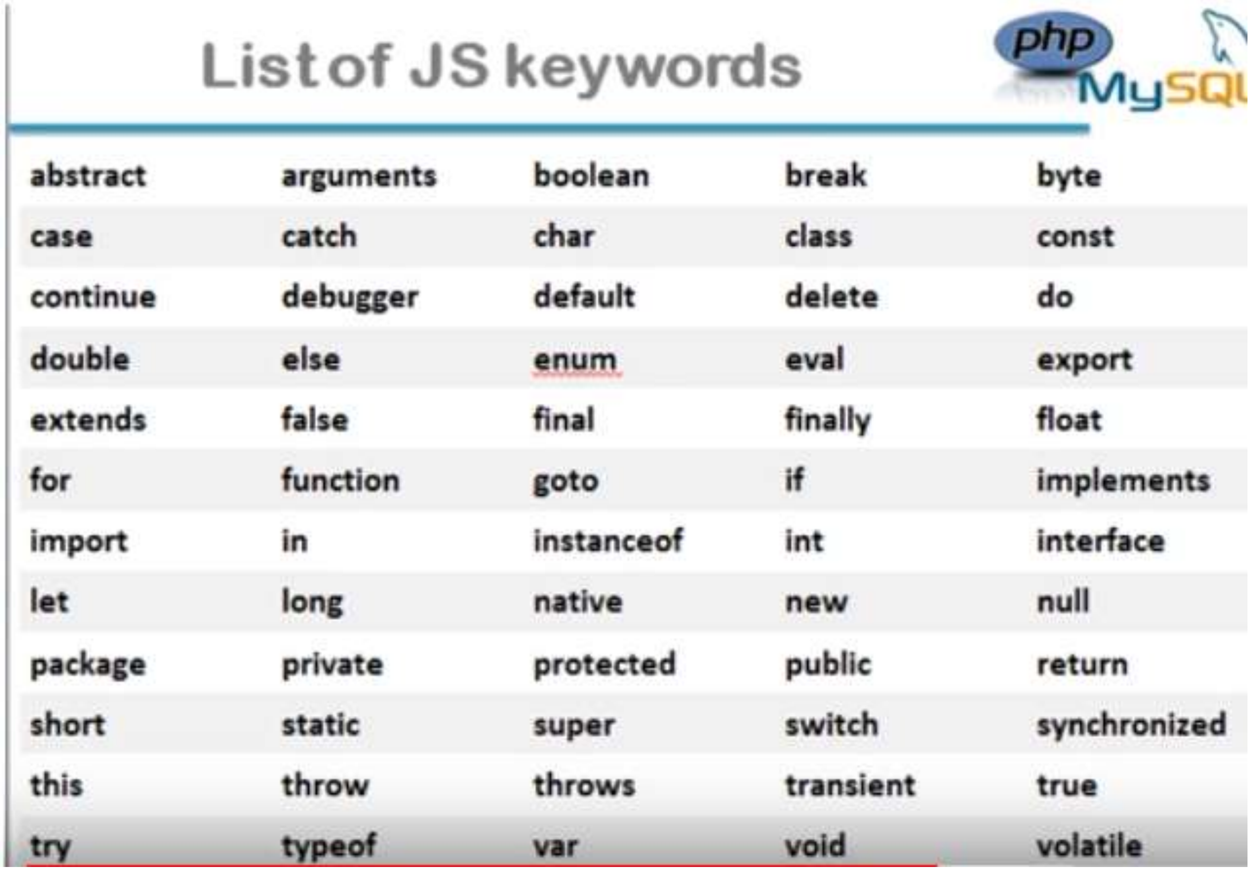
- An expression is a combination of values, variables, and operators, which computes to a value.

```
x=3+4*5;
y= “x=“+x;
```



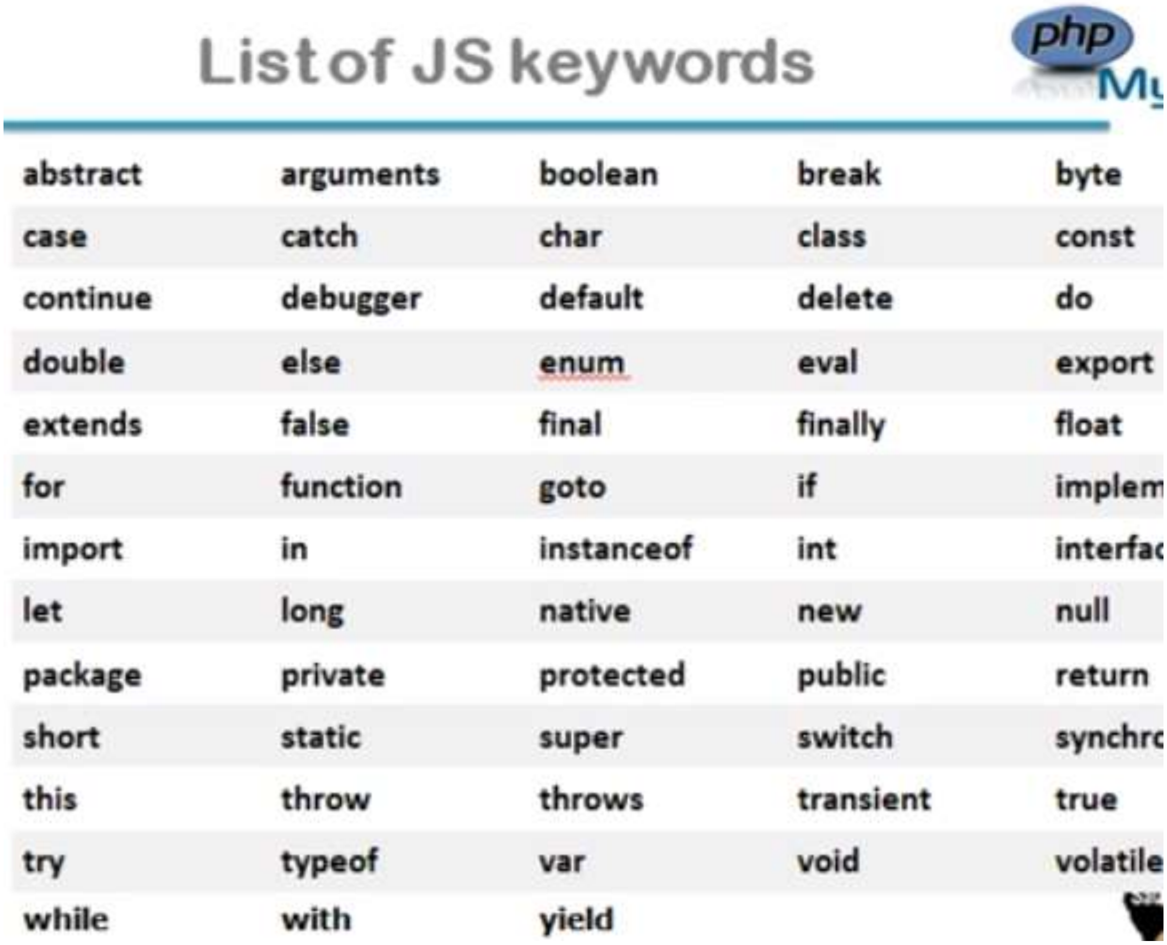
JS keywords

- var is a keyword
- As of now there are 63 keywords in JS



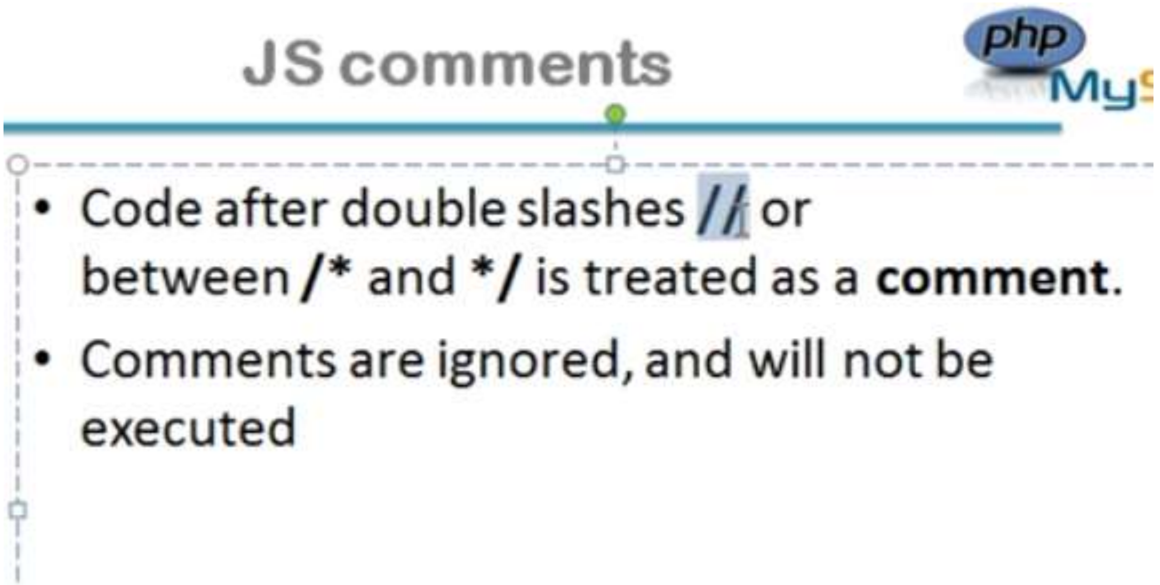
List of JS keywords

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



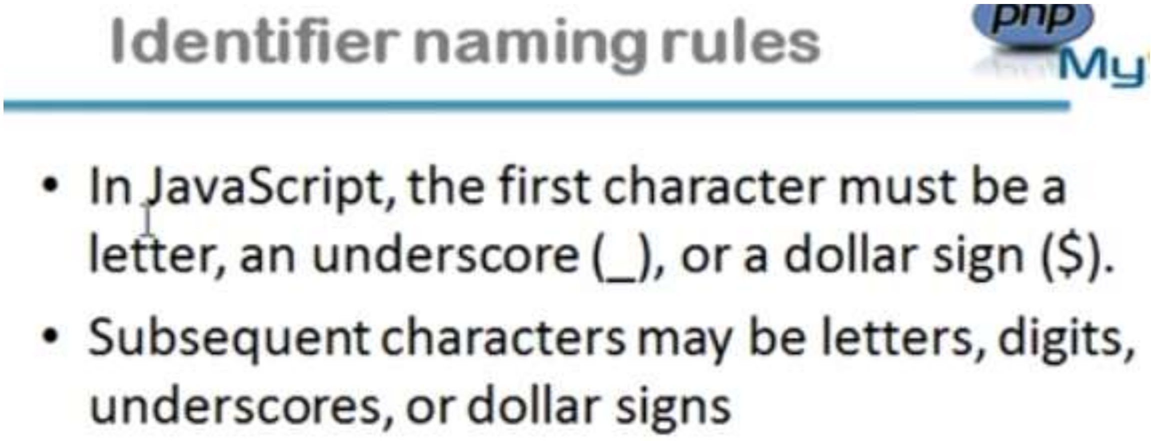
List of JS keywords

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



JS comments

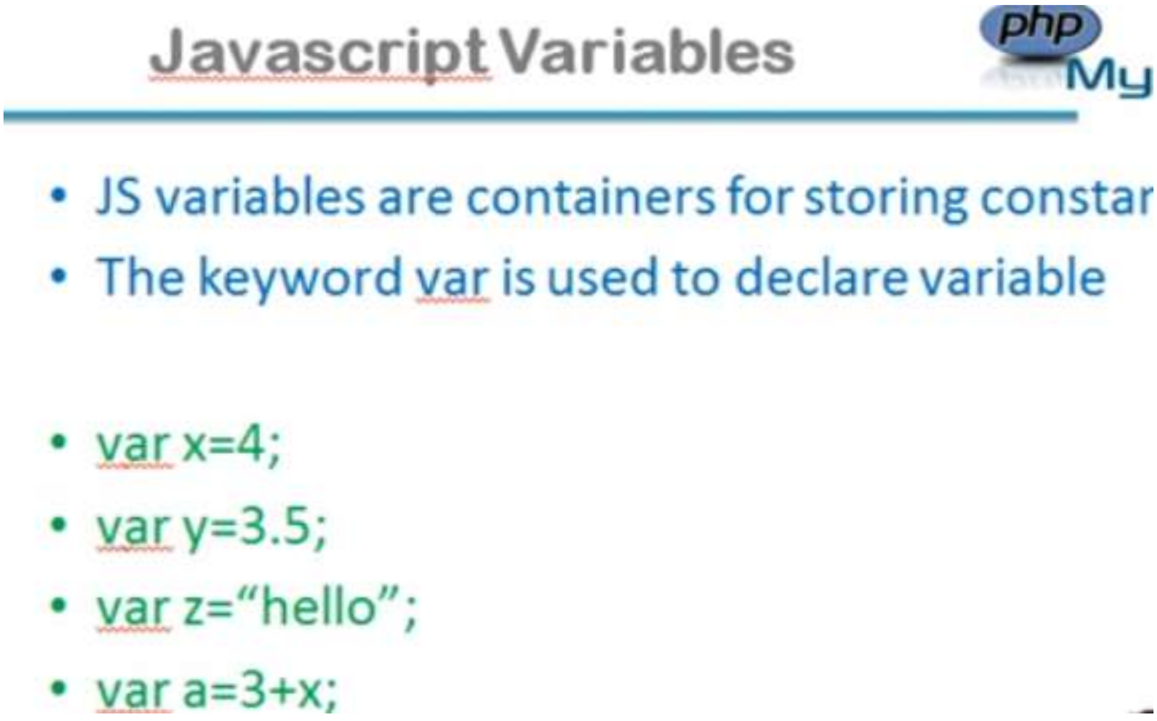
- Code after double slashes // or between **/*** and ***/** is treated as a **comment**.
- Comments are ignored, and will not be executed



Identifier naming rules

- In JavaScript, the first character must be a letter, an underscore (_), or a dollar sign ($).
- Subsequent characters may be letters, digits, underscores, or dollar signs

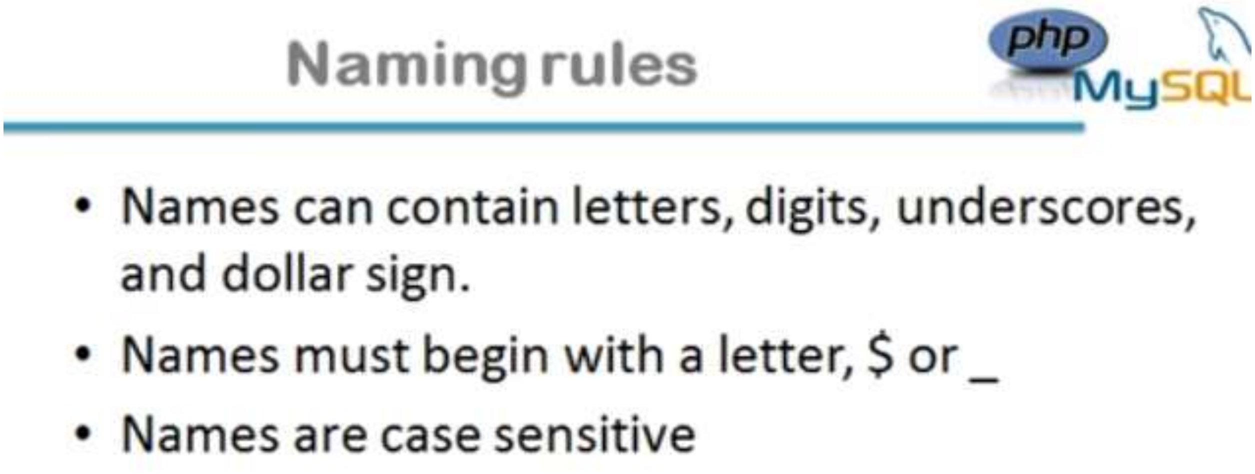
Variables and operators in Javascript



Javascript Variables

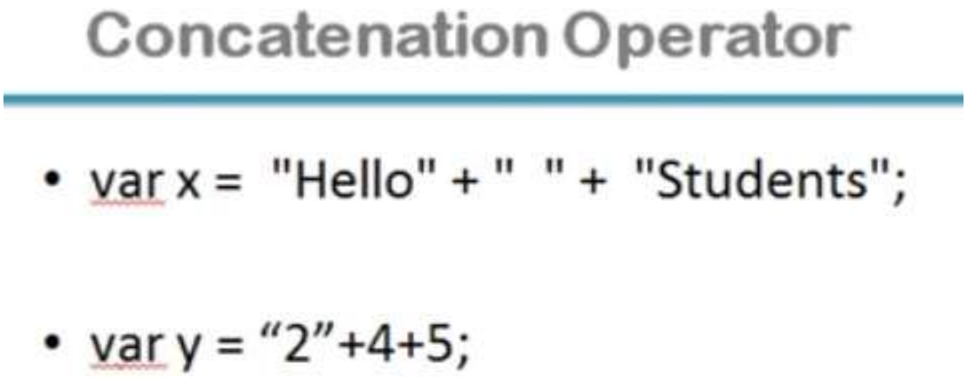
- JS variables are containers for storing constar
- The keyword var is used to declare variable

- var x=4;
- var y=3.5;
- var z="hello";
- var a=3+x;



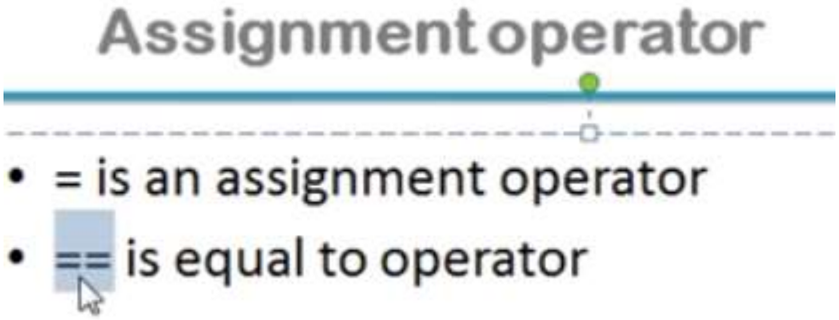
Naming rules

- Names can contain letters, digits, underscores, and dollar sign.
- Names must begin with a letter, $ or _
- Names are case sensitive



Concatenation Operator

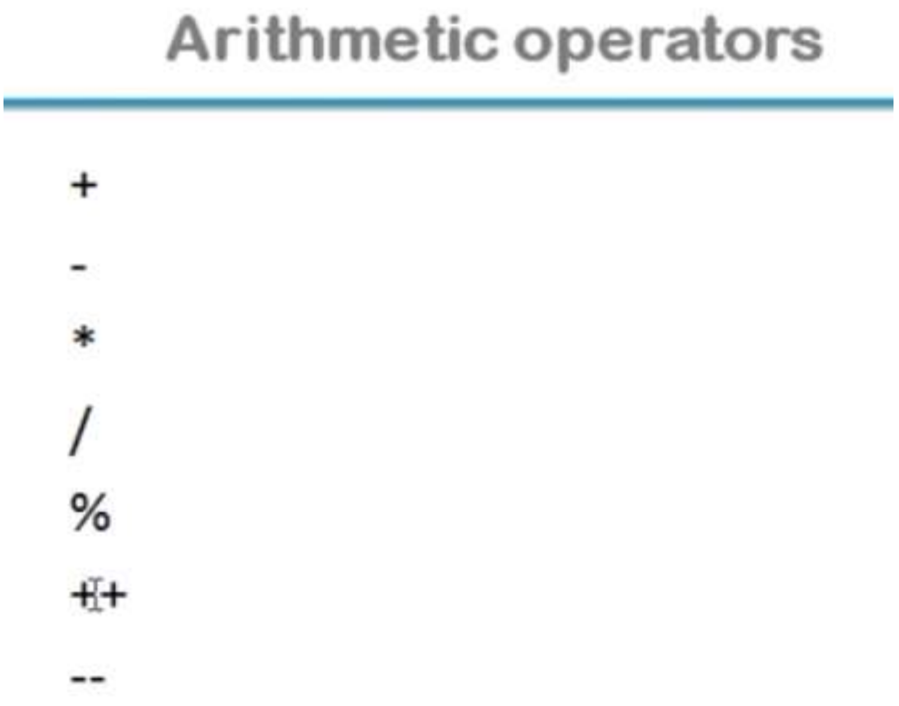
- var x = "Hello" + "   " + "Students";
- var y = “2”+4+5;



Assignment operator

- = is an assignment operator
- == is equal to operator

\



Arithmetic operators

+

-

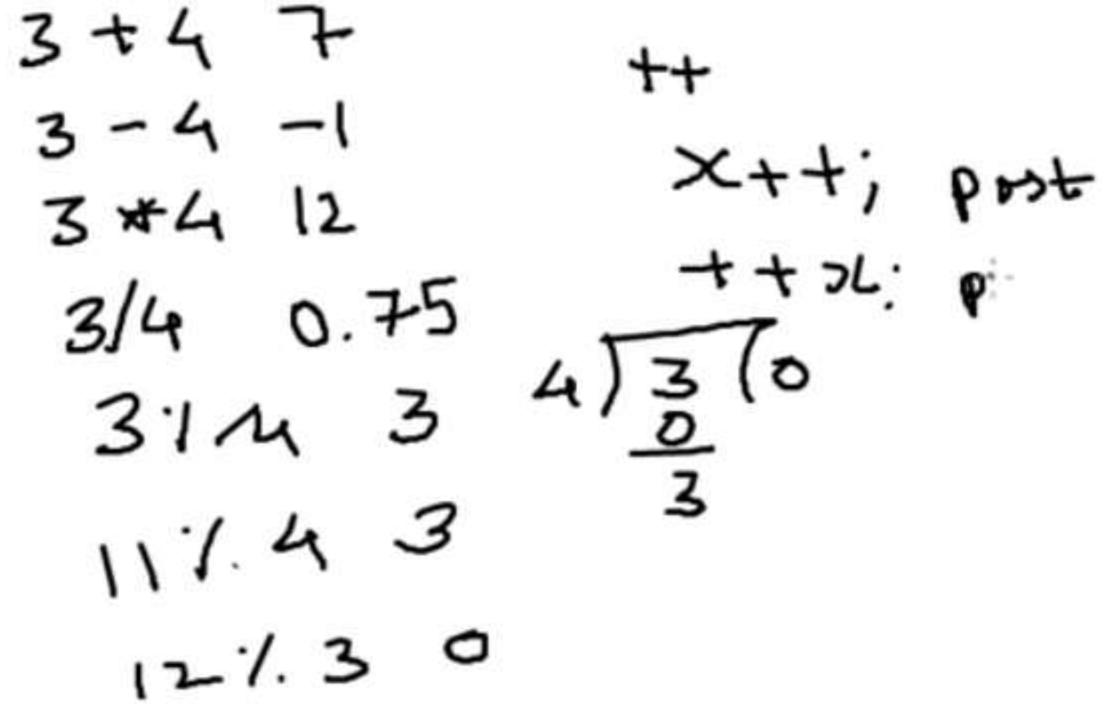
*

/

%

++

--



3 + 4   7

3 - 4   -1

3 * 4   12

3/4   0.75

3 % 4   3

11 % 4   3

12 % 3   0

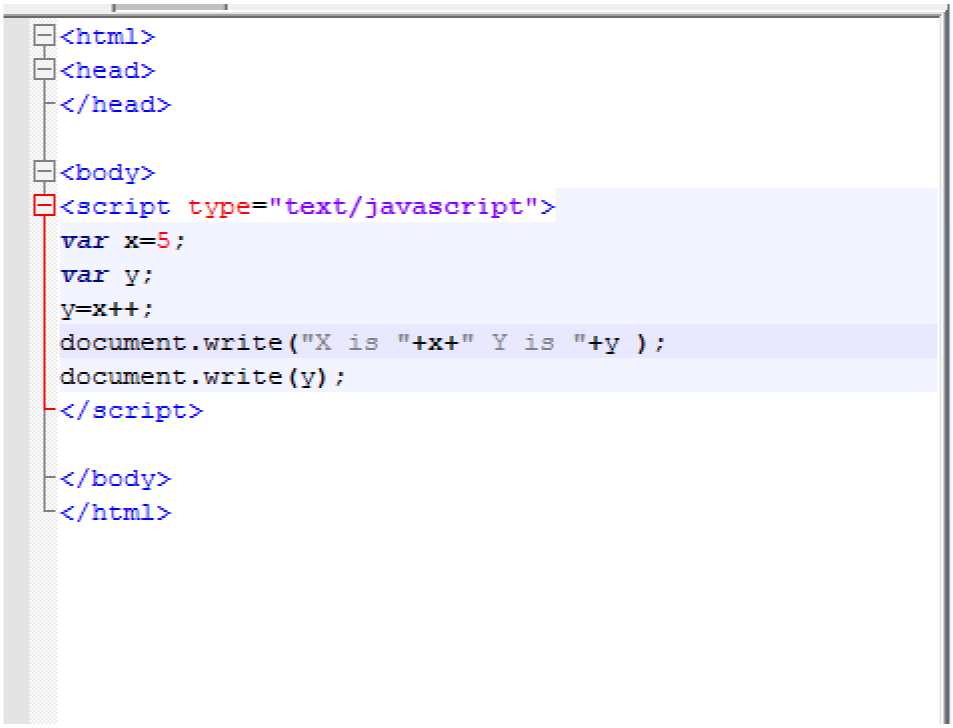
++

x++; post

++x; p

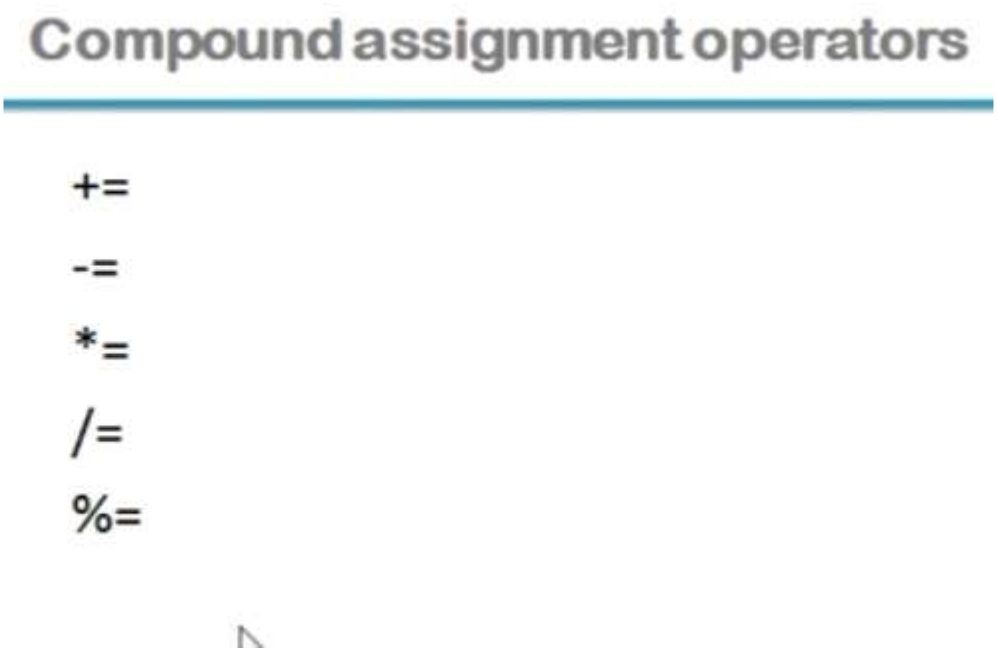
4) 3 (0
0
3

Pre-increment priority is greater unlike post-increment.



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

<body>
<script type="text/javascript">
var x=5;
var y;
y=x++;
document.write("X is "+x+" Y is "+y );
document.write(y);
</script>

</body>
</html>
```

```
X is 6 Y is 55
```



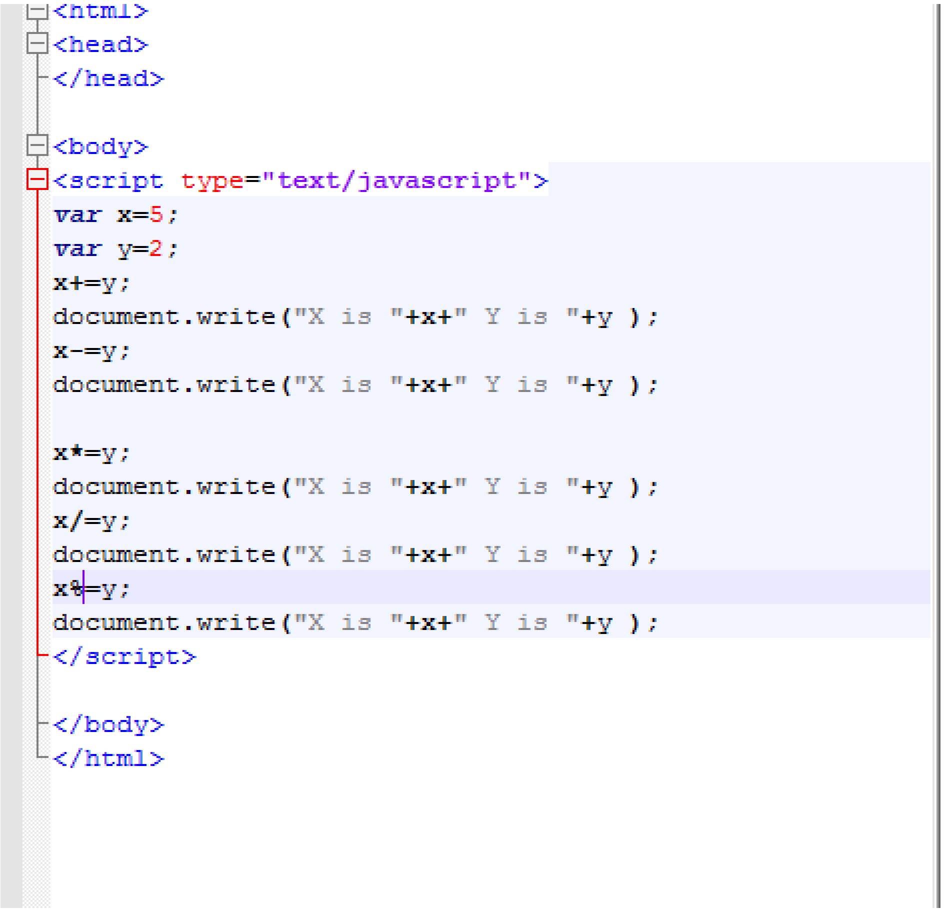
Compound assignment operators

- +=
- -=
- *=
- /=
- %=

Yahn se phle assignment hogi like x ki value y m chle jayegi then x increment hoga

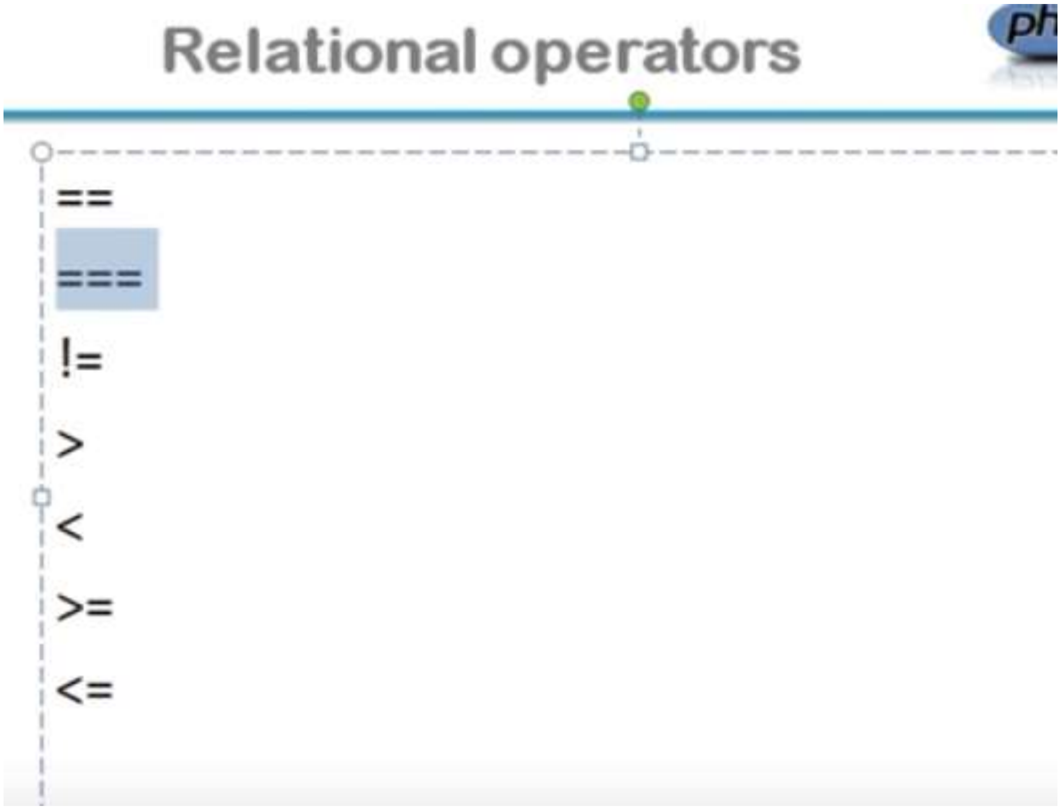
$x += 4$ $x = x+4$

$x *= 3$ $x = x*3$



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

<body>
<script type="text/javascript">
var x=5;
var y=2;
x+=y;
document.write("X is "+x+" Y is "+y );
x-=y;
document.write("X is "+x+" Y is "+y );

x*=y;
document.write("X is "+x+" Y is "+y );
x/=y;
document.write("X is "+x+" Y is "+y );
x%=y;
document.write("X is "+x+" Y is "+y );
</script>

</body>
</html>
```

```
X is 7 Y is 2X is 5 Y is 2X is 10 Y is 2X is 5 Y is 2X is 1 Y is 2
```



Relational operators

==

===

!=

>

<

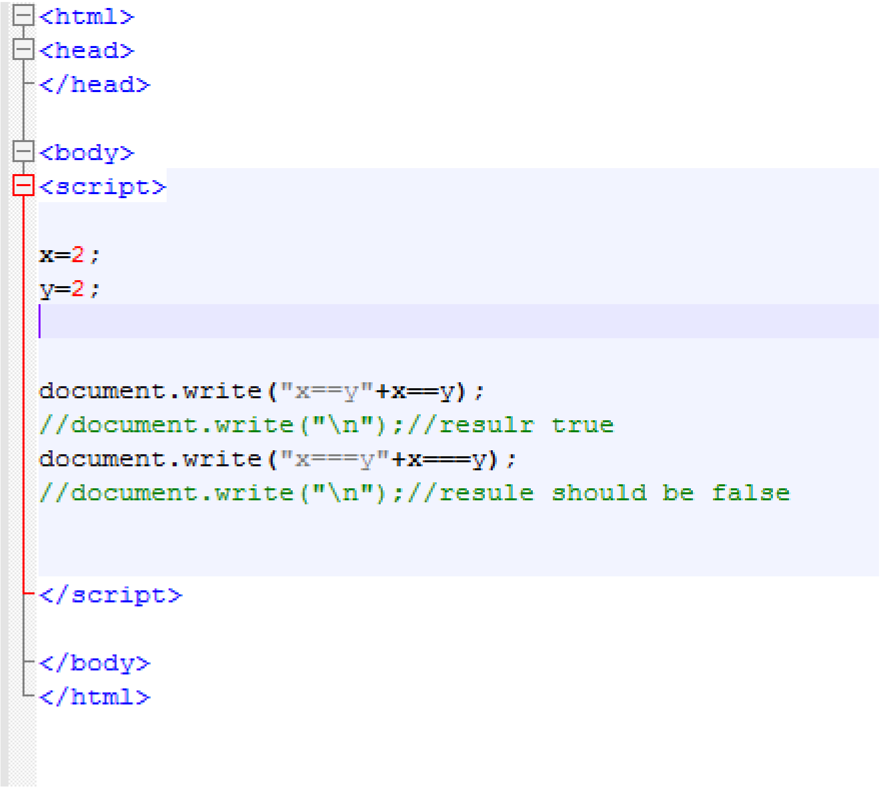
>=

<=

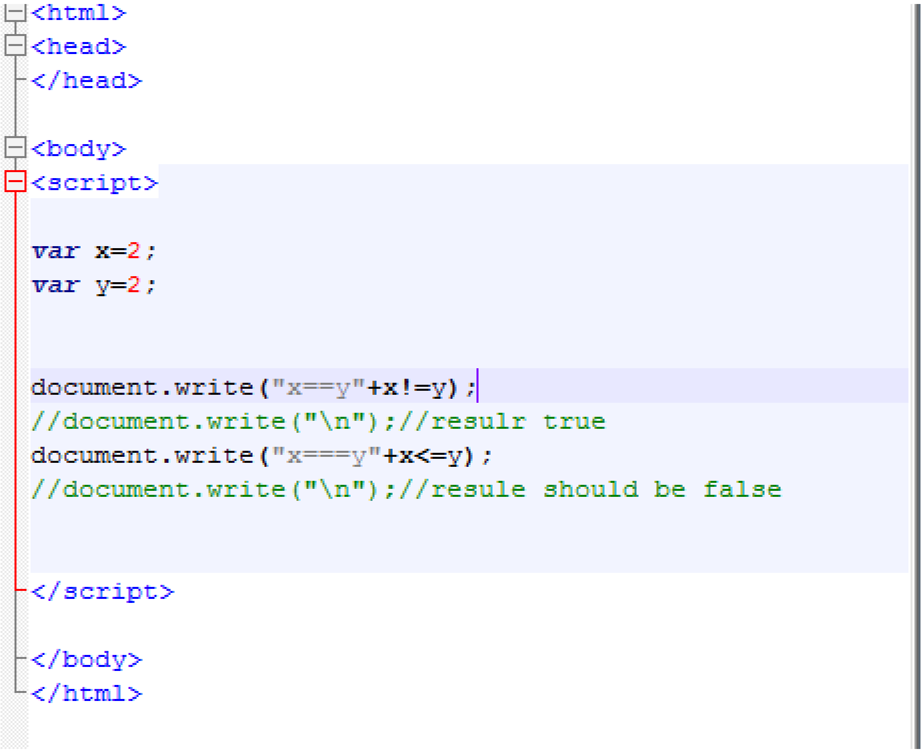
=:for assignment

==:for equality

===:Check phase as well as type

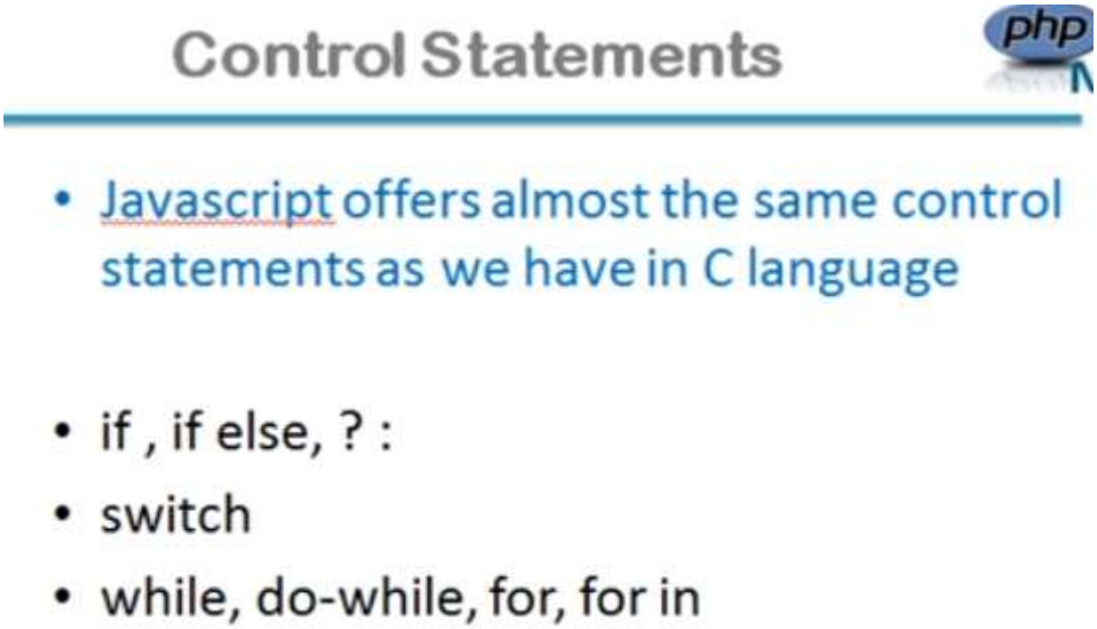


```
<html>
<head>
</head>

<body>
<script>

x=2;
y=2;


document.write("x==y"+x==y);
//document.write("\n");//resulr true
document.write("x===y"+x===y);
//document.write("\n");//resule should be false


</script>

</body>
</html>
```



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

<body>
<script>

var x=2;
var y=2;


document.write("x==y"+x!=y);
//document.write("\n");//resulr true
document.write("x===y"+x<=y);
//document.write("\n");//resule should be false


</script>

</body>
</html>
```

Control Statements in Javascript

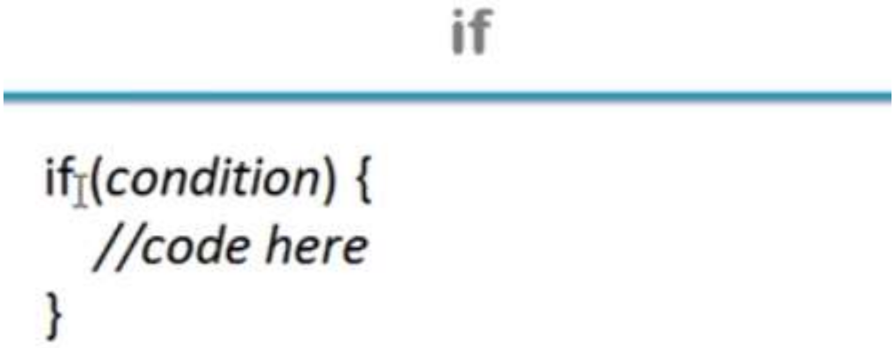


Control Statements

- Javascript offers almost the same control statements as we have in C language

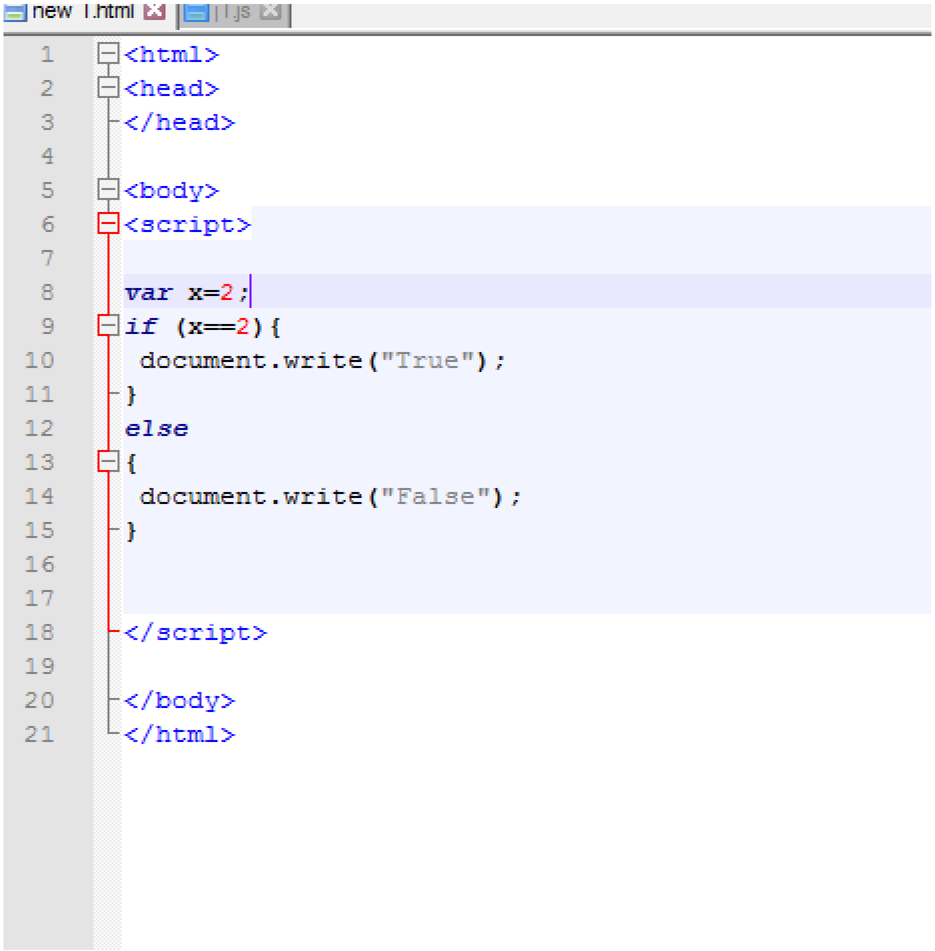
- if , if else, ? :
- switch
- while, do-while, for, for in

Syntax:



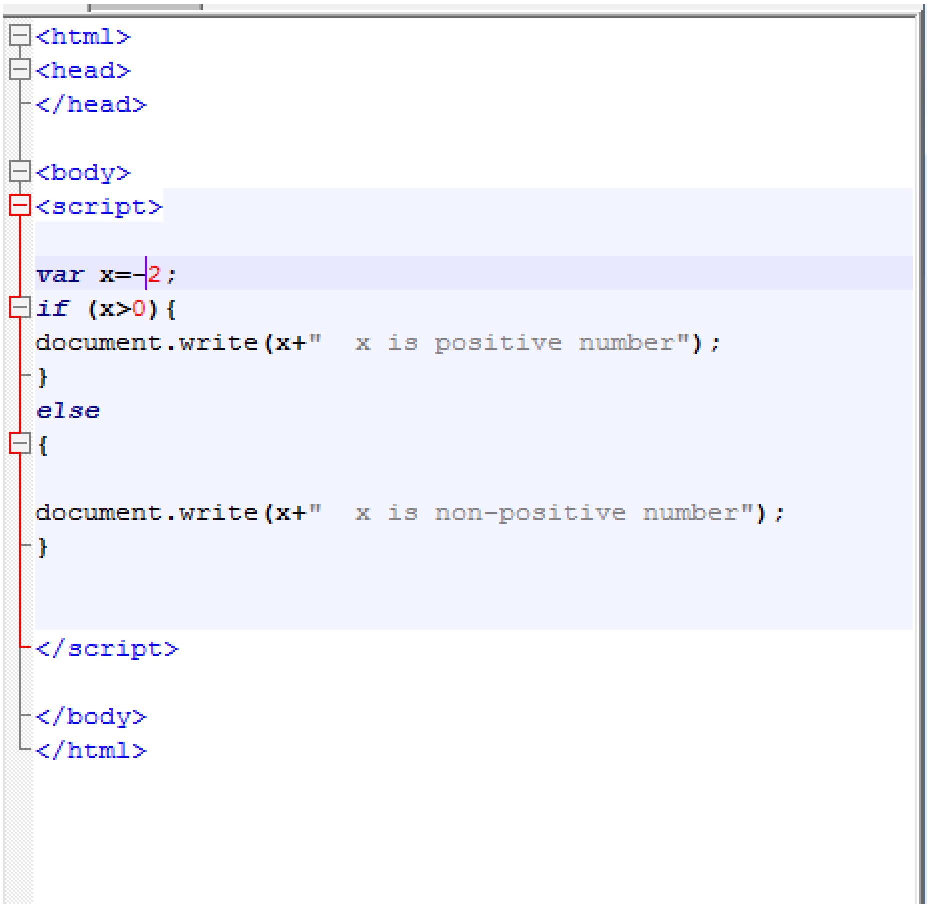
if

```
if (condition) {
   //code here
}
```

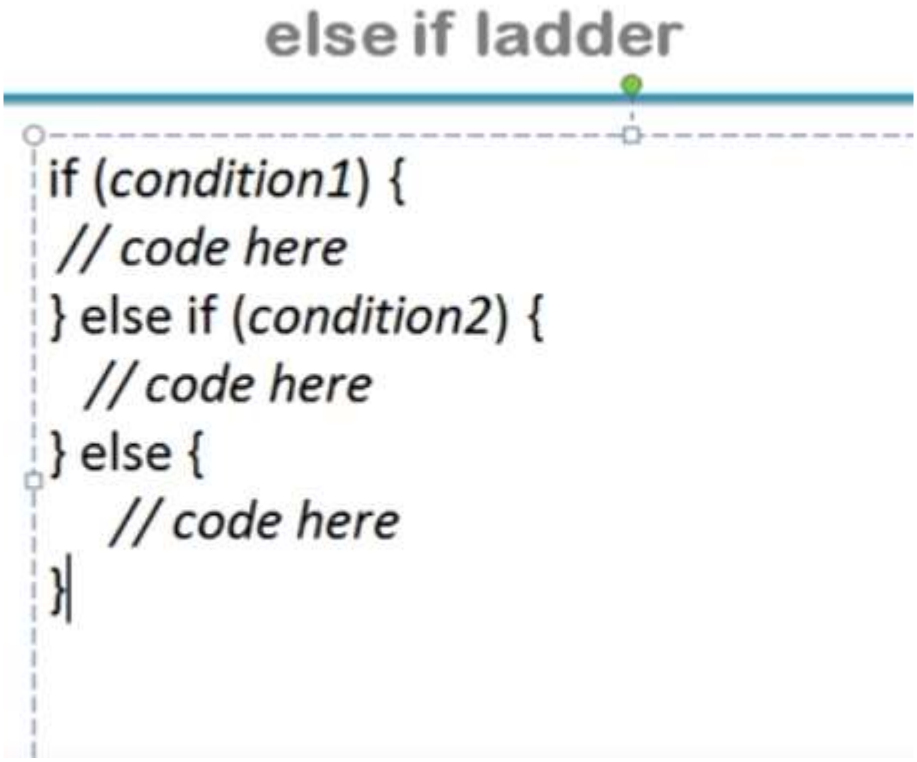


```
<html>
<head>
</head>

<body>
<script>

var x=2;
if (x==2){
 document.write("True");
}
else
{
 document.write("False");
}


</script>

</body>
</html>
```

Output:



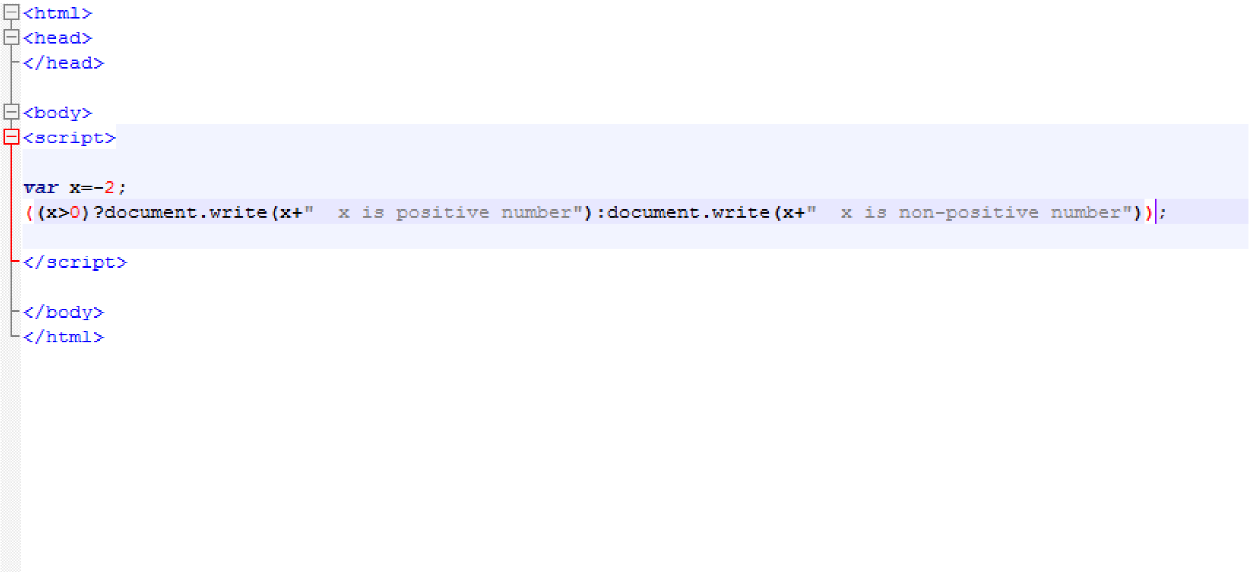
```
<html>
<head>
</head>

<body>
<script>

var x=-2;
if (x>0){
document.write(x+"  x is positive number");
}
else
{

document.write(x+"  x is non-positive number");
}


</script>

</body>
</html>
```



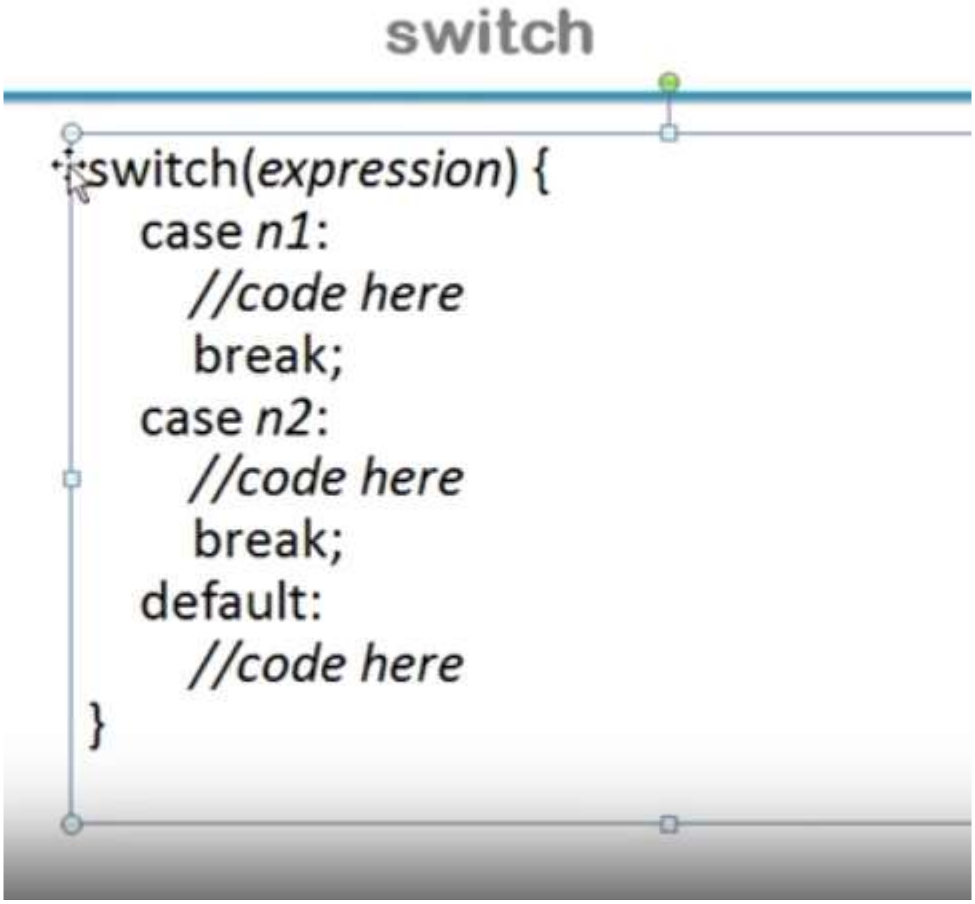
else if ladder

```
if (condition1) {
 // code here
} else if (condition2) {
   // code here
} else {
     // code here
}
```

Conditional Operator:



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

<body>
<script>

var x=-2;
((x>0)?document.write(x+"  x is positive number"):document.write(x+"  x is non-positive number"));

</script>

</body>
</html>
```

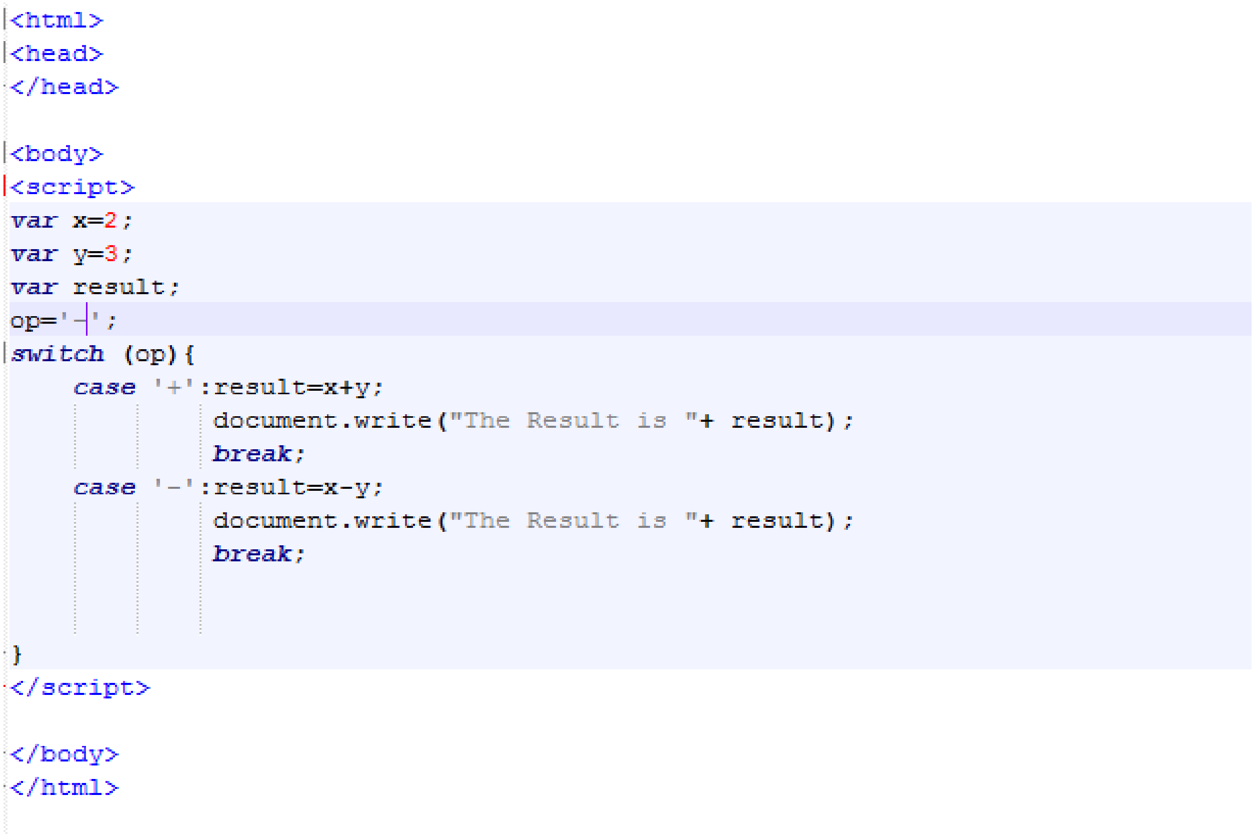


switch

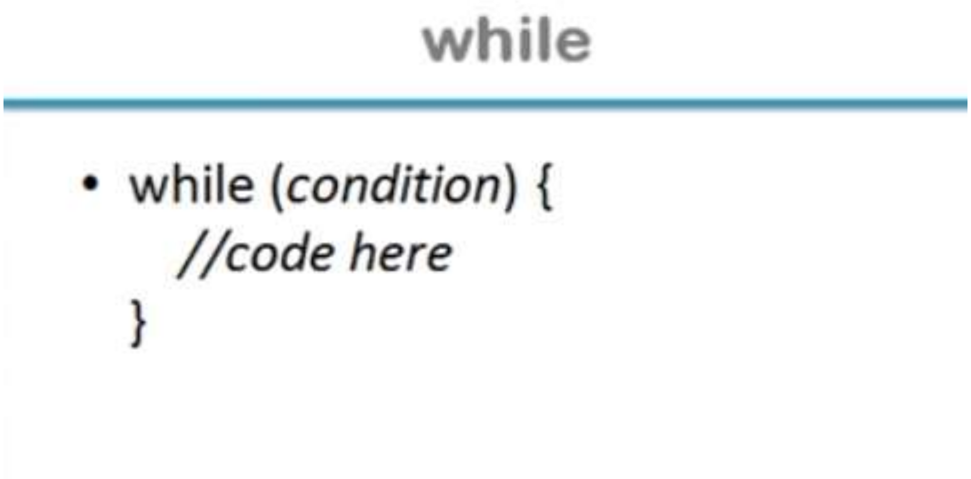
```
switch(expression) {
    case n1:
        //code here
        break;
    case n2:
        //code here
        break;
    default:
        //code here
}
```

Expression ka Iska result case ki

se match ki jayegi



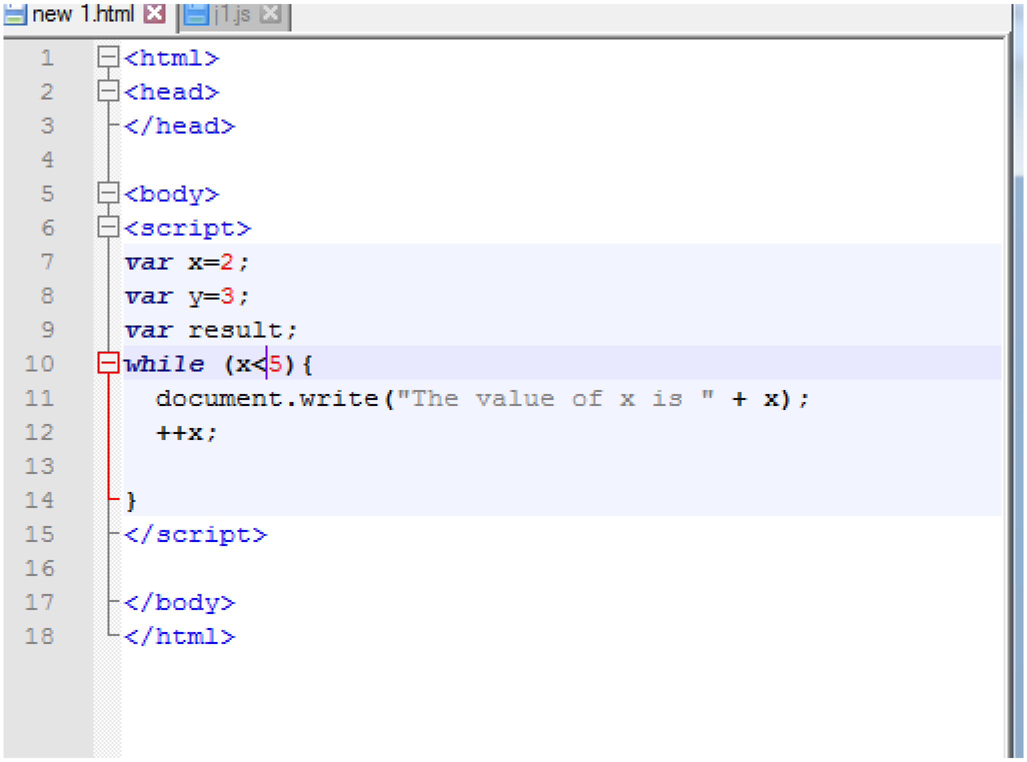
```
<html>
<head>
</head>

<body>
<script>
var x=2;
var y=3;
var result;
op='-';
switch (op){
    case '+':result=x+y;
             document.write("The Result is "+ result);
             break;
    case '-':result=x-y;
             document.write("The Result is "+ result);
             break;

}
</script>

</body>
</html>
```



while

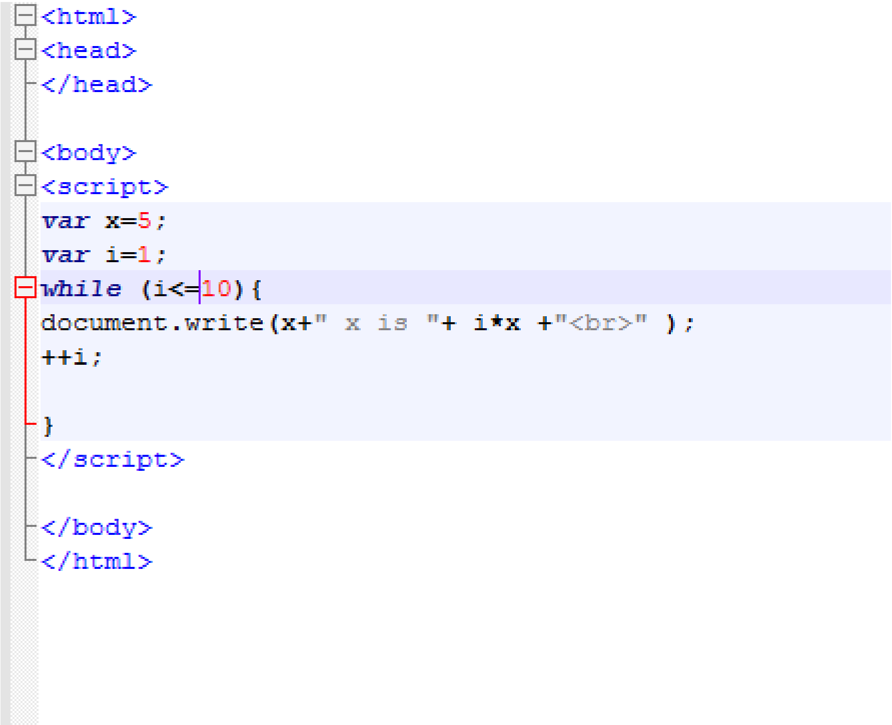
- while (*condition*) {
      //code here
  }

While Example 1



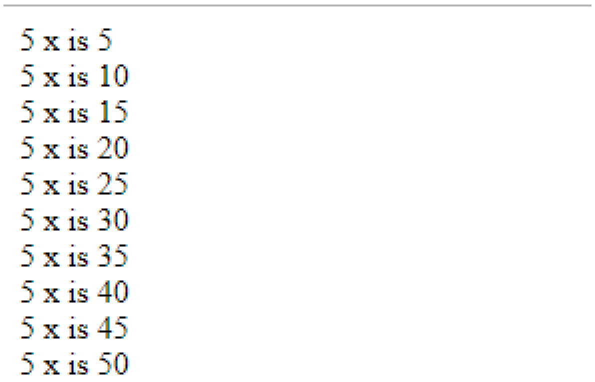
```
<html>
<head>
</head>

<body>
<script>
var x=2;
var y=3;
var result;
while (x<5){
  document.write("The value of x is " + x);
  ++x;

}
</script>

</body>
</html>
```

While Example 2



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

<body>
<script>
var x=5;
var i=1;
while (i<=10){
document.write(x+" x is "+ i*x +"<br>" );
++i;

}
</script>

</body>
</html>
```

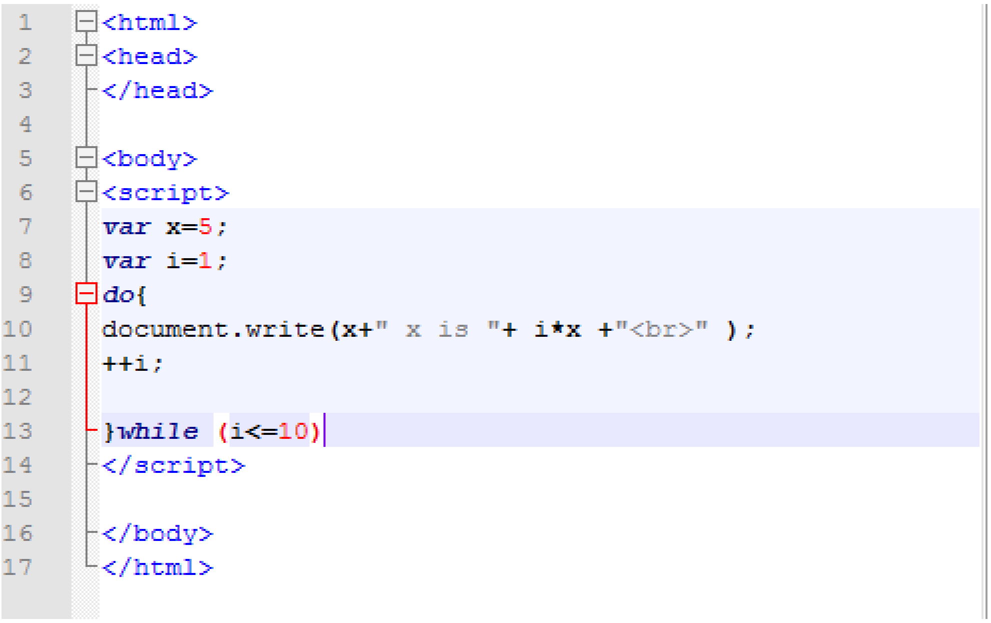
Output:



```
5 x is 5
5 x is 10
5 x is 15
5 x is 20
5 x is 25
5 x is 30
5 x is 35
5 x is 40
5 x is 45
5 x is 50
```

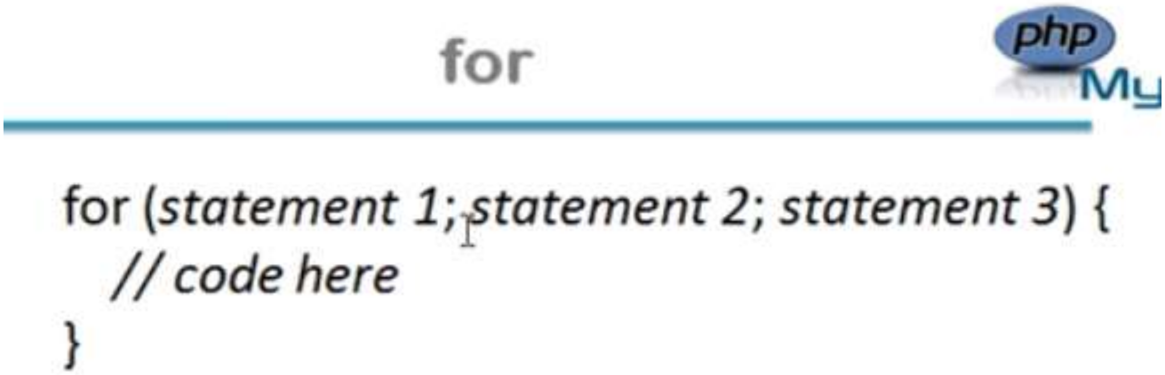
Do While:

Example no 1:



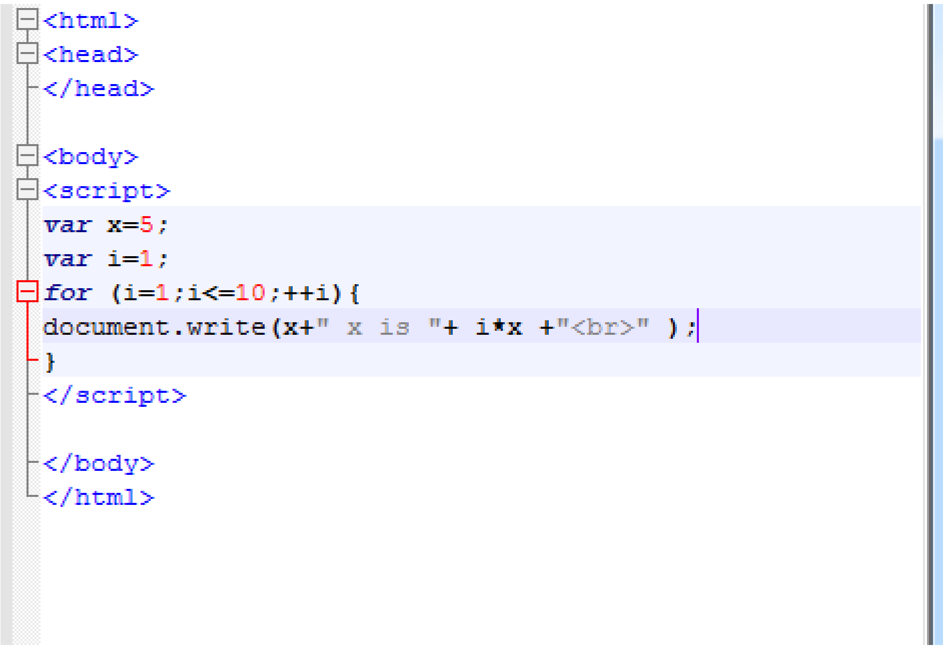
```
<html>
<head>
</head>

<body>
<script>
var x=5;
var i=1;
do{
document.write(x+" x is "+ i*x +"<br>" );
++i;

}while (i<=10)
</script>

</body>
</html>
```

For Loop



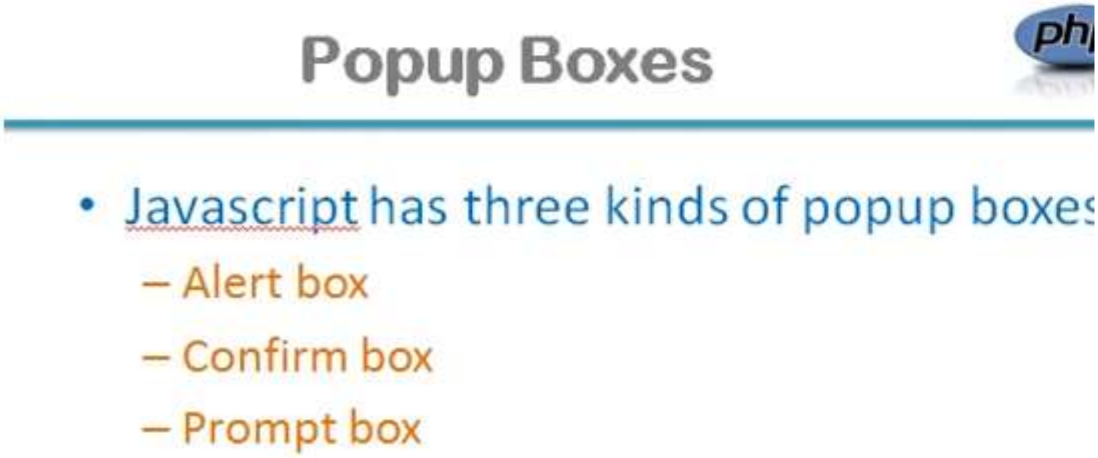
for

```
for (statement 1; statement 2; statement 3) {
   // code here
}
```



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

<body>
<script>
var x=5;
var i=1;
for (i=1;i<=10;++i){
document.write(x+" x is "+ i*x +"<br>" );
}
</script>

</body>
</html>
```

Popup Boxes in Javascript



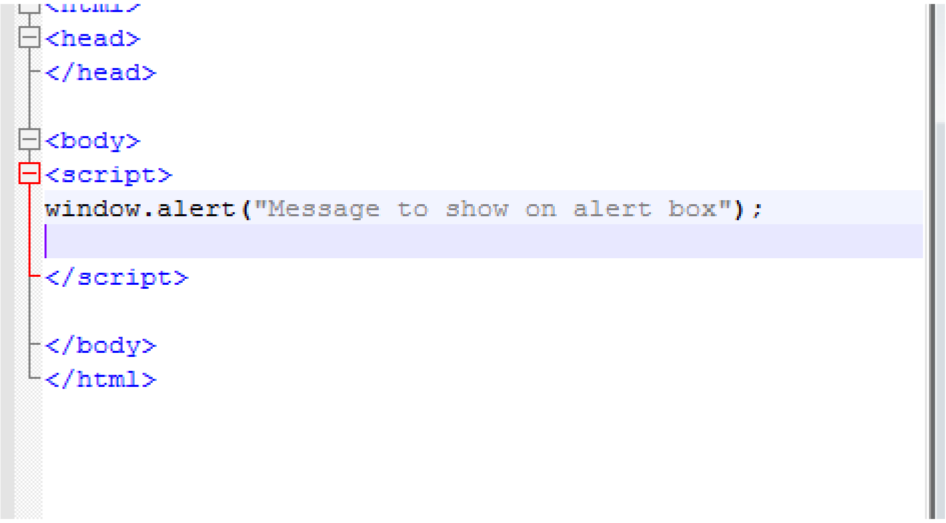
Popup Boxes

- Javascript has three kinds of popup boxes
  - Alert box
  - Confirm box
  - Prompt box

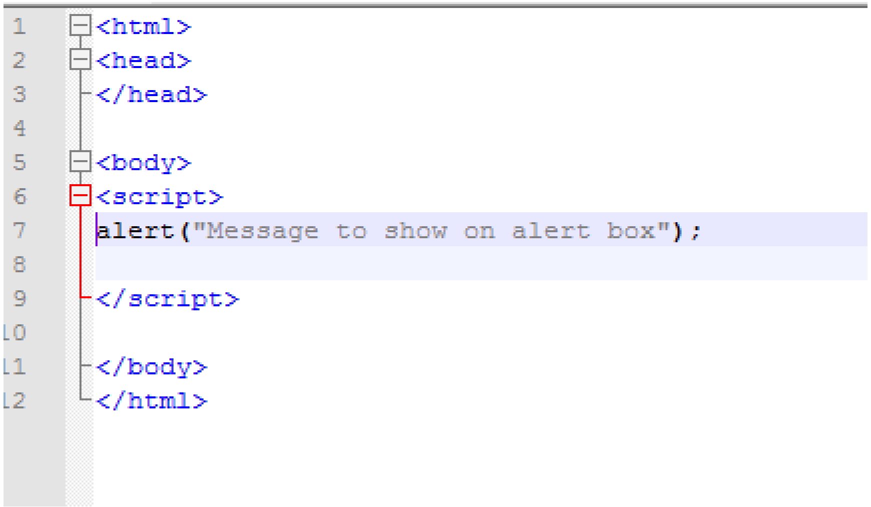
It if from library name windows so we can also write as

Syntax:

Window.alert("");

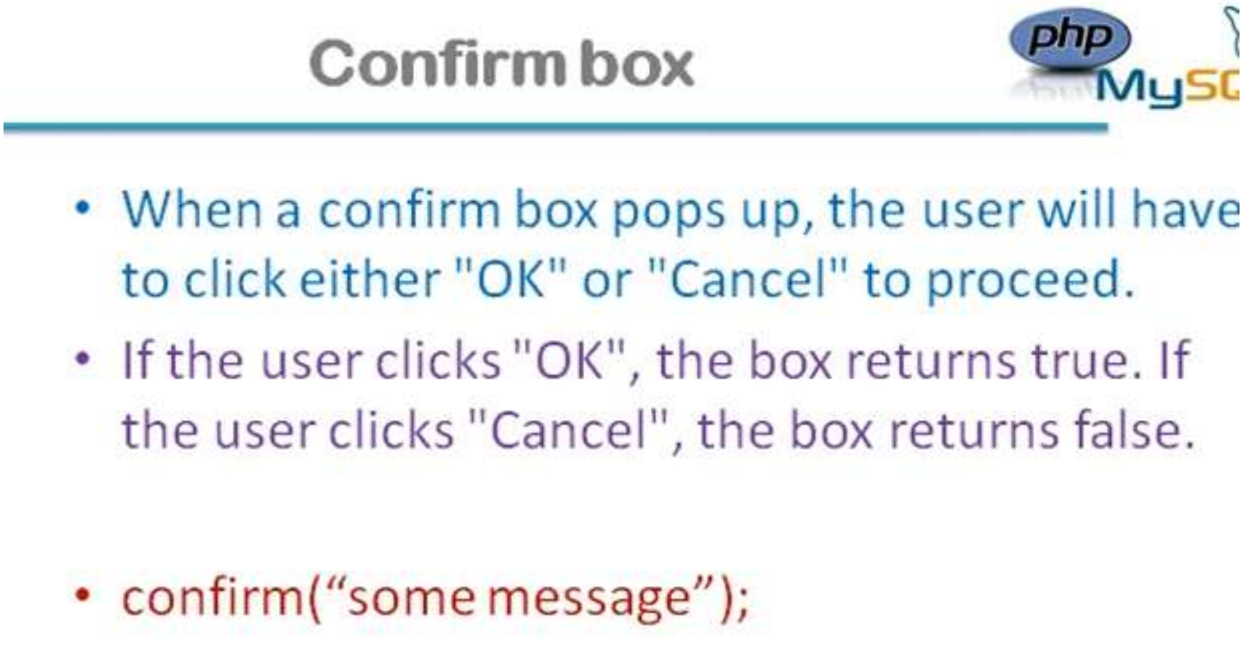


```
<html>
<head>
</head>

<body>
<script>
window.alert("Message to show on alert box");

</script>

</body>
</html>
```



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

<body>
<script>
alert("Message to show on alert box");

</script>

</body>
</html>
```

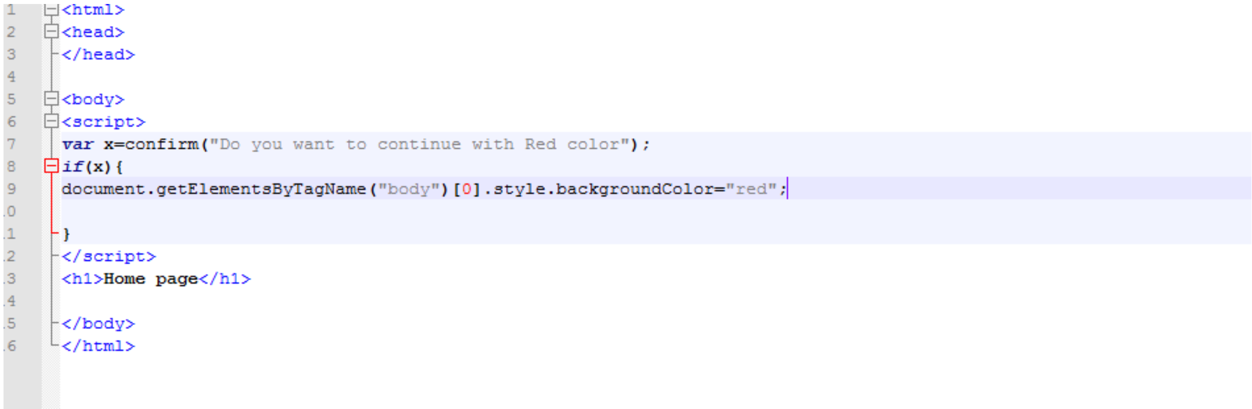
Confirm box:



Confirm box

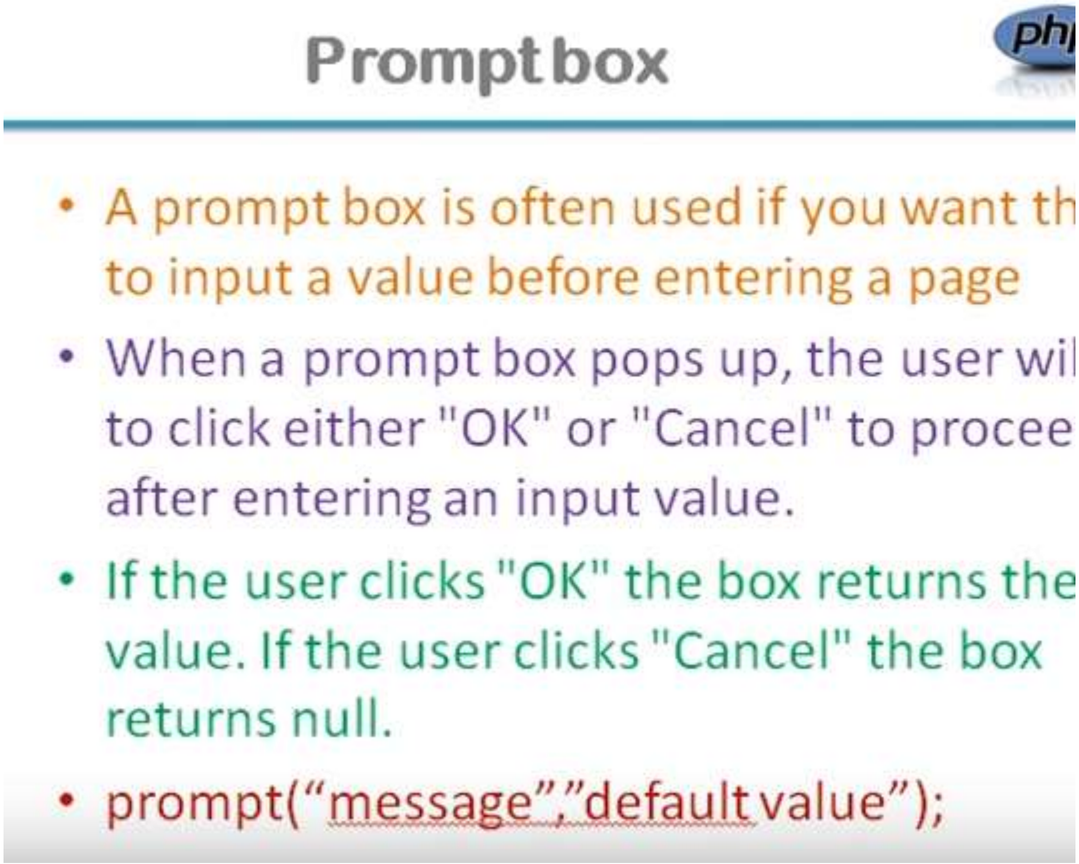
- When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.
- If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

- confirm(“some message”);



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

<body>
<script>
var x=confirm("Do you want to continue with Red color");
if(x){
document.getElementsByTagName("body")[0].style.backgroundColor="red";

}
</script>
<h1>Home page</h1>

</body>
</html>
```

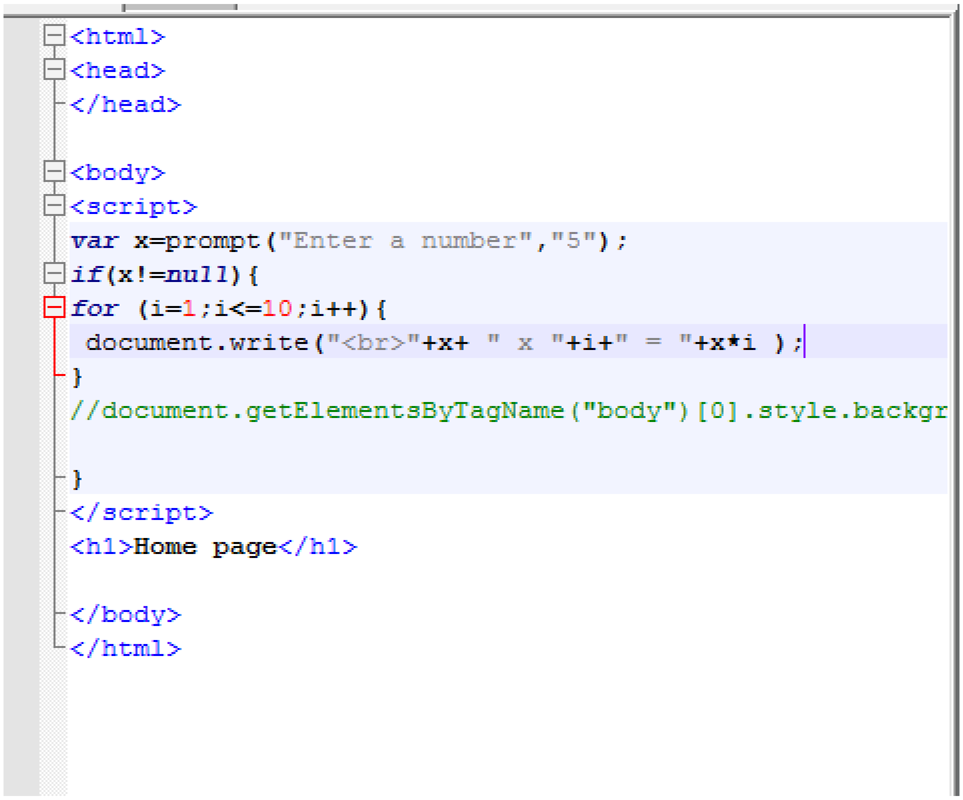
“Prompt Box”:



Promptbox

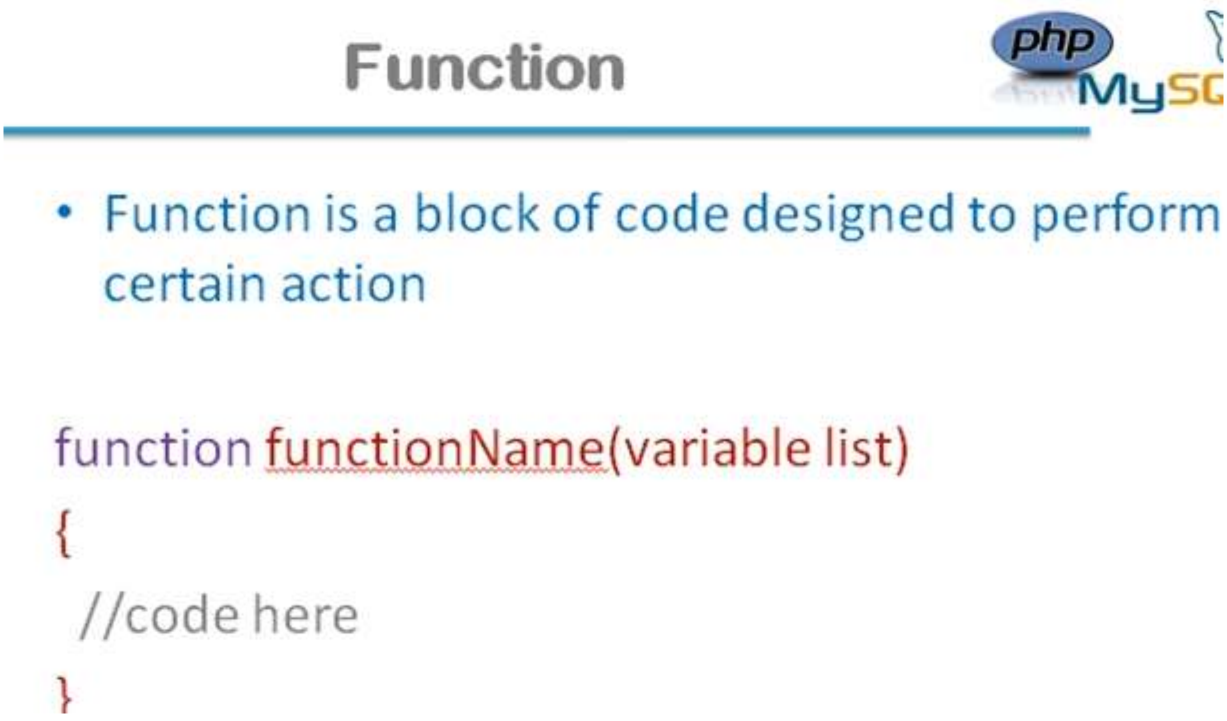
- A prompt box is often used if you want th
  to input a value before entering a page
- When a prompt box pops up, the user wil
  to click either "OK" or "Cancel" to procee
  after entering an input value.
- If the user clicks "OK" the box returns the
  value. If the user clicks "Cancel" the box
  returns null.
- prompt(“message”,”default value”);

Example no 1:



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

<body>
<script>
var x=prompt("Enter a number","5");
if(x!=null){
for (i=1;i<=10;i++){
 document.write("<br>"+x+ " x "+i+" = "+x*i );
}
//document.getElementsByTagName("body")[0].style.backgr

}
</script>
<h1>Home page</h1>

</body>
</html>
```

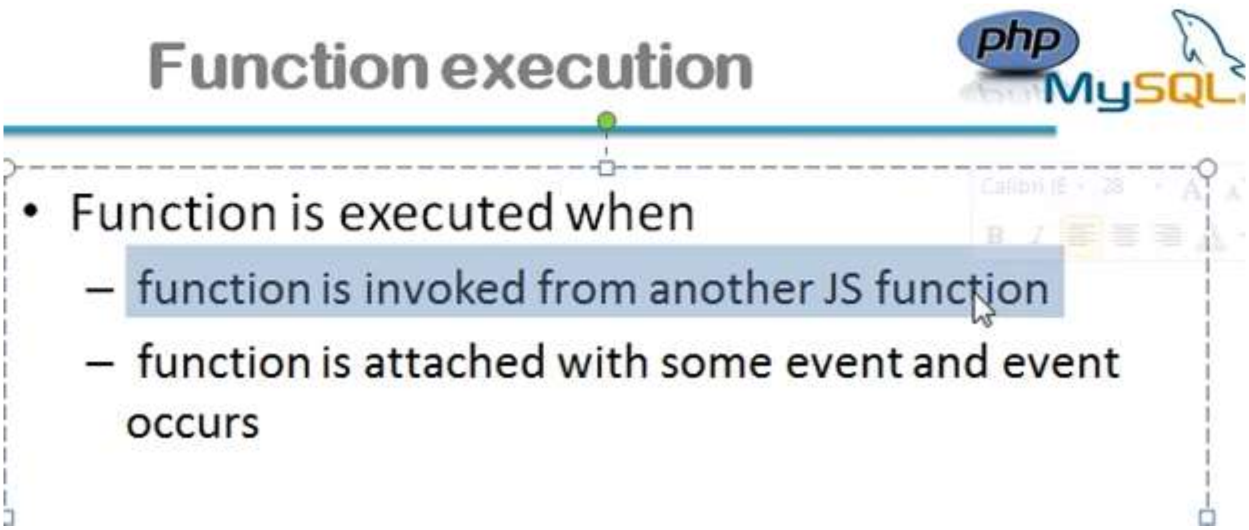
Function



Function

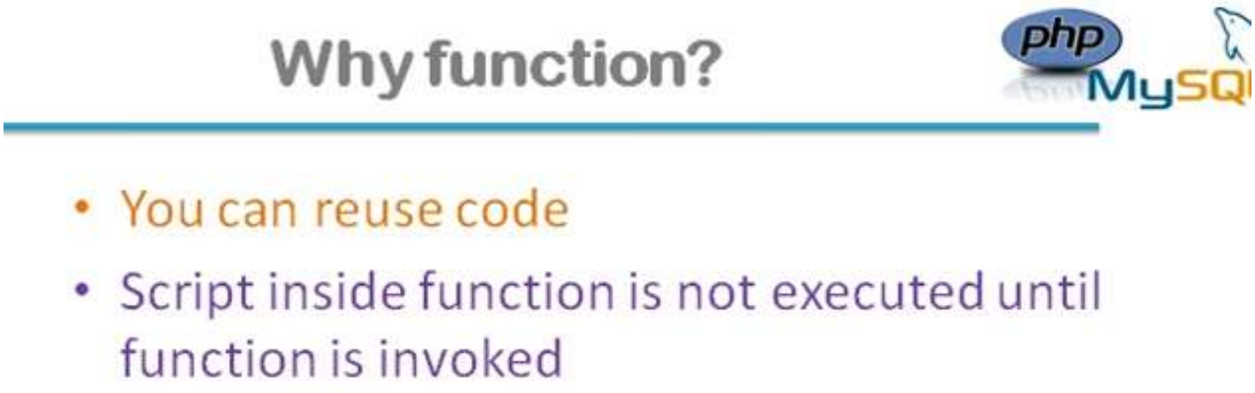
- Function is a block of code designed to perform certain action

```
function functionName(variable list)
{
 //code here
}
```



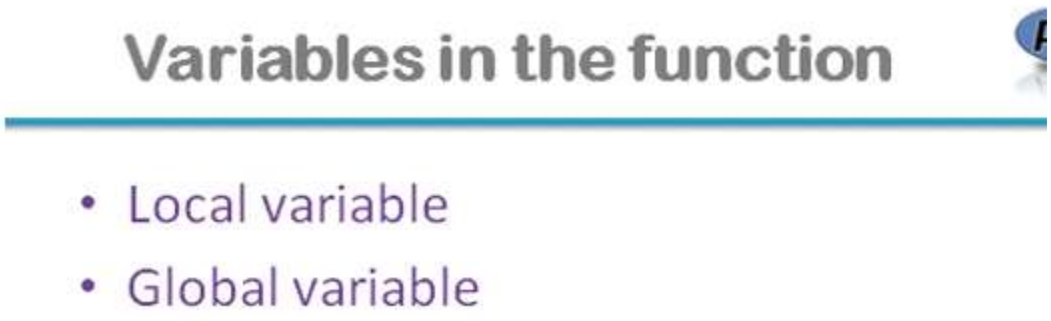
Function execution

- Function is executed when
  - function is invoked from another JS function
  - function is attached with some event and event occurs



Why function?

- You can reuse code
- Script inside function is not executed until function is invoked

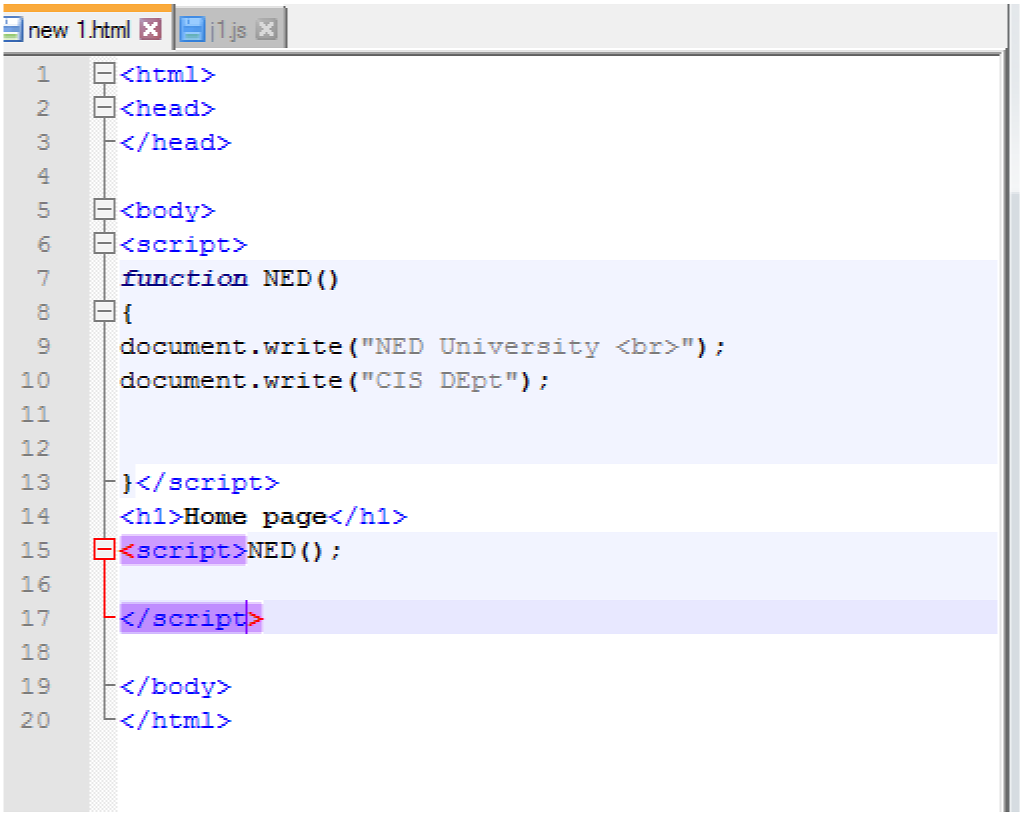


Variables in the function

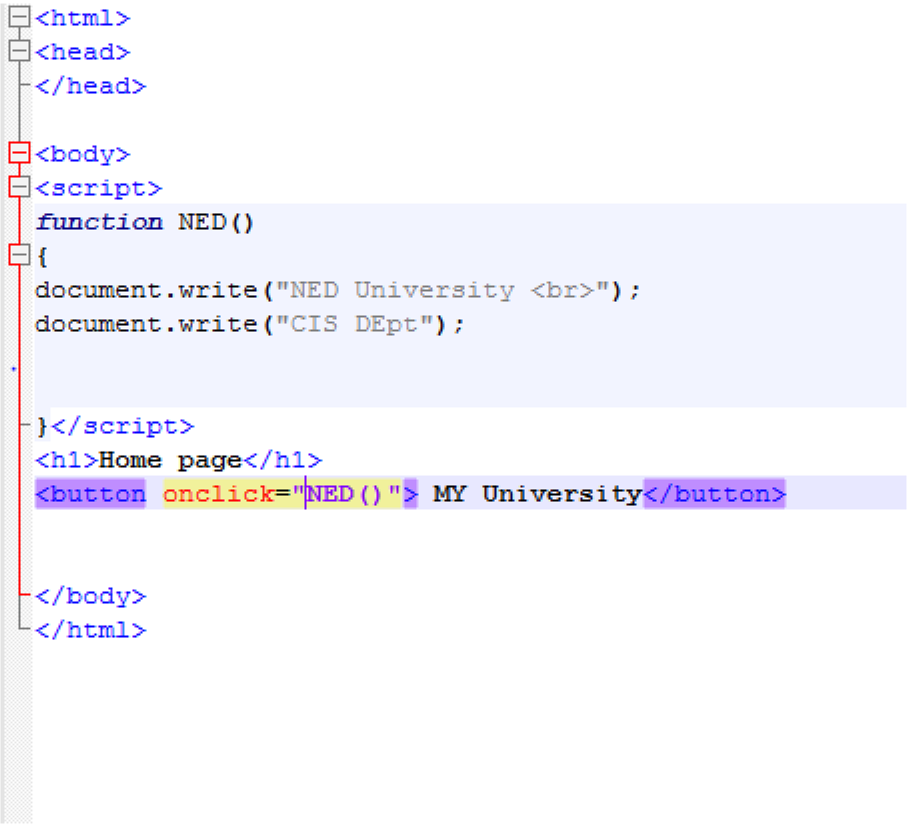
- Local variable
- Global variable

Function k andar agar mne var lkh kr koi variable declare kia h toh it’s a local variable

And if not so it will be global variable

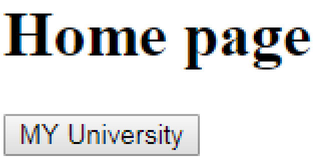


```
<html>
<head>
</head>

<body>
<script>
function NED()
{
document.write("NED University <br>");
document.write("CIS DEpt");


}</script>
<h1>Home page</h1>
<script>NED();

</script>

</body>
</html>
```



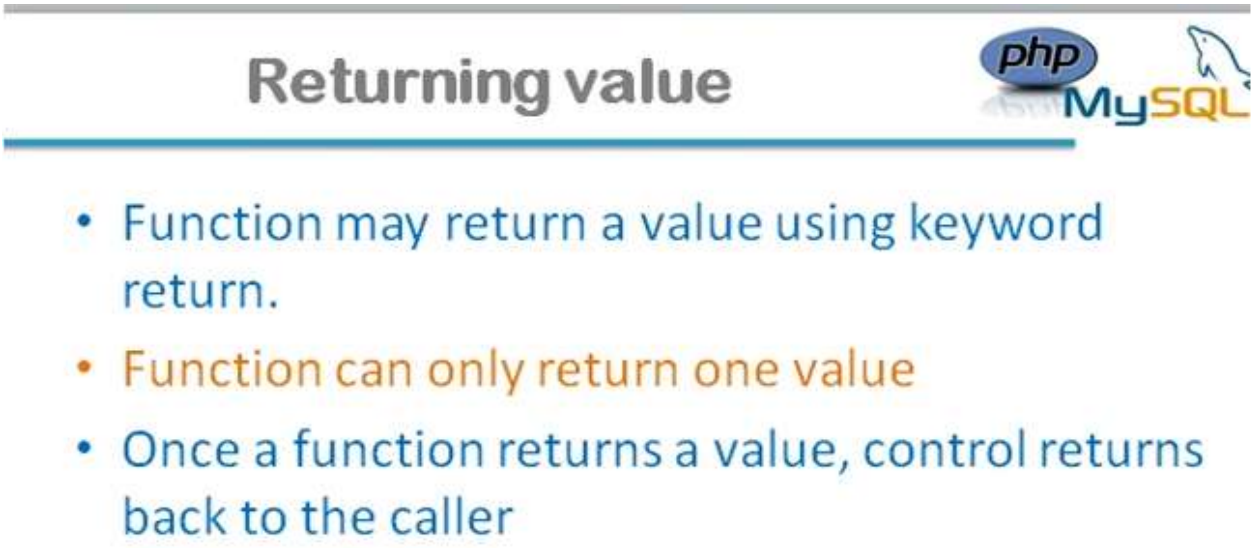
```
<html>
<head>
</head>

<body>
<script>
function NED()
{
document.write("NED University <br>");
document.write("CIS DEpt");


}</script>
<h1>Home page</h1>
<button onclick="NED()"> MY University</button>


</body>
</html>
```

Output



Home page

On click our function page will open as we want we get desired result



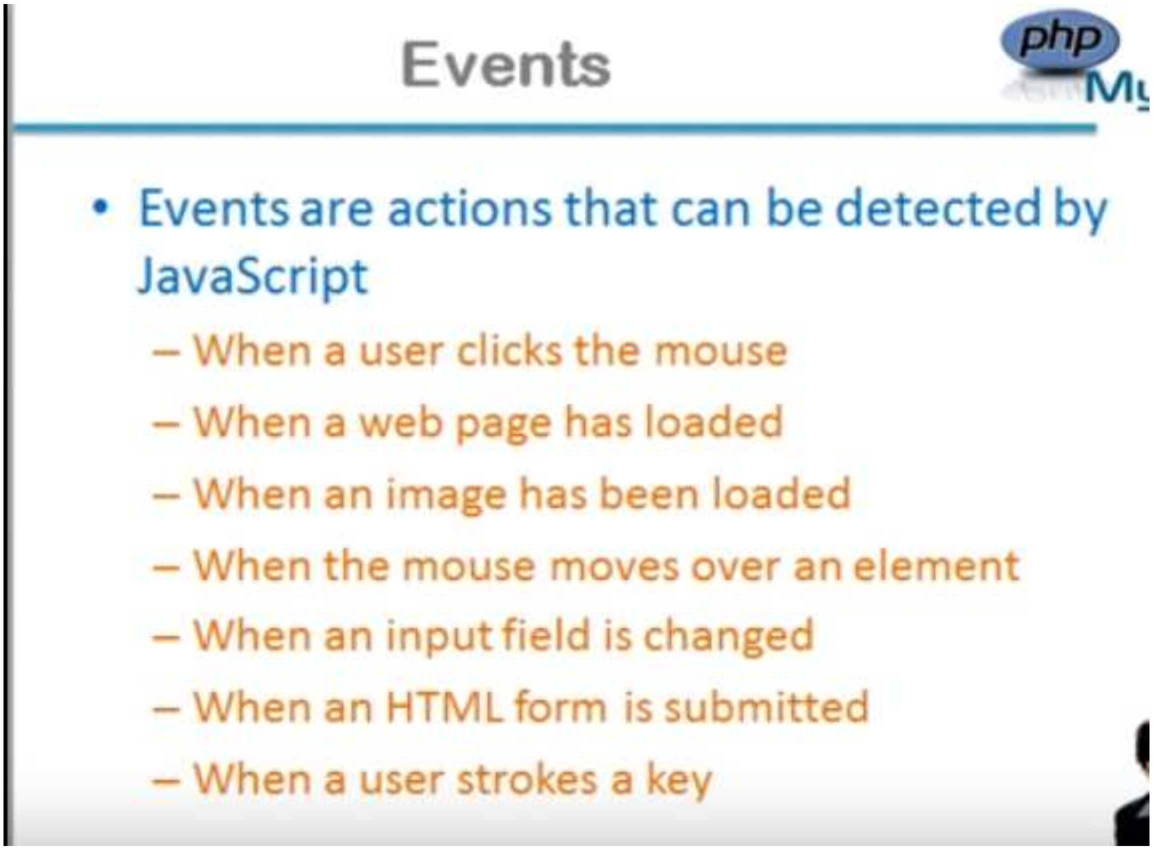
Returning value

- Function may return a value using keyword return.
- Function can only return one value
- Once a function returns a value, control returns back to the caller

Value jab return hoti h toh wahi jati h wahan se call hua h

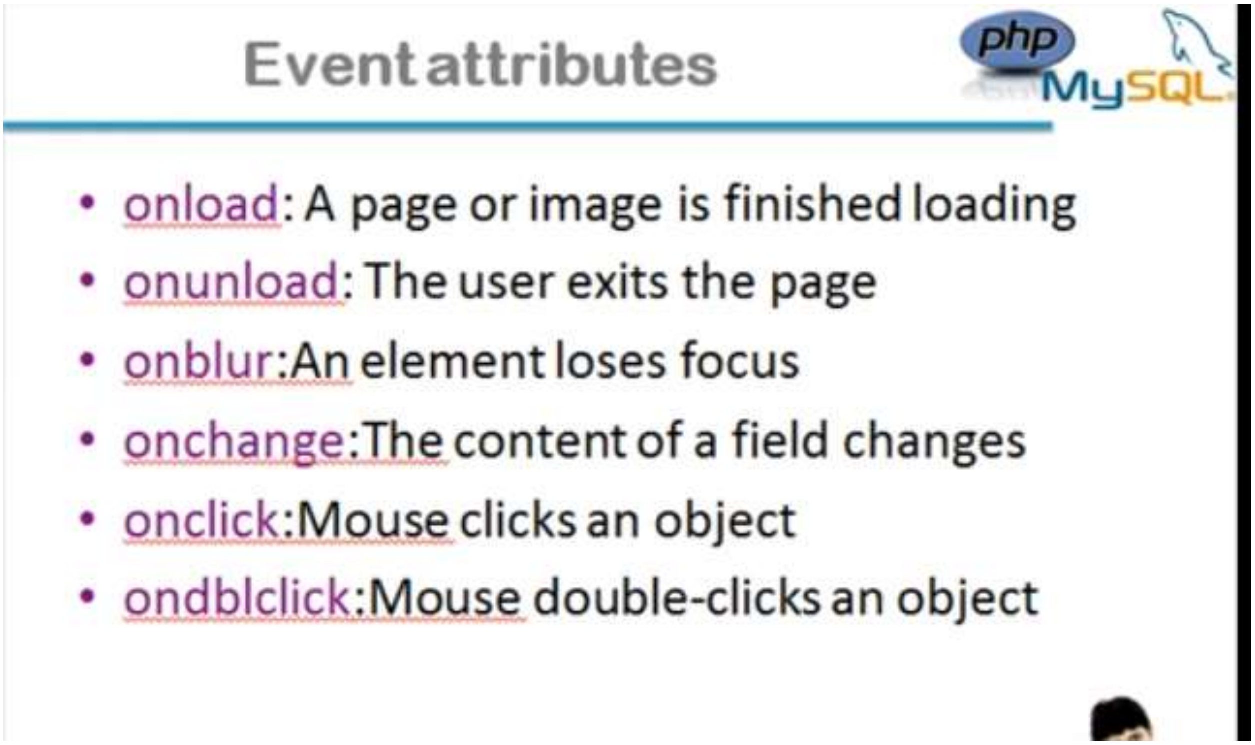
NED University
CIS DEpt

Event Handling in Javascript



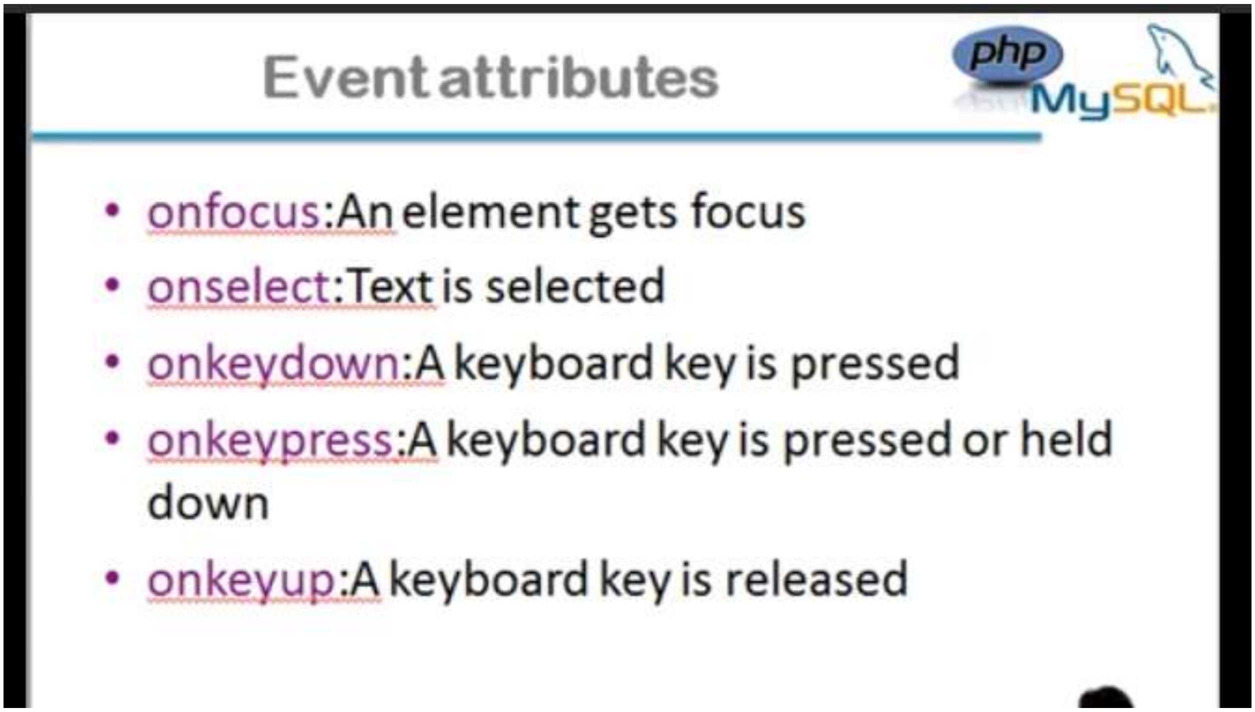
Events

- Events are actions that can be detected by JavaScript
  - When a user clicks the mouse
  - When a web page has loaded
  - When an image has been loaded
  - When the mouse moves over an element
  - When an input field is changed
  - When an HTML form is submitted
  - When a user strokes a key



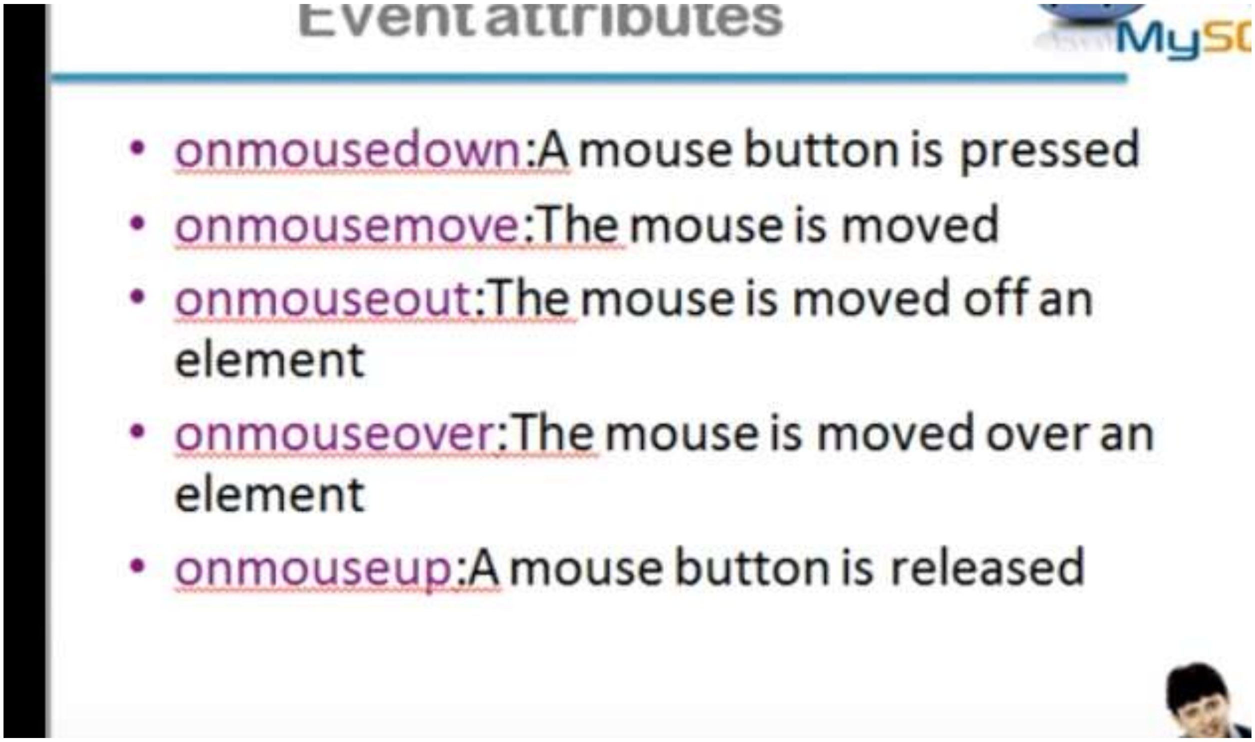
Event attributes

- onload: A page or image is finished loading
- onunload: The user exits the page
- onblur:An element loses focus
- onchange:The content of a field changes
- onclick:Mouse clicks an object
- ondblclick:Mouse double-clicks an object



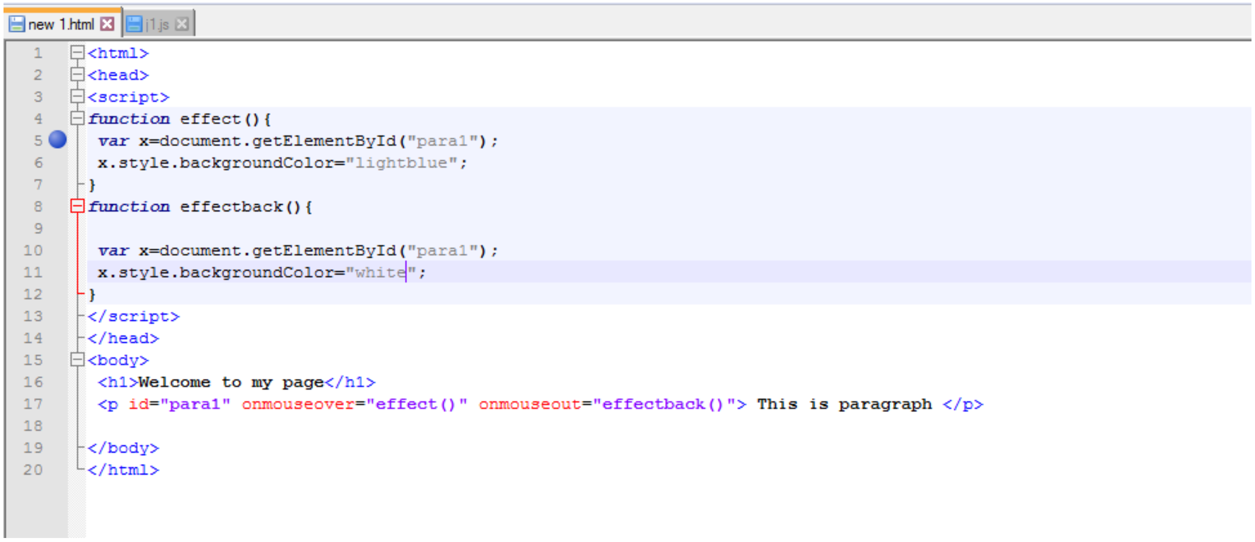
Event attributes

- onfocus:An element gets focus
- onselect:Text is selected
- onkeydown:A keyboard key is pressed
- onkeypress:A keyboard key is pressed or held down
- onkeyup:A keyboard key is released



Event attributes

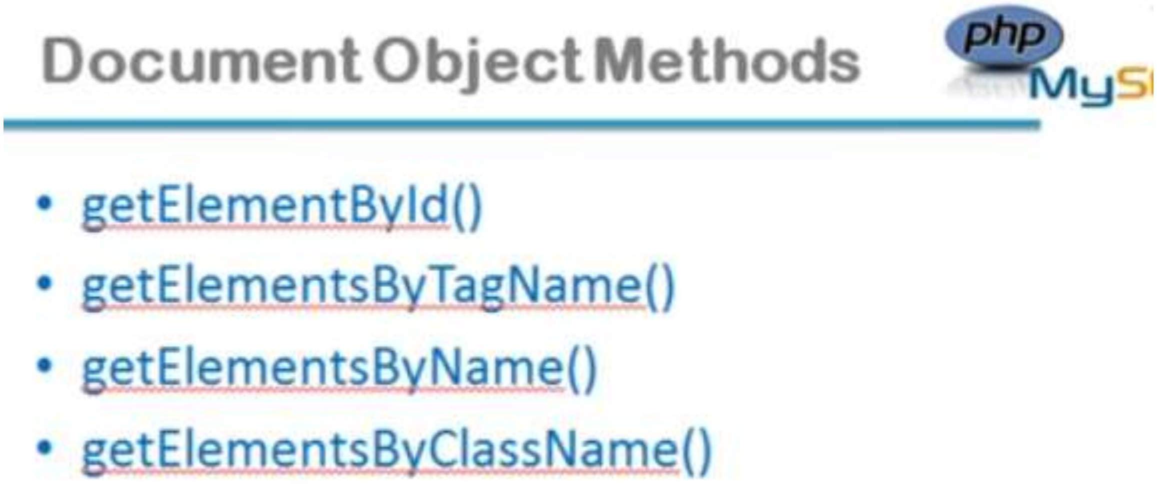
- onmousedown:A mouse button is pressed
- onmousemove:The mouse is moved
- onmouseout:The mouse is moved off an element
- onmouseover:The mouse is moved over an element
- onmouseup:A mouse button is released



```
<html>
<head>
<script>
function effect(){
 var x=document.getElementById("para1");
 x.style.backgroundColor="lightblue";
}
function effectback(){

 var x=document.getElementById("para1");
 x.style.backgroundColor="white";
}
</script>
</head>
<body>
 <h1>Welcome to my page</h1>
 <p id="para1" onmouseover="effect()" onmouseout="effectback()"> This is paragraph </p>

</body>
</html>
```

Onmouseover pe blue color hojayega or dobara krne pe white hojayega

How to access HTML elements in Javascript

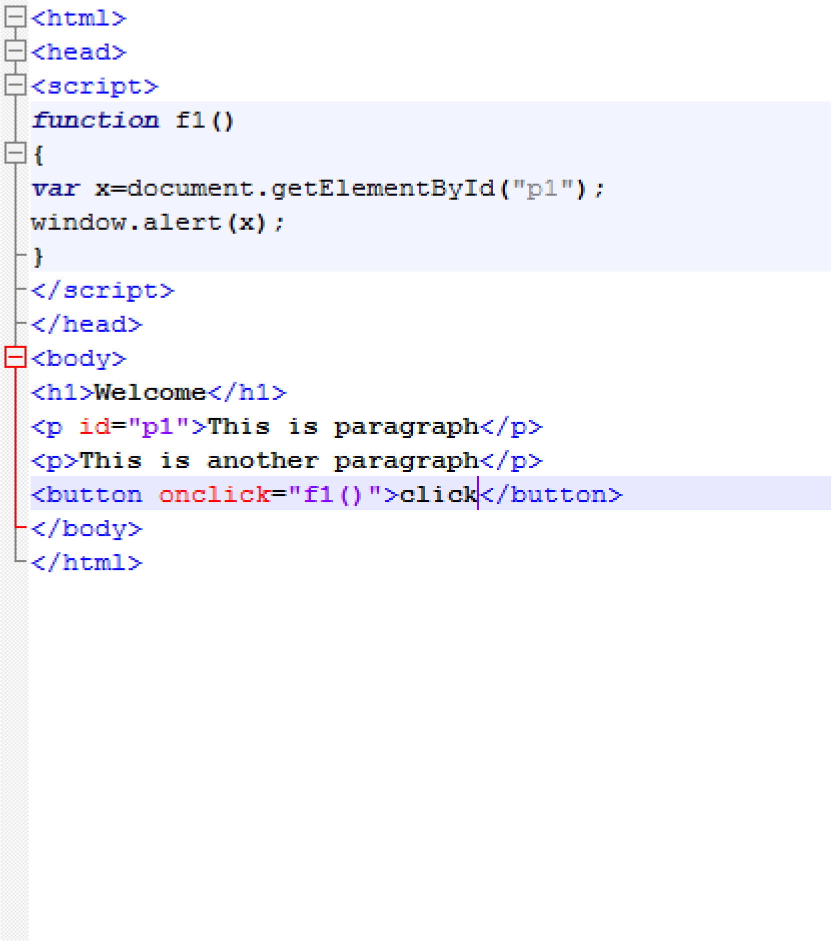


Document Object Methods

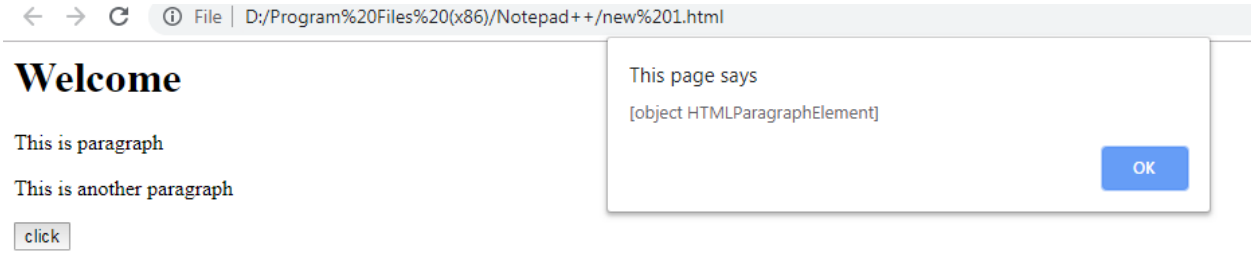
- getElementById()
- getElementsByTagName()
- getElementsByName()
- getElementsByClassName()

Yeh function document object k method h yeh hume object bana kr derhe hh smjho is sab m eke k parguement pass hota h or wo string hota h

Id k ealwa sa object ka array return kregy isee leye element lkha h elements nh



```
<html>
<head>
<script>
function f1()
{
var x=document.getElementById("p1");
window.alert(x);
}
</script>
</head>
<body>
<h1>Welcome</h1>
<p id="p1">This is paragraph</p>
<p>This is another paragraph</p>
<button onclick="f1()">click</button>
</body>
</html>
```



Welcome

This is paragraph

This is another paragraph

click

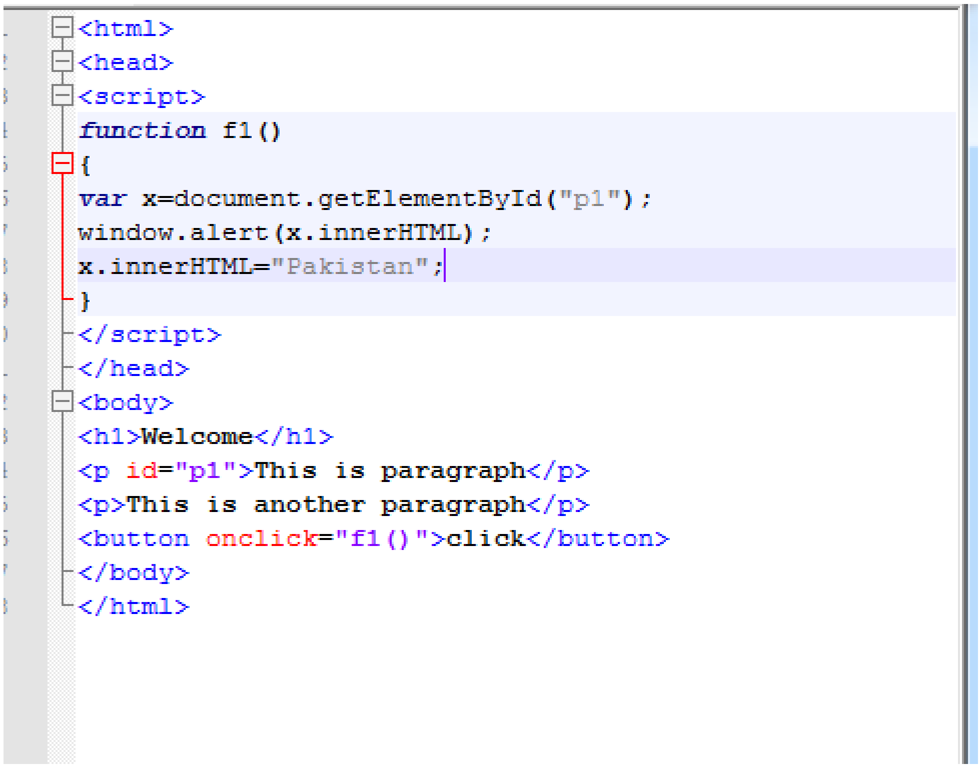
This page says

[object HTMLParagraphElement]

OK

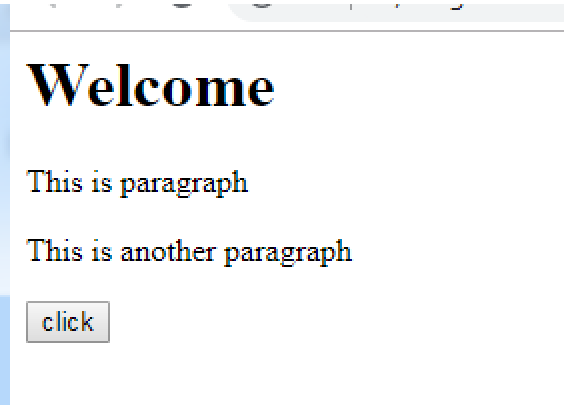
innerHTML property us k contain ko repsrent kregaa

or agar getName to p ko kregaa



```
<html>
<head>
<script>
function f1()
{
var x=document.getElementById("p1");
window.alert(x.innerHTML);
x.innerHTML="Pakistan";
}
</script>
</head>
<body>
<h1>Welcome</h1>
<p id="p1">This is paragraph</p>
<p>This is another paragraph</p>
<button onclick="f1()">click</button>
</body>
</html>
```

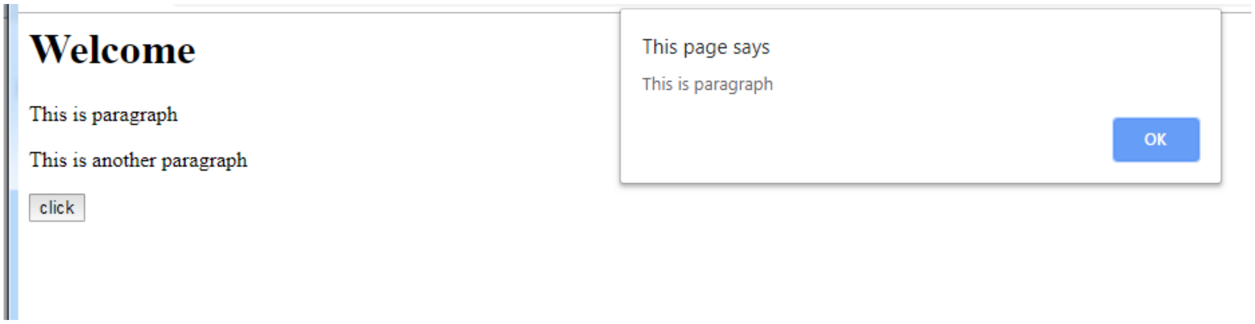
Before clicking ok



Welcome

This is paragraph

This is another paragraph



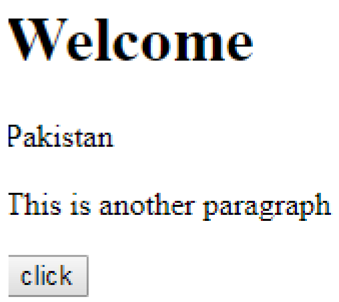
Welcome

This is paragraph

This is another paragraph

This page says

This is paragraph

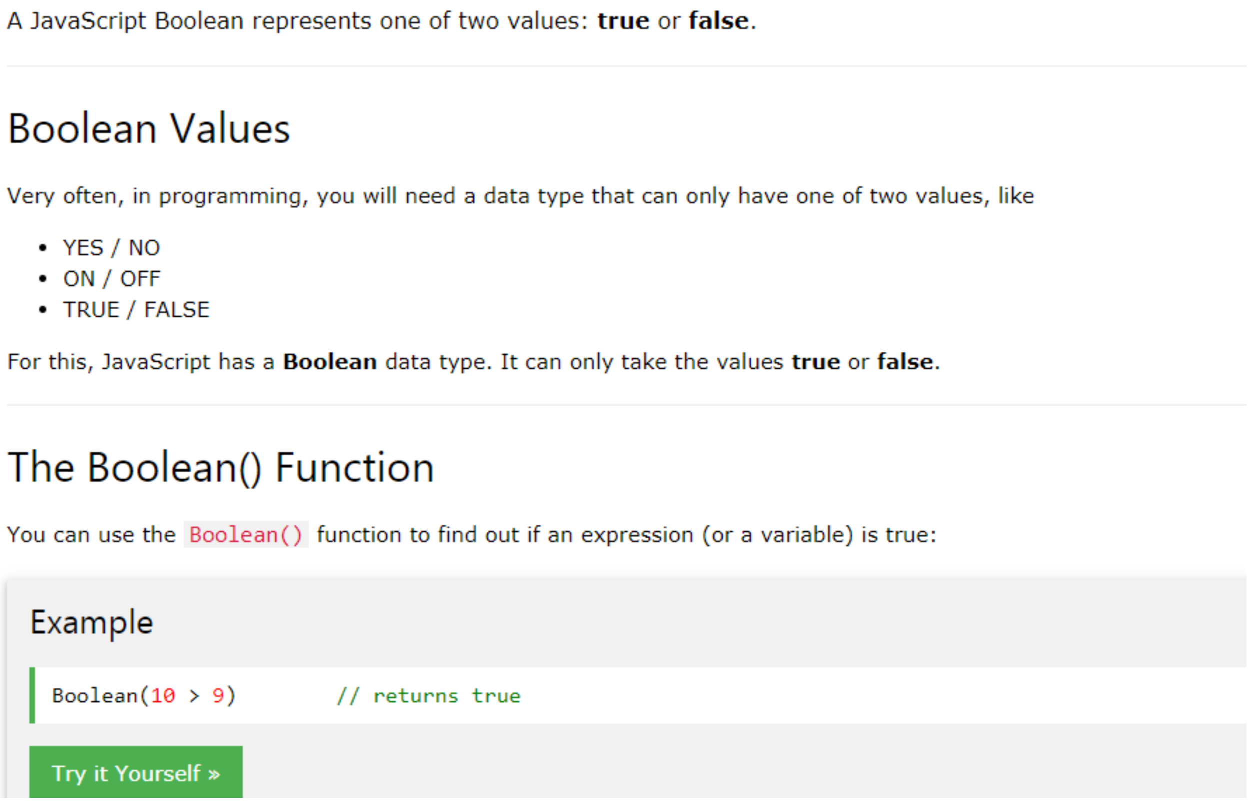


Welcome

Pakistan

This is another paragraph

DIY:



A JavaScript Boolean represents one of two values: **true** or **false**.

Boolean Values

Very often, in programming, you will need a data type that can only have one of two values, like

- YES / NO
- ON / OFF
- TRUE / FALSE

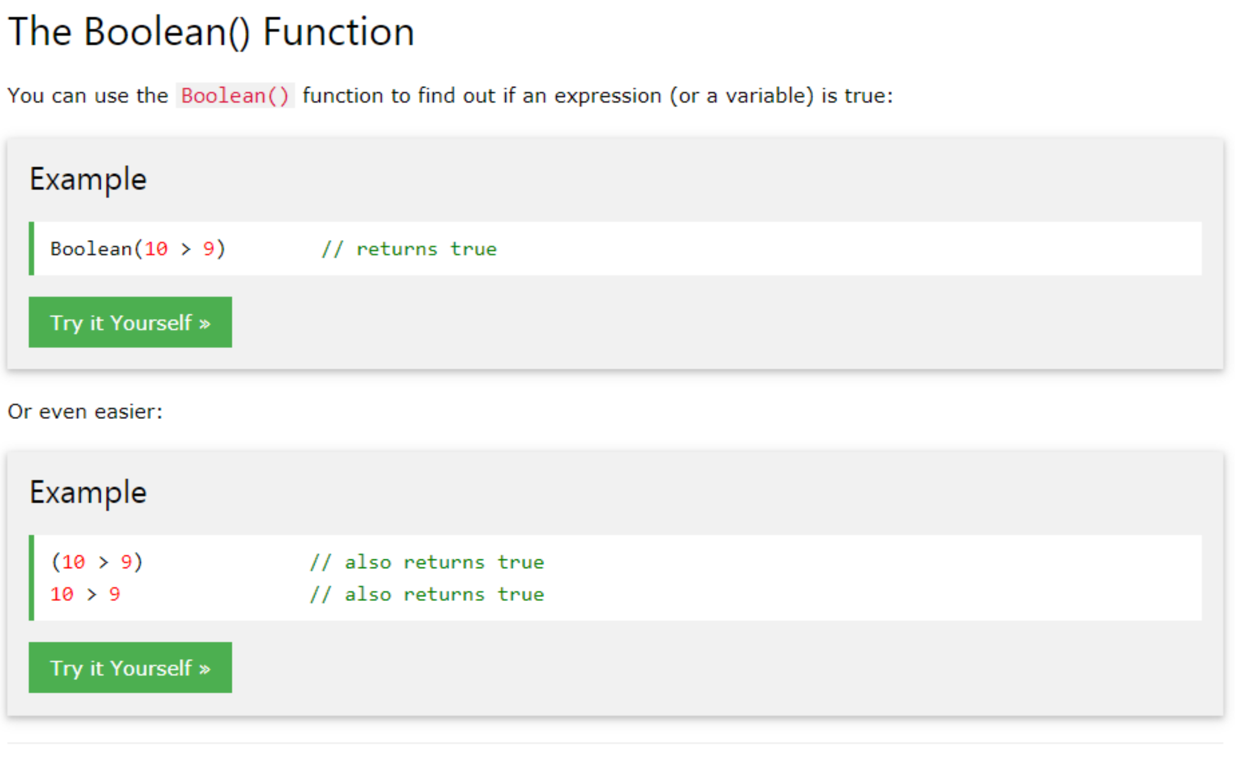
For this, JavaScript has a **Boolean** data type. It can only take the values **true** or **false**.

The Boolean() Function

You can use the `Boolean()` function to find out if an expression (or a variable) is true:

Example

```
Boolean(10 > 9)        // returns true
```



The Boolean() Function

You can use the `Boolean()` function to find out if an expression (or a variable) is true:

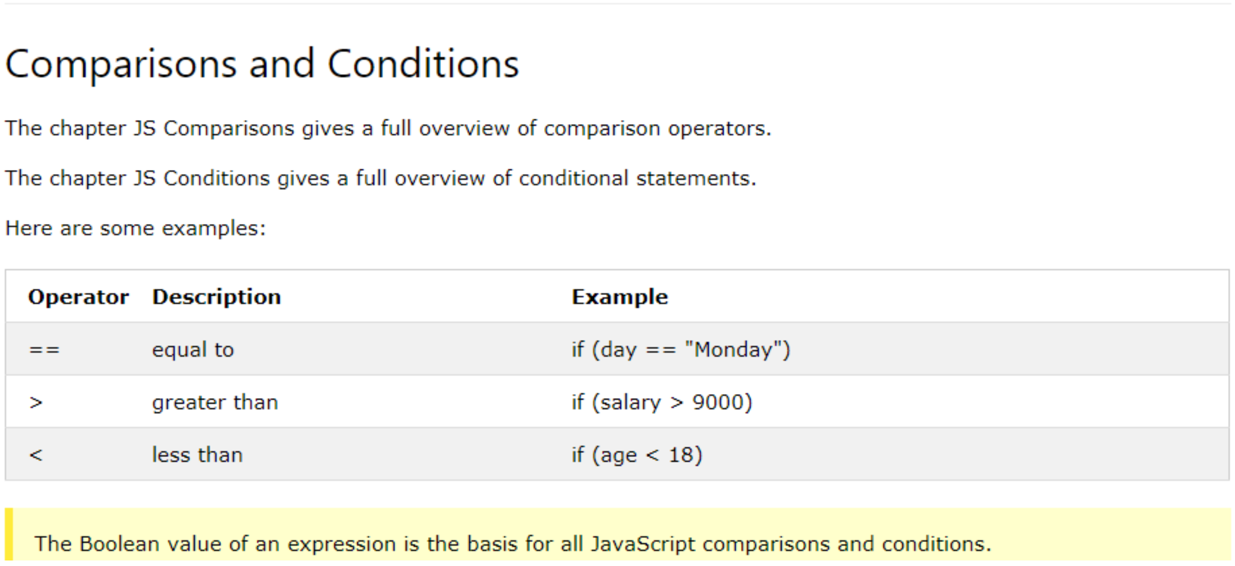
Example

```
Boolean(10 > 9)        // returns true
```

Or even easier:

Example

```
(10 > 9)              // also returns true
10 > 9                // also returns true
```



Comparisons and Conditions

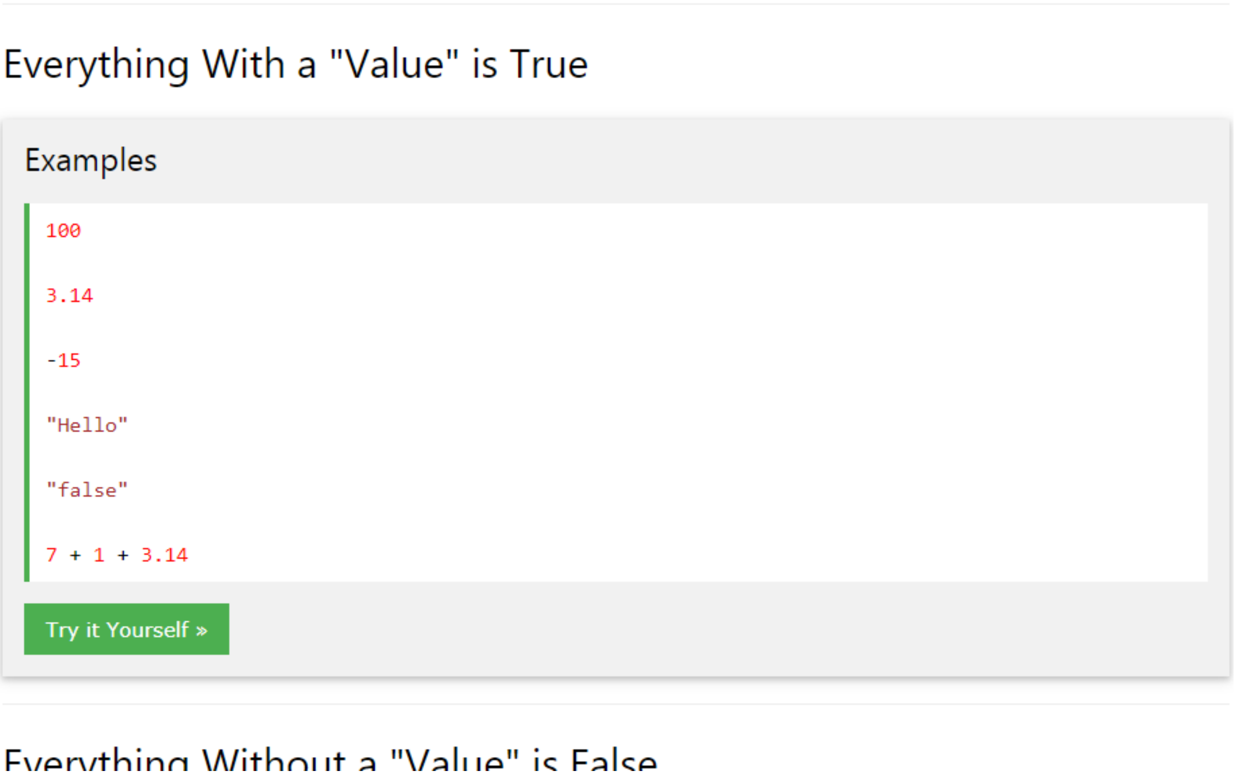
The chapter JS Comparisons gives a full overview of comparison operators.

The chapter JS Conditions gives a full overview of conditional statements.

Here are some examples:

Operator	Description	Example
==	equal to	if (day == "Monday")
>	greater than	if (salary > 9000)
<	less than	if (age < 18)

The Boolean value of an expression is the basis for all JavaScript comparisons and conditions.

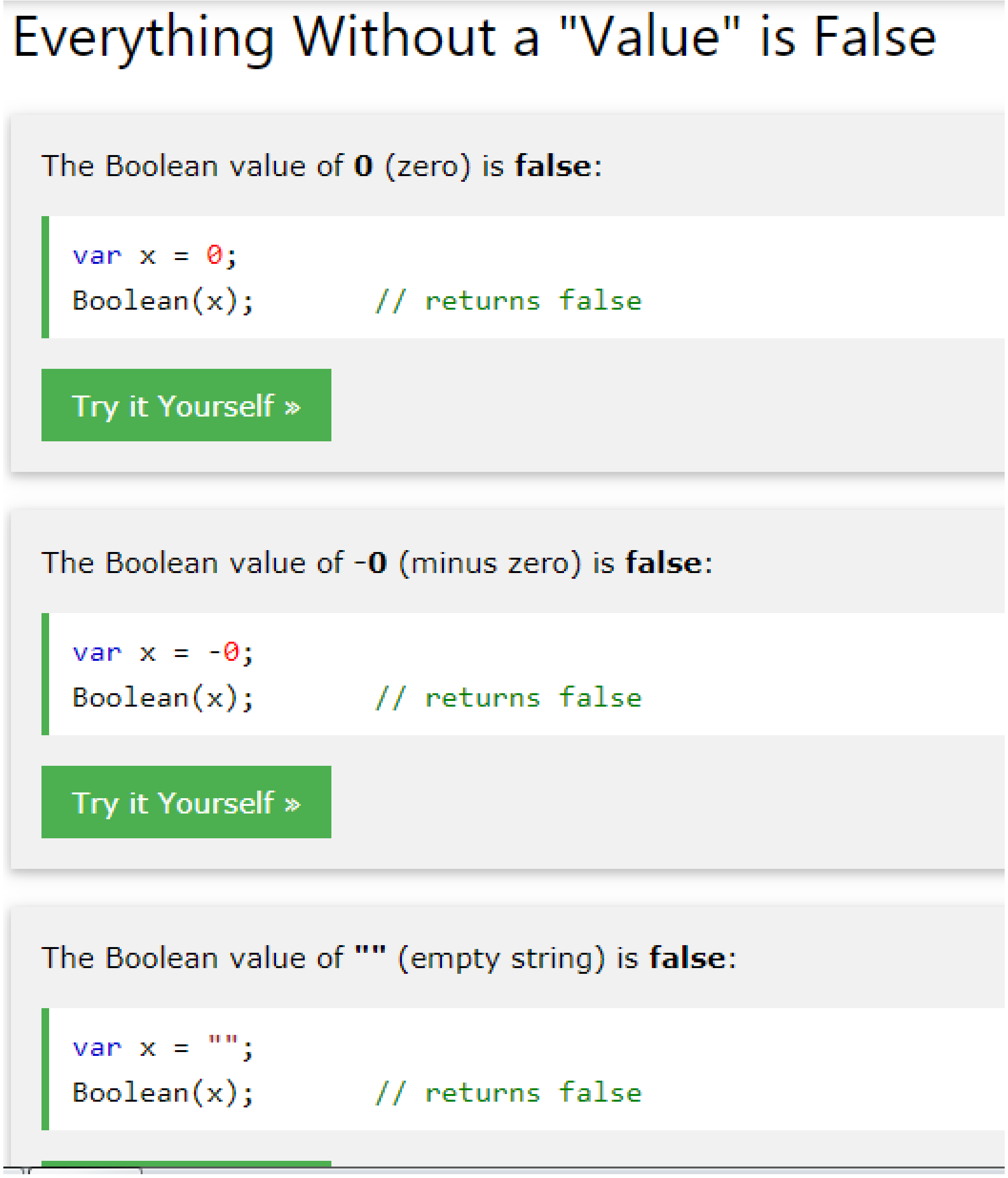


Everything With a "Value" is True

Examples

```
100

3.14

-15

"Hello"

"false"

7 + 1 + 3.14
```

Everything Without a "Value" is False



Everything Without a "Value" is False

The Boolean value of **0** (zero) is **false**:

```
var x = 0;
Boolean(x);       // returns false
```

The Boolean value of -**0** (minus zero) is **false**:

```
var x = -0;
Boolean(x);       // returns false
```

The Boolean value of **""** (empty string) is **false**:

```
var x = "";
Boolean(x);       // returns false
```

Switch

The JavaScript Switch Statement

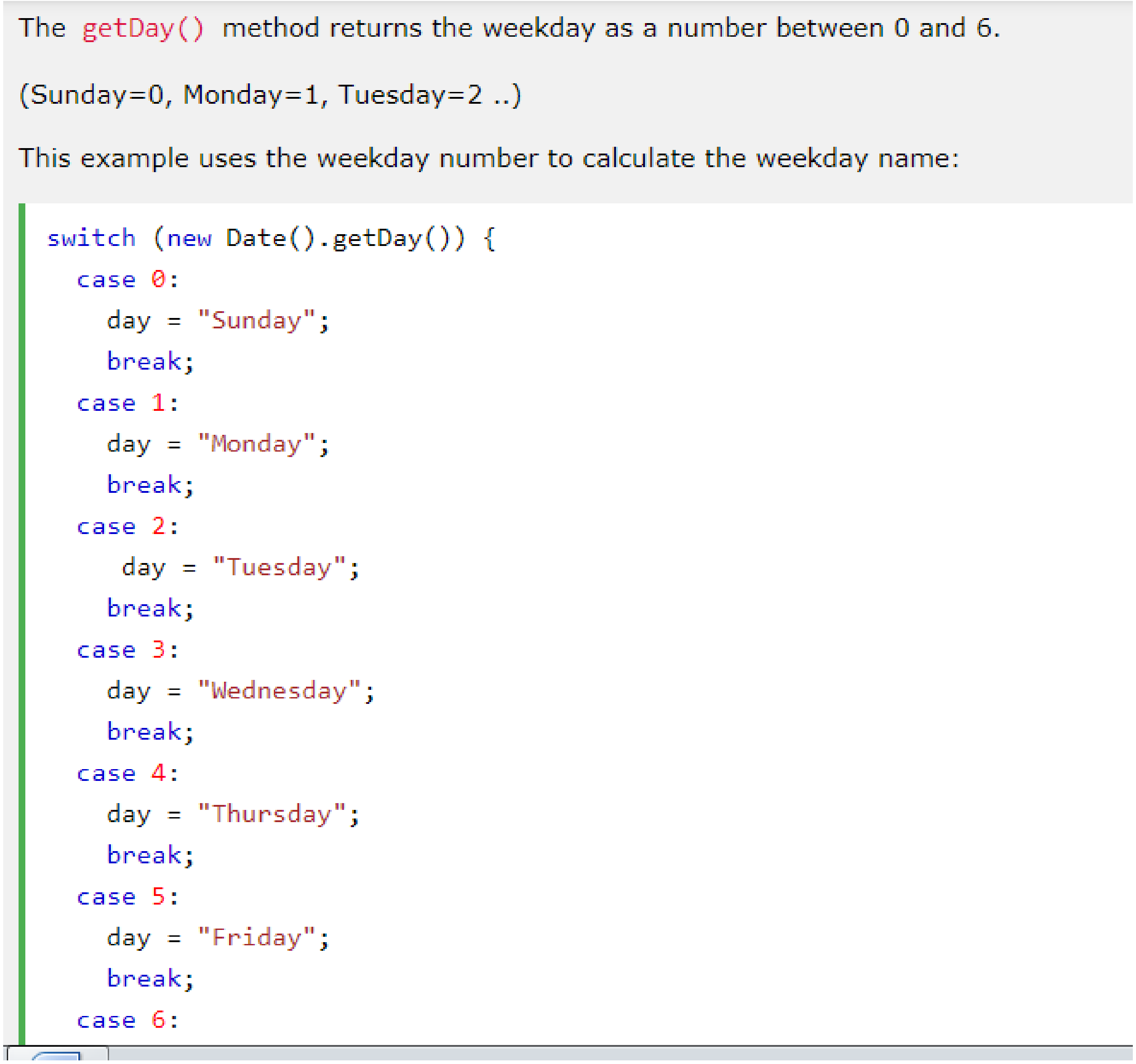
Use the `switch` statement to select one of many code blocks to be executed.

Syntax

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

This is how it works:

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.

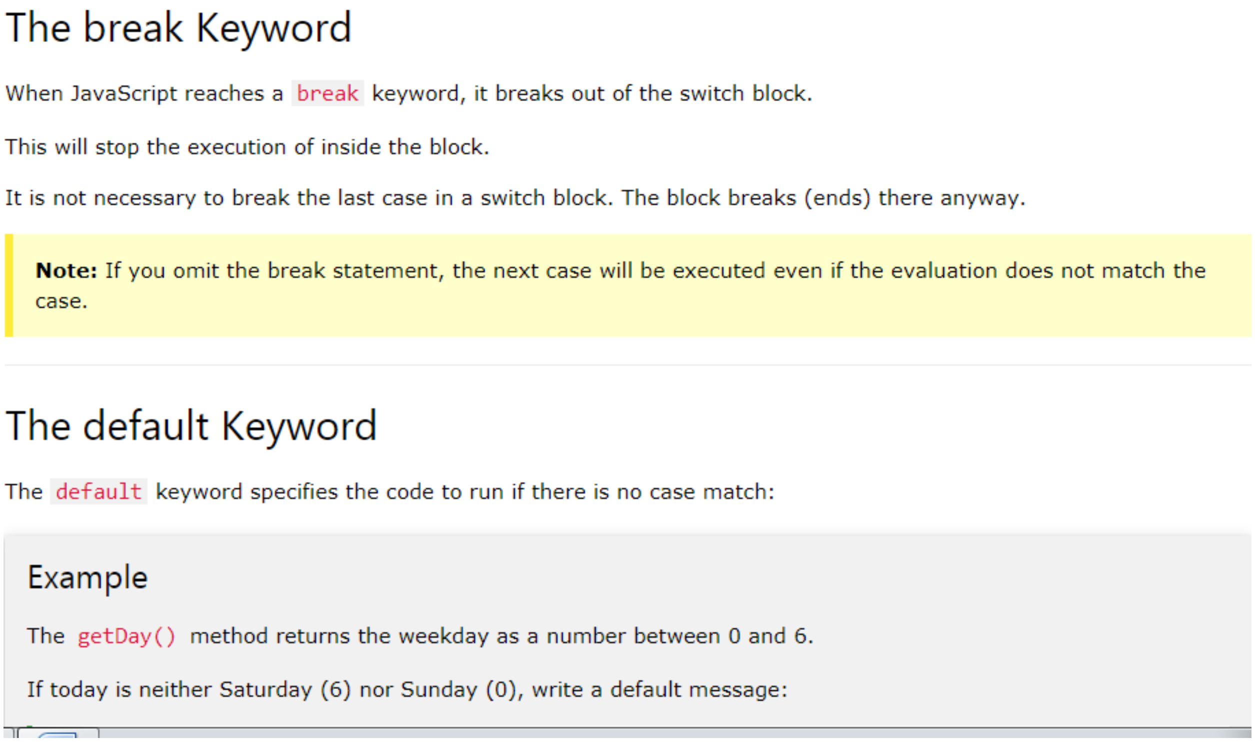


The getDay() method returns the weekday as a number between 0 and 6.

(Sunday=0, Monday=1, Tuesday=2 ..)

This example uses the weekday number to calculate the weekday name:

```
switch (new Date().getDay()) {
  case 0:
    day = "Sunday";
    break;
  case 1:
    day = "Monday";
    break;
  case 2:
     day = "Tuesday";
    break;
  case 3:
    day = "Wednesday";
    break;
  case 4:
    day = "Thursday";
    break;
  case 5:
    day = "Friday";
    break;
  case 6:
```



The break Keyword

When JavaScript reaches a `break` keyword, it breaks out of the switch block.

This will stop the execution of inside the block.

It is not necessary to break the last case in a switch block. The block breaks (ends) there anyway.

> **Note:** If you omit the break statement, the next case will be executed even if the evaluation does not match the case.

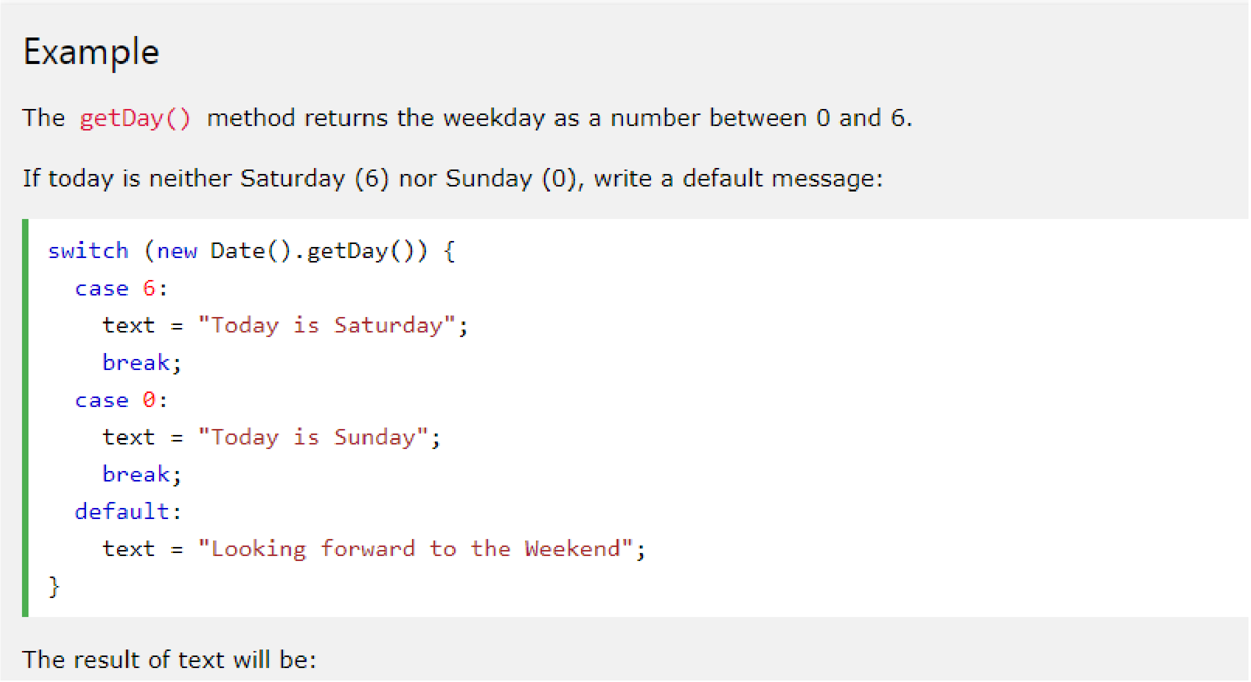
The default Keyword

The `default` keyword specifies the code to run if there is no case match:

Example

The `getDay()` method returns the weekday as a number between 0 and 6.

If today is neither Saturday (6) nor Sunday (0), write a default message:



Example

The `getDay()` method returns the weekday as a number between 0 and 6.

If today is neither Saturday (6) nor Sunday (0), write a default message:

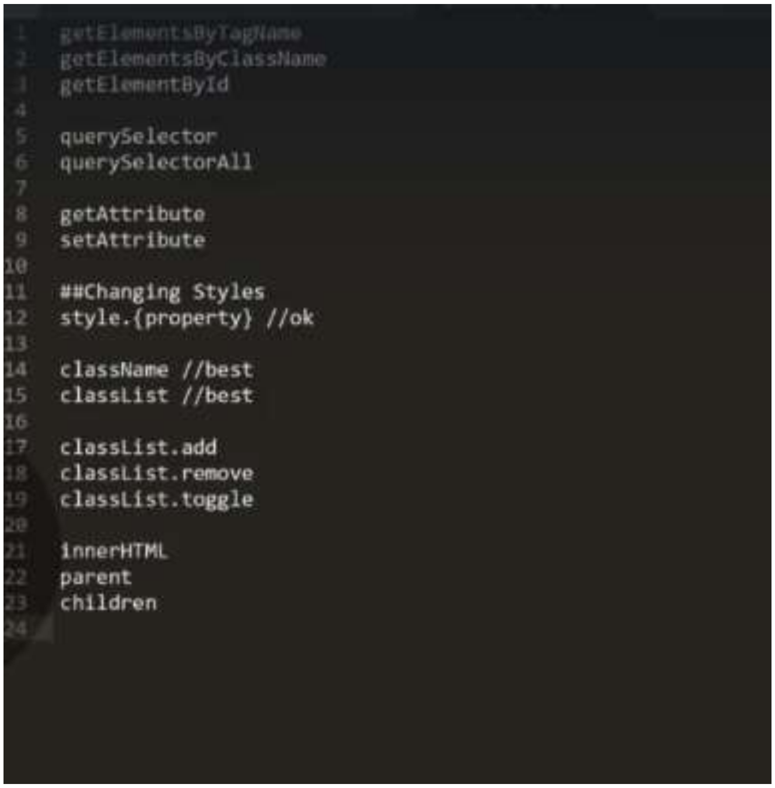
```
switch (new Date().getDay()) {
  case 6:
    text = "Today is Saturday";
    break;
  case 0:
    text = "Today is Sunday";
    break;
  default:
    text = "Looking forward to the Weekend";
}
```

The result of text will be:

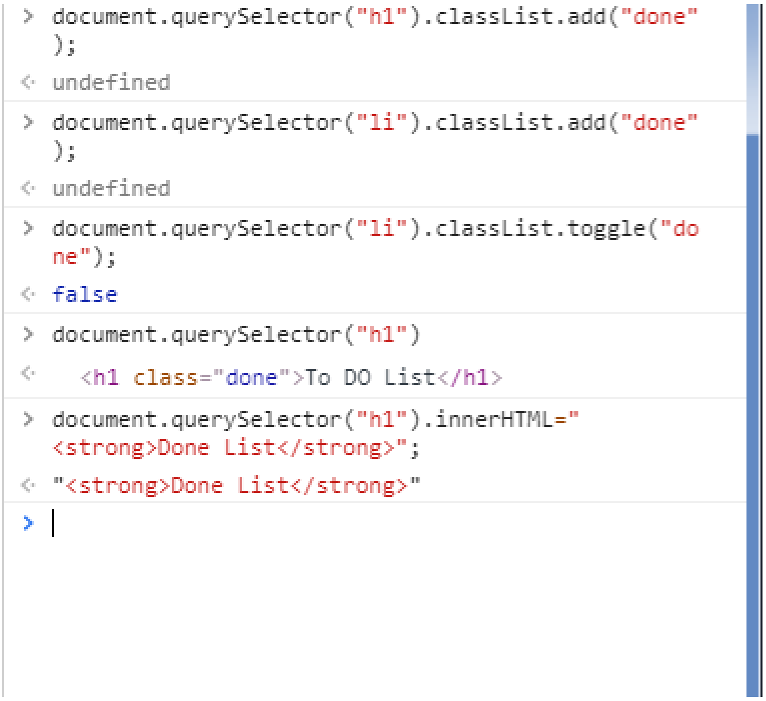
Social Media:

```
//var list=["item1","item2","item3","item4"];

//console.log(list[0]);


//var num=[1,2,3,4,5,6,7,8,9,0];

//var object={key:'Value1',key2:2,key3:false}


//.shift() is simply use for fire  niklne k lye list se element


//var field=["batting","bowling","feilding","wicket_leeping"];

//var player = {

 //  name:"Tendulkar",

  // age:28,

   //height:5.7,

   //married:true





//}


var database=[{

        username:"Mohsin1",

        password:"pass1"

},

{

        username:"Mohsin2",
        password:"pass2"
},
{
        username:"Mohsi3",
        password:"pass3"
}];


var newsfeed=[{
  username:'John',
  timeline:'Yeah ! i am so excited'
},
{
  username:'Tommy',
  timeline:'Yeah ! i am so tired'
}];

var usernamePrompt=prompt("Enter Your username");
var passwordPrompt=prompt("Enter Your Passowrd");


function validUser(user,pass){
                for(i=0;i<database.length;i++){
if(user==database[i].username && pass==database[i].password){
     return true;}//if
```

```
else
{   return false;
}//else
}//loop


}




function signIn(user,pass){
        if (validUser(user,pass)){
                console.log(newsfeed);
        }
        else
        {
                alert("Incorect Id or Passowrd");
        }

}//func

signIn(usernamePrompt,passwordPrompt)
```



```
getElementsByTagName
getElementsByClassName
getElementById

querySelector
querySelectorAll

getAttribute
setAttribute

##Changing Styles
style.{property} //ok

className //best
classList //best

classList.add
classList.remove
classList.toggle

innerHTML
parent
children

```



```
> document.querySelector("h1").classList.add("done"
  );
<· undefined
> document.querySelector("li").classList.add("done"
  );
<· undefined
> document.querySelector("li").classList.toggle("do
  ne");
<· false
> document.querySelector("h1")
<·   <h1 class="done">To DO List</h1>
> document.querySelector("h1").innerHTML="
  <strong>Done List</strong>";
<· "<strong>Done List</strong>"
> |
```

DOM Common Selectors

Following are the common DOM Selectors

getElementsByTagName

getElementsByClassName

getElementById

querySelector

querySelectorAll

getAttribute

setAttribute

##Changing Styles

style.{property} //ok

className //best

classList //best

classList.add

classList.remove

classList.toggle

Bonus Command

innerHTML

parentElement

children

##It is important to CACHE selectors in variables

Page DO TO LIST

Javascript file:

```
var button= document.getElementById("enter");

var input= document.getElementById("userinput");

var ul=document.querySelector("ul");


function inputLength(){

        return input.value.length;

}
```

```
function ListElement(){
   var li = document.createElement("li");
         li.appendChild(document.createTextNode(input.value));
         ul.appendChild(li);
   input.value=" ";
}



button.addEventListener("click",function(){
         if(inputLength() > 0){
         console.log(input.value);
}//if
         }//finction
)




input.addEventListener("keypress",function(event){
         if(inputLength() >0 && event.keyCode==13){
         console.log(input.value);
         ListElement()
}//if


})
```

```
button.addEventListener("click",function(){
  console.log("User Clicked");
})
```

HTML FILE:

```
<!DOCTYPE html>
<html>
<head>
        <title>javaFile with DOM</title>
        <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
        <h1>To DO List</h1>
        <p id="first">Jobs to Do Today</p>
        <input id="userinput" type="text" placeholder="Enter Items Here">
        <p class="second">This is Class Text</p>
        <button id="enter">Click</button>
        <ul>
                <li random="23">Wake Up</li>
                <li>brush teech</li>
                <li>go to work</li>
                <li>play footbal</li>
        </ul>
        <script type="text/javascript" src="java.js"> </script>
</body>
</html>
```

CSS FILE

```
.done{
        text-decoration: line-through;
}
```

```
function isUserValid(bool){
return bool;
}

var login=isUserValid(false)?"Login Succes" | "Login Denied";

function motion(direction){
 var move;

 switch(direction){

   case "up":var move="You are flying";

             break;

    case "down":var move="You are going down";

             break;

      case "right":var move="You are moving right ";

             break;

         case "left":var move="You are moving left";

             break;



 }//switch


}
```

Code With harry

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


 console.log({harry:"this"})

 console.table({harry:"Table"})

 console.warn("This is warning")
 console.timeEnd('Your code end')

var age=120

 console.assert(age>189,"Not possible")

 console.error("This is an error")

/*Mutiline error*/

// var,let,const is type k varibel bantey h

var age=10

var age1

if(age==10){

        console.log("Hello");


}

</script>
```

```
var name="Harry"

var channel;

var marks=3444

console.log(name,channel,marks)
```

Const and let:

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

                const ownersName="Hari Ram";
```

```
        console.log(ownersName) //const aise varible jo chnage nh hosktey

        /*let simply use istead of va ryeh apne scope m hi use hota h iska apna scope hota h har varible ka let=Block leevl scope */{

        let city="RamPur"

   console.log(city);

    }



    const arr1=[1,2,3,4,5,6];

    arr1.push(45)

    console.log(arr1);

    const object={

        1:{11:"oneone"},

        2:{22:"twotwo"}

    }

    console.log(object)



</script>
```

Primivitive database m memoryallocation stack m hota h (Base)

reference data m heap m store hota h(Object classes)

Permitive datatype:

1-string

2-number

3-boolean

4-undefined

5-sysmbol

Reference DataType:

1-Array

2-Object literal

3-Functions

4-Dates

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

        /**/

        let name="Harry"

        console.log("My string is "+typeof(name))


        myarr=[1,2,4,5,,7,"string",true]

        console.log("Data type is "+typeof(myarr))

        //

        let marks={

                'harry':89,

                'shutam':34,

                'rohamDas':33

        }


        function findname(){



        }
        console.log(marks)




</script>
```

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

        /**/

        let a=window;

        console.log(a)

        //allert is window object but window golabl h islye use mention nh krna parta

        alert("Hello Man")

        a=prompt("Ths is ")

        a=confirm("Are You sure")

        //these three alert prompt and comfirm are object of window yeh sare chzein ab website m use nh hota h

        a=window.innerHeight;

        console.log(a)

console.log(scrollY)

   console.log(scrollX)

   console.log(location) //give locations mean ip address wagera yeh sare chzein ab itne use nh hoti par maloom hona chahye
```

<u>DOM:</u>

```
    a=document.all;

    console.log(a);
    Array.from(a).forEach(function(element){
console.log(element)
//array form karo a ka or phir function alagaya us ko dia lement ko or element ko print krdia


    })
```

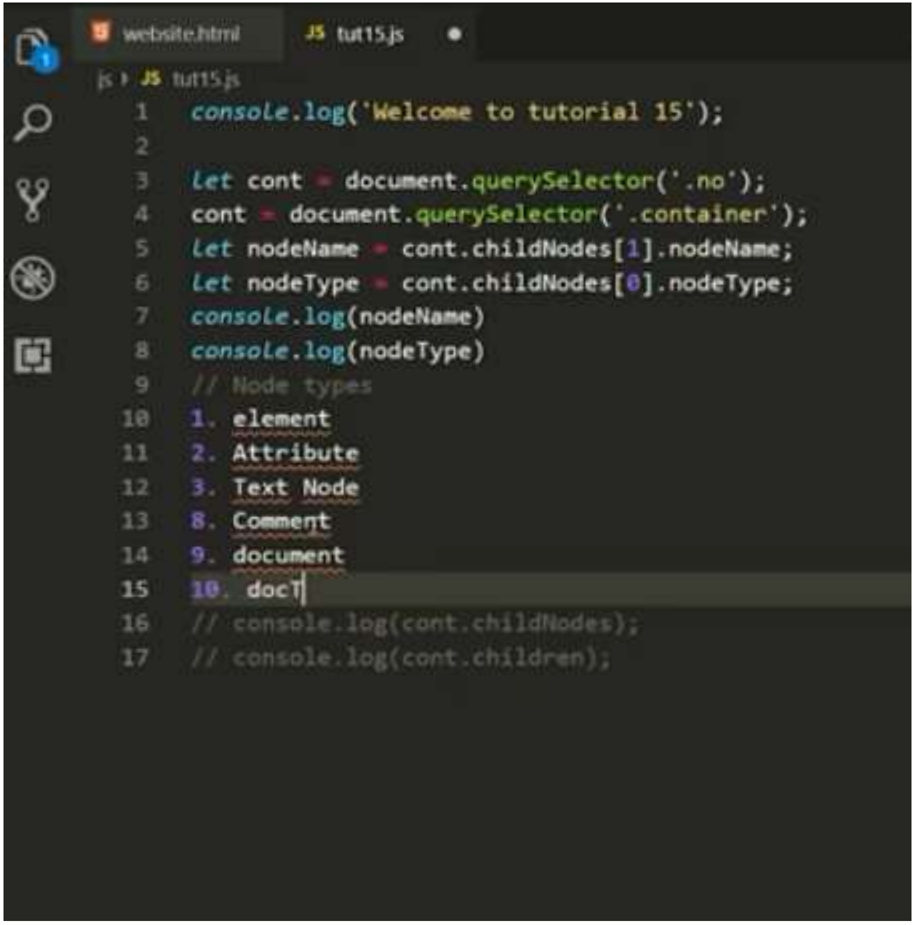
```
    b=document.links[0] //phela link dega
    console.log(b)
```

```
</script>
```

```
<!DOCTYPE html>
<html>
<head>
        <title>Practice</title>
</head>
<body>
<p>
        <div class="container">
                <h1 id="heading">Welcome to club</h1>
                <div class="child red" id="first">Child 1</div>
                <div class="child">Child 2</div>
                <div class="child red">Child 3</div>
                <div class="child">Child 4</div>
<a href="www.facebook.com">Here</a>
        </div>
        <form action="none.html" method="post">
                <input type="text" name="Hello" placeholder="hello">
                <img src="2.jpg">


        </form>
        <div class="container" >

<span id="myText"></span>
        </div>
        <script type="text/javascript">
                //to display all chzin in html page write

a=document.images
Array.from(a).forEach(function(element){
  console.log(element);

})


</script>
</p>
```

```
</body>

</html>-
```

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

  /*

  1-Single Element Selector

  2-MultiSelector


  */

  let element=document.getElementById('first')

  //element=element.childNodes;

  //console.log(element)

  //element=element.parentNode;

    // console.log(element)

     element.style.color ='red';

     console.log(element)

     element.innerText="Harry is a good boy"

     element.innerHTML='<br>Hello World </br>'

      let sel=document.querySelector('#first') // id jis ki firs tho wo select hojayeselector can be id

      console.log(sel)

      let sel1=document.querySelector('.child') //child class ka phlea element jo bhi h

      console.log(sel1)


       let sel2=document.querySelector('div') //phela hi div milega isse
     sel2.style.color='blue';


let elemnt=document.getElementsByClassName('child');
console.log(elemnt)
let elem=document.getElementsByTagName('div');
Array.from(elem).forEach(element => {
        console.log(element)
        element.style='bold'
})
console.log(elem)

</script>
```



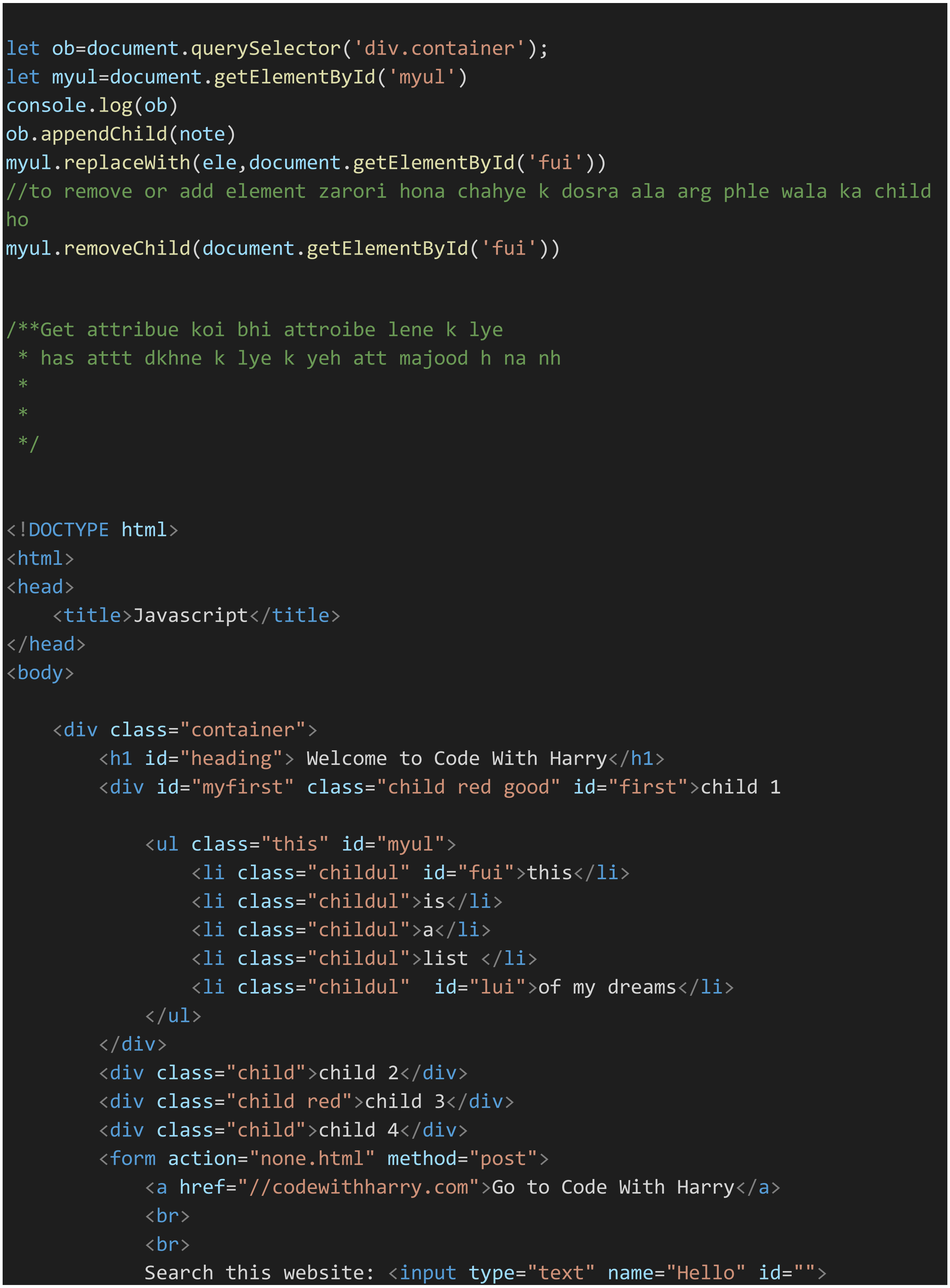
```
console.log('Welcome to tutorial 15');

let cont = document.querySelector('.no');
cont = document.querySelector('.container');
let nodeName = cont.childNodes[1].nodeName;
let nodeType = cont.childNodes[0].nodeType;
console.log(nodeName)
console.log(nodeType)
// Node types
1. element
2. Attribute
3. Text Node
8. Comment
9. document
10. docT
// console.log(cont.childNodes);
// console.log(cont.children);
```

```
let element=document.getElementById("first")
console.log(element)
console.log(element.parentNode)

element.parentNode.style.color='red'
console.log(element)
element.innerText="Harry harry"
element.innerHTML="Hello"
```

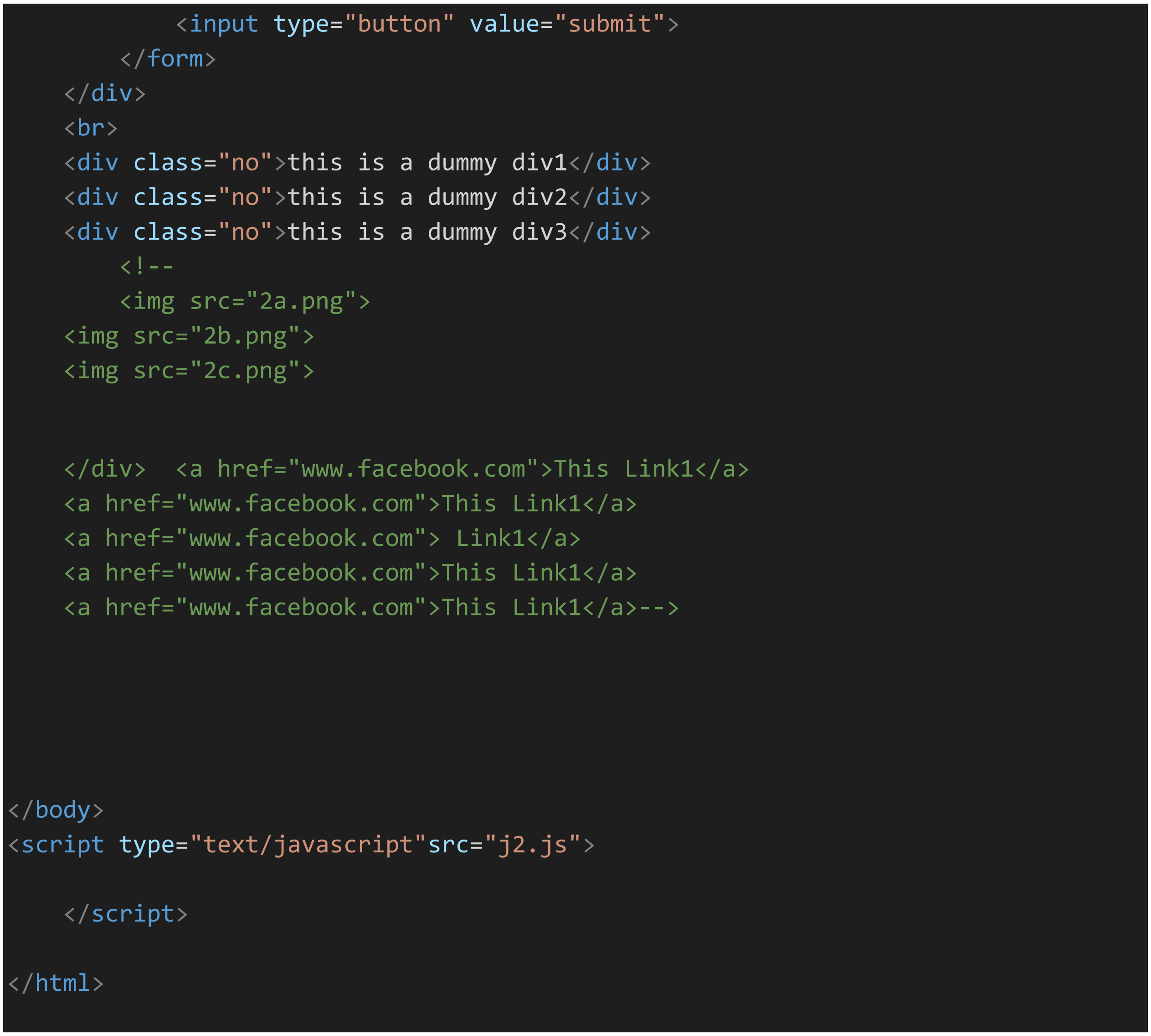
```
let element1=document.getElementsByTagName('div');
Array.from(element1).forEach(function(element){
   console.log(element)

})

let contianer=document.querySelector('div .contianer')
 console.log(contianer.lastChild)
 console.log(contianer.lastElementChild)
console.log(contianer.childElementCount)

console.log(contianer.firstElementChild)
```

```
let element=document.createElement("li")
element.className='childul';
element.id="created"
//eleemnt create kr k usko class or id de h
element.setAttribute('title','myTitle')
element.innerHTML='Hello world'

let ul=document.querySelector('ul .this')

console.log(ul)
console.log(element)
```

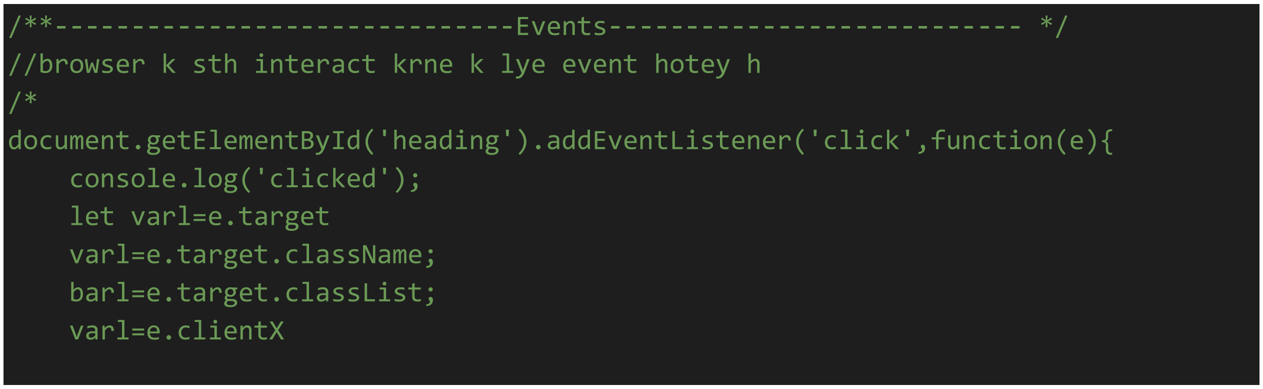
```
//console.log("Tuts 16")
//to create element in ava we use document .creat eelement

let element=document.createElement('li');
element.className='childul';
element.id='createdLI'
//console.log(element)
element.setAttribute('Title','Mytitle');
element.innerText="This is "
//ab kse fit kre dom m hume this class m dalna h toh phle this class ko slect kro
 phir us m append krdo
let ul=document.querySelector('ul.this');
//class bhi select hogaye ab append krna h
//console.log(ul)
ul.appendChild(element)

let element2=document.createElement('p')
element.innerHTML="This is created text"
element.className="created"
element.id="created"


let el=document.querySelector('div.container');
let text1=document.createTextNode("this is text node");
el.appendChild(text1)

//console.log(el)
el.appendChild(element2)

//hamesha 2 element create krne paregy ek jo bananna h dosre jis m select kr k ad
d krna h


//How to replace an item
let elem2=document.createElement('h3')
elem2.id='elem2';
elem2.className="elem2";
let note=document.createTextNode("This is created note")
elem2.appendChild(note)
let ele=document.createElement('h1');
ele.className="marium"
ele.id='marium'
ele.innerText="This is Marium"
```



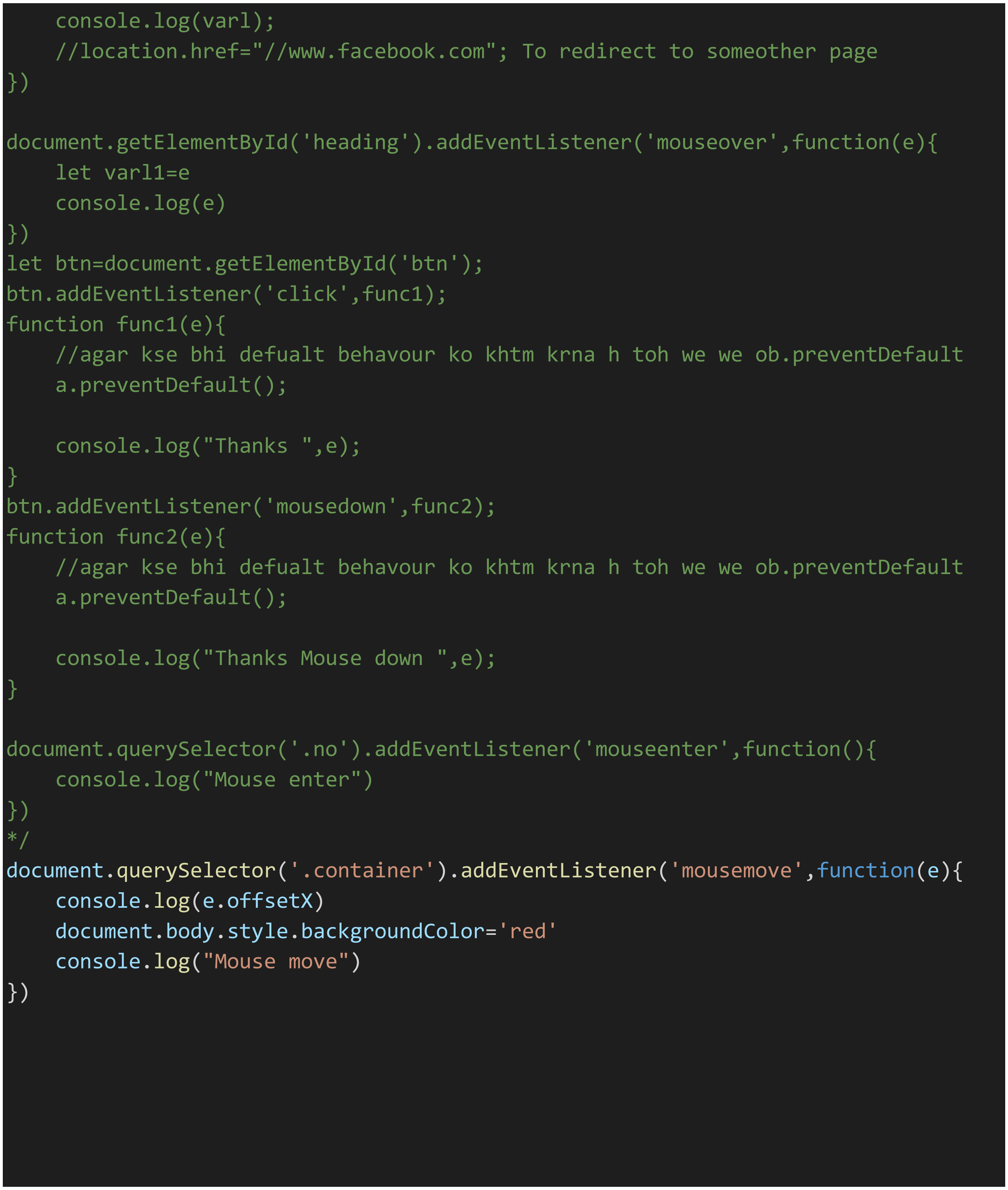
```
let ob=document.querySelector('div.container');
let myul=document.getElementById('myul')
console.log(ob)
ob.appendChild(note)
myul.replaceWith(ele,document.getElementById('fui'))
//to remove or add element zarori hona chahye k dosra ala arg phle wala ka child
ho
myul.removeChild(document.getElementById('fui'))


/**Get attribue koi bhi attroibe lene k lye
 * has attt dkhne k lye k yeh att majood h na nh
 *
 *
 */



<!DOCTYPE html>
<html>
<head>
    <title>Javascript</title>
</head>
<body>

    <div class="container">
        <h1 id="heading"> Welcome to Code With Harry</h1>
        <div id="myfirst" class="child red good" id="first">child 1

            <ul class="this" id="myul">
                <li class="childul" id="fui">this</li>
                <li class="childul">is</li>
                <li class="childul">a</li>
                <li class="childul">list </li>
                <li class="childul"  id="lui">of my dreams</li>
            </ul>
        </div>
        <div class="child">child 2</div>
        <div class="child red">child 3</div>
        <div class="child">child 4</div>
        <form action="none.html" method="post">
            <a href="//codewithharry.com">Go to Code With Harry</a>
            <br>
            <br>
            Search this website: <input type="text" name="Hello" id="">
```



```
            <input type="button" value="submit">
        </form>
    </div>
    <br>
    <div class="no">this is a dummy div1</div>
    <div class="no">this is a dummy div2</div>
    <div class="no">this is a dummy div3</div>
        <!--
        <img src="2a.png">
    <img src="2b.png">
    <img src="2c.png">


    </div>  <a href="www.facebook.com">This Link1</a>
    <a href="www.facebook.com">This Link1</a>
    <a href="www.facebook.com"> Link1</a>
    <a href="www.facebook.com">This Link1</a>
    <a href="www.facebook.com">This Link1</a>-->






</body>
<script type="text/javascript"src="j2.js">

    </script>

</html>
```



```
/**------------------------------Events--------------------------- */
//browser k sth interact krne k lye event hotey h
/*
document.getElementById('heading').addEventListener('click',function(e){
    console.log('clicked');
    let varl=e.target
    varl=e.target.className;
    barl=e.target.classList;
    varl=e.clientX
```



```
    console.log(varl);
    //location.href="//www.facebook.com"; To redirect to someother page
})

document.getElementById('heading').addEventListener('mouseover',function(e){
    let varl1=e
    console.log(e)
})
let btn=document.getElementById('btn');
btn.addEventListener('click',func1);
function func1(e){
    //agar kse bhi defualt behavour ko khtm krna h toh we we ob.preventDefault
    a.preventDefault();

    console.log("Thanks ",e);
}
btn.addEventListener('mousedown',func2);
function func2(e){
    //agar kse bhi defualt behavour ko khtm krna h toh we we ob.preventDefault
    a.preventDefault();

    console.log("Thanks Mouse down ",e);
}

document.querySelector('.no').addEventListener('mouseenter',function(){
    console.log("Mouse enter")
})
*/
document.querySelector('.container').addEventListener('mousemove',function(e){
    console.log(e.offsetX)
    document.body.style.backgroundColor='red'
    console.log("Mouse move")
})
```

Local Storage:

```
console.log('This is tut 20');
let impArray = ['adrak', 'pyaz', 'bhindi'];

// // Add a key-value pair inside local Storage
// localStorage.setItem('Name', 'Harry');
// localStorage.setItem('Name2', 'Rohan');
// localStorage.setItem('Sabzi', JSON.stringify(impArray));

// Clears the entire local storage
// localStorage.clear();

// Clear a particular key-value pair
// localStorage.removeItem('Name2');

// Retrieve an item from the local Storage
let name = localStorage.getItem('Name');
name = JSON.parse(localStorage.getItem('Sabzi'));
console.log(name)

// sessionStorage.setItem('sessionName', 'sHarry');
// sessionStorage.setItem('sessionName2', 'sRohan');
// sessionStorage.setItem('sessionSabzi', JSON.stringify(impArray));
```

Project (Note Taking)(Also in Github)

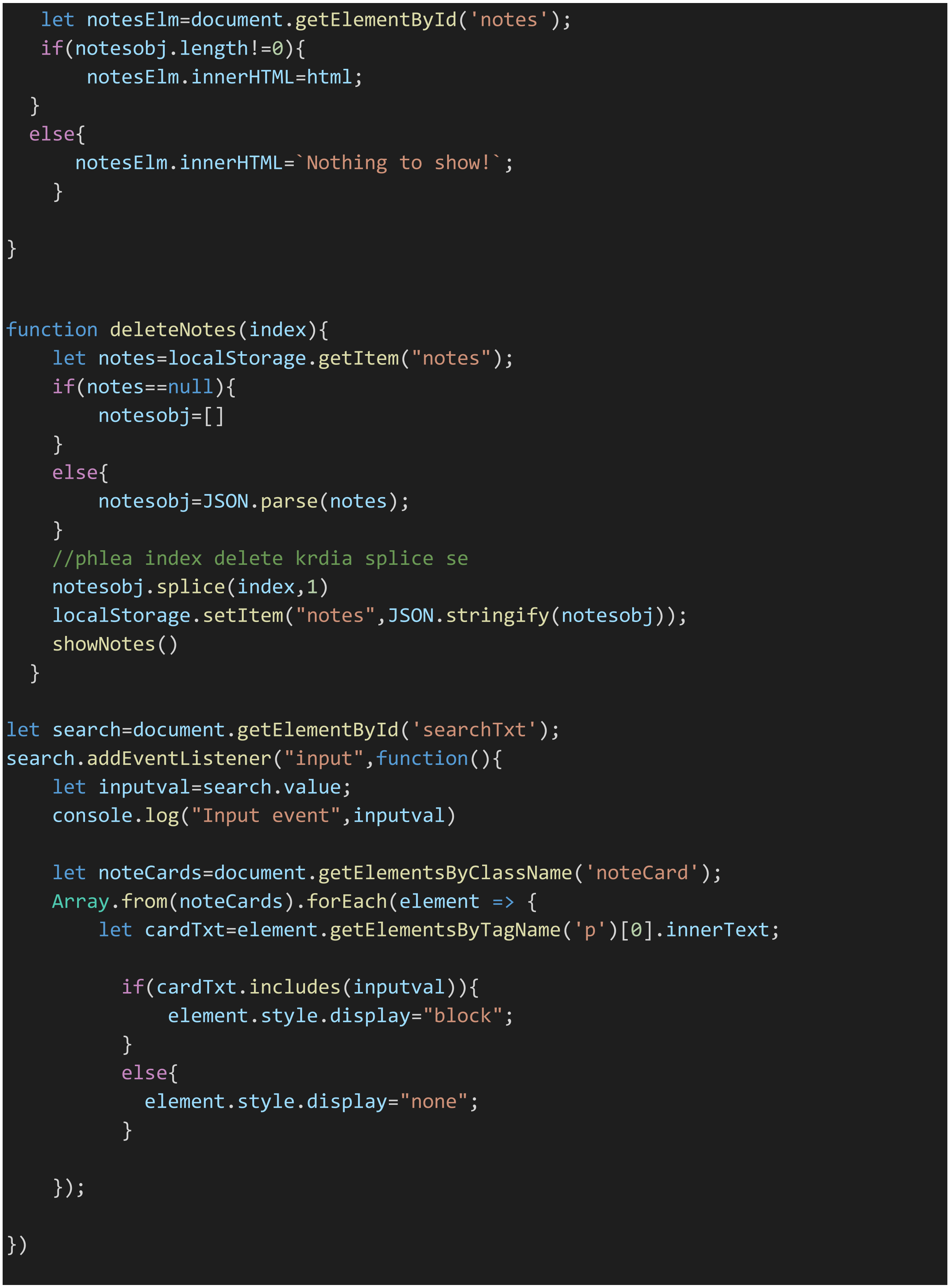
Index.hml

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

    <title>Note Apps</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <a class="navbar-brand" href="#">Marium's Notes</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
              <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
            </li>

          </ul>
          <form class="form-inline my-2 my-lg-0">
            <input id="searchTxt" class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
            <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
          </form>
        </div>
```

```
      </nav>

      <div class="container my-5">
         <h1>Welcome to Mariums's Note</h1>

        <div class="card">

          <div class="card-body">
            <h5 class="card-title">ADD A NOTE</h5>
            <div class="form-group">
              <label for="exampleFormControlTextarea1">Example textarea</label>
              <textarea class="form-control" id="addTxt" rows="3"></textarea>
            </div>
            <button id="addBtn"class="btn btn-primary">Add note</button>
          </div>
        </div>

          <h1>Your Notes</h1>
          <hr>
          <div id="notes" class="container-fluid row">


          </div>


      </div>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-
3.4.1.slim.min.js" integrity="sha384-
J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="an
onymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.mi
n.js" integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="an
onymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.
min.js" integrity="sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="an
onymous"></script>
    <script src="js/apps.js"></script>
</body>
</html>
```

Js

```
showNotes()
//if user add a note and add it to a localstorage
let addBtn=document.getElementById('addBtn').addEventListener("click",function(e)
{
    let addText=document.getElementById("addTxt")
    let notes=localStorage.getItem("notes");
    if(notes==null){
         notesobj=[];
    }
    else{
        //qk local storage se hume obj milta h or hum array m km krna charhe h is lye json.parse
        notesobj=JSON.parse(notes)
    }
    notesobj.push(addText.value);
    localStorage.setItem("notes",JSON.stringify(notesobj))
    addText.value="";
    console.log(notesobj);

})

//read then show element from localstorage
function showNotes(){
    let notes=localStorage.getItem("notes");
    if(notes==null){
        notesobj=[]
    }
    else{
        notesobj=JSON.parse(notes)
    }
    //html var use ko display dynamil in index page
    let html="";
    notesobj.forEach(function(element,index) {
       html +=
      `<div class="mx-2 my-2 card noteCard" style="width: 18rem;">
      <div class="card-body">
        <h5 class="card-title">${index+1}</h5>
        <p class="card-text">${element}</p>
        <button id="${index}" onclick="deleteNotes(this.id)" class="btn btn-primary">Delete Note</button>
      </div>
    </div>`;
});
```



```
   let notesElm=document.getElementById('notes');
   if(notesobj.length!=0){
       notesElm.innerHTML=html;
  }
  else{
      notesElm.innerHTML=`Nothing to show!`;
    }

}


function deleteNotes(index){
    let notes=localStorage.getItem("notes");
    if(notes==null){
        notesobj=[]
    }
    else{
        notesobj=JSON.parse(notes);
    }
    //phlea index delete krdia splice se
    notesobj.splice(index,1)
    localStorage.setItem("notes",JSON.stringify(notesobj));
    showNotes()
  }

let search=document.getElementById('searchTxt');
search.addEventListener("input",function(){
    let inputval=search.value;
    console.log("Input event",inputval)

    let noteCards=document.getElementsByClassName('noteCard');
    Array.from(noteCards).forEach(element => {
        let cardTxt=element.getElementsByTagName('p')[0].innerText;

          if(cardTxt.includes(inputval)){
              element.style.display="block";
          }
          else{
            element.style.display="none";
          }

    });

})
```

Maths and date

```
console.log("Maths")
let x=3
let y=6
let z;
z=x+y
z=Math.PI
z=Math.ceil(5.6)
z=Math.round(4.5)
z=Math.floor(-5.3)
z=Math.abs(-5)
z=Math.sqrt(5);
z=Math.pow(2,3);
z=Math.min(2,3,4,5,666,777)
//50 or 100 k beach ka forumula banaya h
z=Math.ceil(50+(100-50)*Math.random(0,1))

console.log(z)

//Exploeing maths

let date=new Date()
console.log(date)
let otherDate=new Date('9-27-1997')
console.log(otherDate)
let a=otherDate.getDay()
a=otherDate.getFullYear();
a=date.getTime()
a=date.getMonth();
otherDate.setHours('23')
a=otherDate
otherDate.setFullYear('1998')
console.log(a)
```

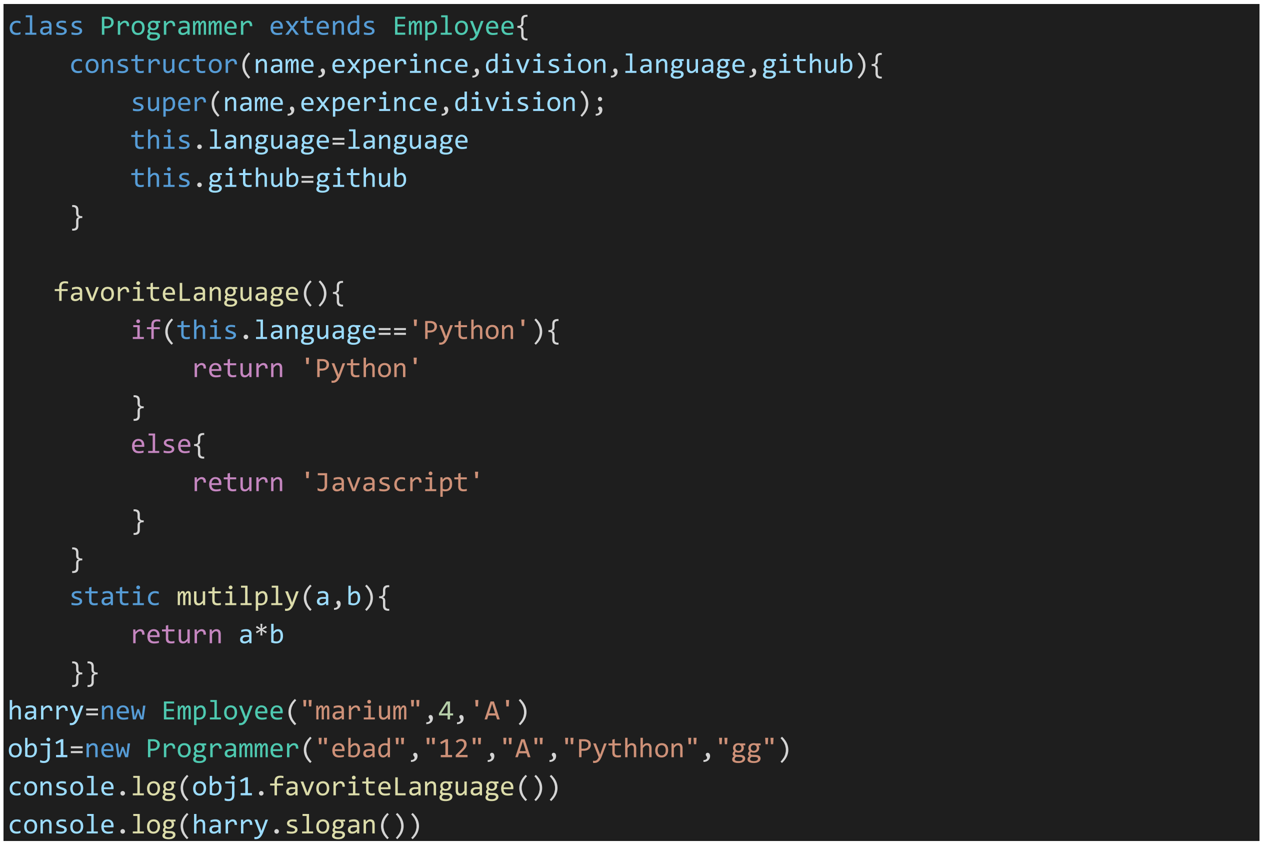
Object Oreinted Lammguage

```
//object using object literals
let car={
    name:"Maruti 800",
    topSpeed:66,
    run:function(){
        console.log("Car is running");
    }
}
console.log(car)
//constructor
//new Date() -> This () is constructor
//creating cons

function GeneralCar(name,topSpeed){
    this.name=name;
    this.topSpeed=topSpeed;
    this.run=function(){
        console.log(`${this.name} is running`)
    }
    this.analyze=function(){
        console.log(`this car is slower by ${200-this.topSpeed} by mercedess`)
    }
}

car1 = new GeneralCar("Nissan",180)
//Object literal
let obj={
    name:"marium",
    channel:"Marium hhaha",
    "address":"Dnia"
}

function Obj(givenName){
   this.name=givenName
}

Obj.prototype.getName=function()
{
    return this.name
}
Obj.prototype.setName=function(newName){
    this.name=newName
}
```

```
let obj2=new Obj("Ebad")
console.log(obj2)
```

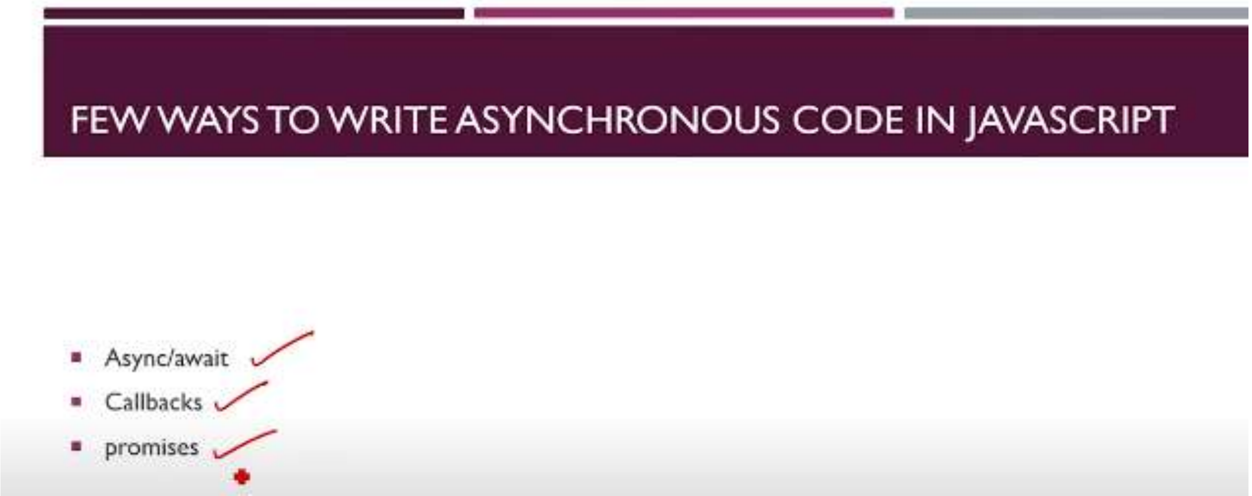
ES6 Classes and Inheritance

```
class Employee{
    constructor(name,experince,division){
        this.name=name;
        this.experince=experince;
        this.division=division;
    }
    slogan(){
        return `I am ${this.name} and this is company`
    }
    joiningYear()
    {
        return 2020-this.experince
    }
    static add(a,b){
        return a+b
    }
   }
```



```
class Programmer extends Employee{
    constructor(name,experince,division,language,github){
        super(name,experince,division);
        this.language=language
        this.github=github
    }

   favoriteLanguage(){
        if(this.language=='Python'){
            return 'Python'
        }
        else{
            return 'Javascript'
        }
    }
    static mutilply(a,b){
        return a*b
    }}
harry=new Employee("marium",4,'A')
obj1=new Programmer("ebad","12","A","Pythhon","gg")
console.log(obj1.favoriteLanguage())
console.log(harry.slogan())
```

What is Asynchronous Programming?



FEW WAYS TO WRITE ASYNCHRONOUS CODE IN JAVASCRIPT

- Async/await ✓
- Callbacks ✓
- promises ✓

Assy

```
//Synchornous
setInterval(() => {
    for (let index = 0; index < 4000; index++) {
        const element = index;
        console.log("This is index number"+index)
    }

}, 100);
//set timeout se yeh assn hogaya ab wo for loop chlta rahega apne time se or dosr
a km bhi hota rahegaa
console.log("Done printing")
```

CallBack:

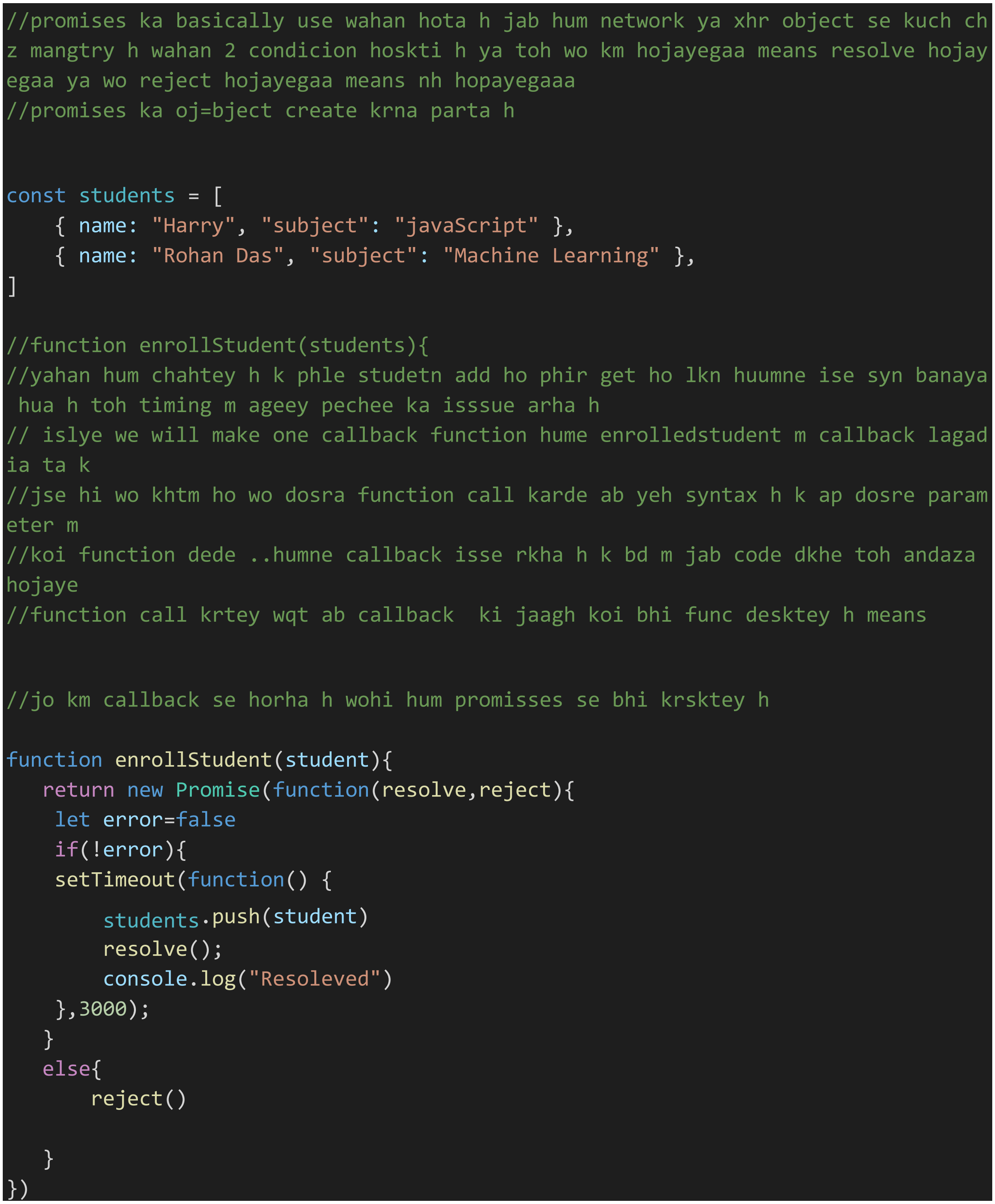
```
//Synchornous

/*setInterval(() => {
    for (let index = 0; index < 4000; index++) {
        const element = index;
        console.log("This is index number"+index)
    }

}, 100);
//set timeout se yeh assn hogaya ab wo for loop chlta rahega apne time se or dosr
a km bhi hota rahegaa
```

```
console.log("Done printing")*/
const students = [
    { name: "Harry", "subject": "javaScript" },
    { name: "Rohan Das", "subject": "Machine Learning" },
]

//function enrollStudent(students){
//yahan hum chahtey h k phle studetn add ho phir ead ho lkn huumne ise syn banaya hua h toh timing m ageey pechee ka isssue arha h
// islye we will make one callback function hume enrolledstudent m callback lagadia ta k
//jse hi wo khtm ho wo dosra function call karde ab yeh syntax h k ap dosre parameter m
//koi function dede ..humne callback isse rkha h k bd m jab code dkhe toh andaza hojaye
//function call krtey wqt ab callback  ki jaagh koi bhi func desktey h means
function enrollStudent(student,callback){
    setTimeout(function() {
        students.push(student)
        callback();
    },3000);

}

function getStudent(){
setTimeout(function() {
    let str="";
    students.forEach(student=> {
        str+=`<li>${student.name}</li>`
});
body=document.getElementById("student");
body.innerHTML=str
},1000);  }
let newStudent={'name':"Heehehe","subject":"Rona"}
  enrollStudent(newStudent, getStudent)
```

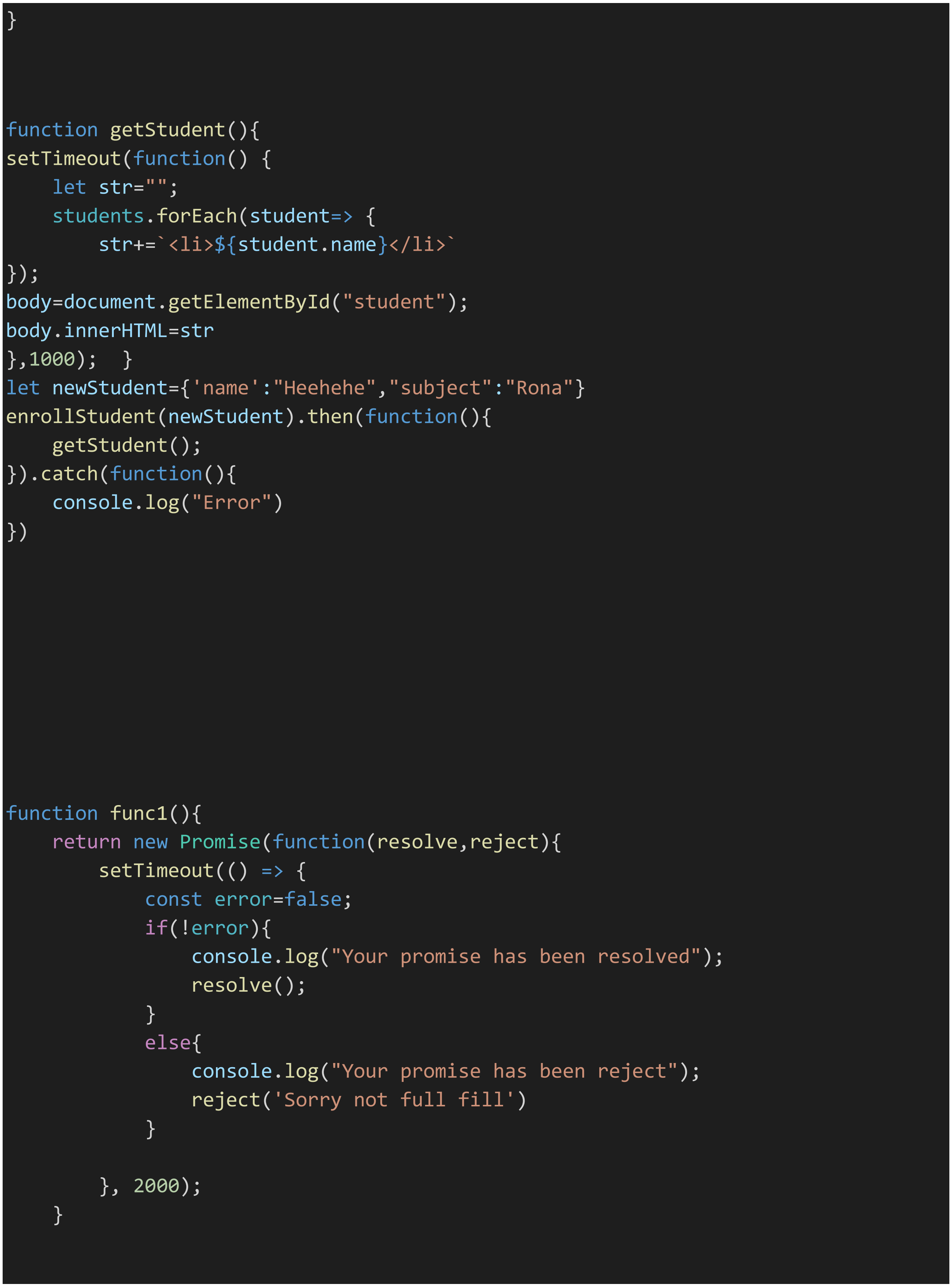
Imdex of Callback

```
 <h1>Student List</h1>
    <ul id="student"></ul>
```

Promises Basics, Promise.then() & Promise.catch() :



```
//promises ka basically use wahan hota h jab hum network ya xhr object se kuch chz mangtry h wahan 2 condicion hoskti h ya toh wo km hojayegaa means resolve hojayegaa ya wo reject hojayegaa means nh hopayegaaa
//promises ka oj=bject create krna parta h


const students = [
    { name: "Harry", "subject": "javaScript" },
    { name: "Rohan Das", "subject": "Machine Learning" },
]

//function enrollStudent(students){
//yahan hum chahtey h k phle studetn add ho phir get ho lkn huumne ise syn banaya hua h toh timing m ageey pechee ka isssue arha h
// islye we will make one callback function hume enrolledstudent m callback lagadia ta k
//jse hi wo khtm ho wo dosra function call karde ab yeh syntax h k ap dosre parameter m
//koi function dede ..humne callback isse rkha h k bd m jab code dkhe toh andaza hojaye
//function call krtey wqt ab callback  ki jaagh koi bhi func desktey h means


//jo km callback se horha h wohi hum promisses se bhi krsktey h

function enrollStudent(student){
   return new Promise(function(resolve,reject){
    let error=false
    if(!error){
    setTimeout(function() {
        students.push(student)
        resolve();
        console.log("Resoleved")
    },3000);
   }
   else{
       reject()

   }
})
```



```
}



function getStudent(){
setTimeout(function() {
    let str="";
    students.forEach(student=> {
        str+=`<li>${student.name}</li>`
});
body=document.getElementById("student");
body.innerHTML=str
},1000);  }
let newStudent={'name':"Heehehe","subject":"Rona"}
enrollStudent(newStudent).then(function(){
    getStudent();
}).catch(function(){
    console.log("Error")
})








function func1(){
    return new Promise(function(resolve,reject){
        setTimeout(() => {
            const error=false;
            if(!error){
                console.log("Your promise has been resolved");
                resolve();
            }
            else{
                console.log("Your promise has been reject");
                reject('Sorry not full fill')
            }

        }, 2000);
    }
    )}

    func1().then(function(){
        console.log("Thanks for resolving")
    }).catch(function(){
        console.log("Bad")
    })
```

Arrow functions in Javascript

```
//we have three ways of defining function
function harry1(){
    console.log("Method 1 of defining Function")}
const harry2=function(){
    console.log("Method 2 of defining Function")}
const harry3=()=>{console.log("Method 3 of defining Function")}
harry1()
harry2()
harry3()
const greet= ()=> "Good Morning";
console.log(greet())
const greet1 = (name,ending)=> "Hello" + name + "Good Morning" +ending
console.log(greet1("Marium","Now okay bye"))
```

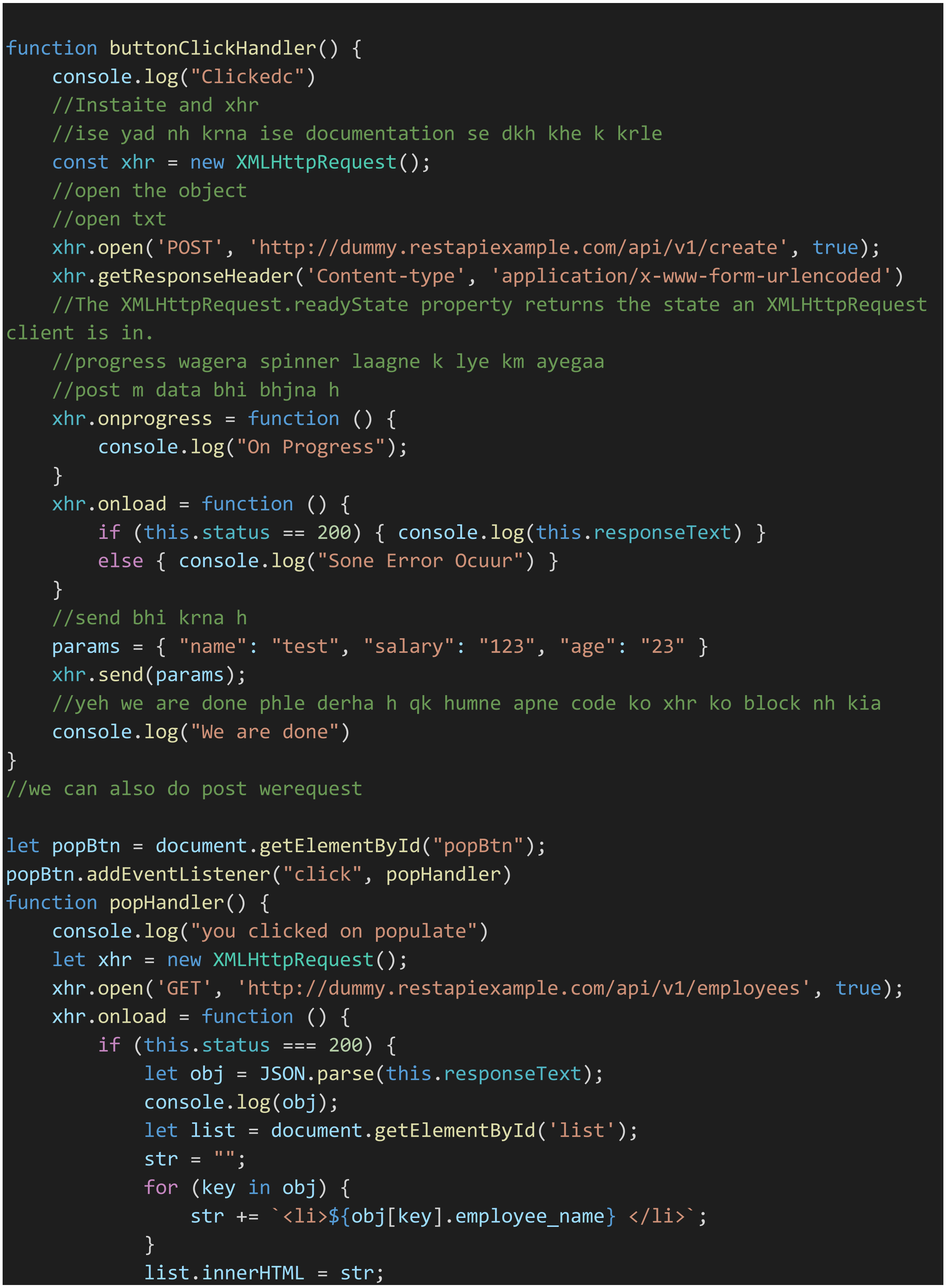
Fetch api

```
console.log("Fetch API")
let myBtn=document.getElementById("myBtn")
let content=document.getElementById("content")

function getData(){
    url="harry.txt";
    fetch('harry.txt').then((response)=>{
        return response.text;
    }).then((data)=> {
        console.log(data)
    }
    )
}

getData()
```

AJAX

JS:

```
let fetchBtn = document.getElementById('fetchBtn');
fetchBtn.addEventListener('click', buttonClickHandler)
/*function buttonClickHandler(){
    console.log("Clickedc")
    //Instaite and xhr
    //ise yad nh krna ise documentation se dkh khe k krle
    const xhr=new XMLHttpRequest();
    //open the object
    //open txt
    xhr.open('GET','https://jsonplaceholder.typicode.com/todos/1',true);

    //The XMLHttpRequest.readyState property returns the state an XMLHttpRequest
client is in.
    //progress wagera spinner laagne k lye km ayegaa

    xhr.onprogress=function(){
        console.log("On Progress");
    }
    xhr.onload=function(){
        if(this.status==200){
        console.log(this.responseText)}
        else{
            console.log("Sone Error Ocuur")
        }
    }
    //send bhi krna h
    xhr.send();
  //yeh we are done phle derha h qk humne apne code ko xhr ko block nh kia
    console.log("We are done")
}*/
```



```
function buttonClickHandler() {
    console.log("Clickedc")
    //Instaite and xhr
    //ise yad nh krna ise documentation se dkh khe k krle
    const xhr = new XMLHttpRequest();
    //open the object
    //open txt
    xhr.open('POST', 'http://dummy.restapiexample.com/api/v1/create', true);
    xhr.getResponseHeader('Content-type', 'application/x-www-form-urlencoded')
    //The XMLHttpRequest.readyState property returns the state an XMLHttpRequest
client is in.
    //progress wagera spinner laagne k lye km ayegaa
    //post m data bhi bhjna h
    xhr.onprogress = function () {
        console.log("On Progress");
    }
    xhr.onload = function () {
        if (this.status == 200) { console.log(this.responseText) }
        else { console.log("Sone Error Ocuur") }
    }
    //send bhi krna h
    params = { "name": "test", "salary": "123", "age": "23" }
    xhr.send(params);
    //yeh we are done phle derha h qk humne apne code ko xhr ko block nh kia
    console.log("We are done")
}
//we can also do post werequest

let popBtn = document.getElementById("popBtn");
popBtn.addEventListener("click", popHandler)
function popHandler() {
    console.log("you clicked on populate")
    let xhr = new XMLHttpRequest();
    xhr.open('GET', 'http://dummy.restapiexample.com/api/v1/employees', true);
    xhr.onload = function () {
        if (this.status === 200) {
            let obj = JSON.parse(this.responseText);
            console.log(obj);
            let list = document.getElementById('list');
            str = "";
            for (key in obj) {
                str += `<li>${obj[key].employee_name} </li>`;
            }
            list.innerHTML = str;
        }
        else {
            console.log("Some error occured")
        }
    }


    xhr.send()

}
```

Index:

```
<h1>Ajax tutorial</h1>
    <button type="button" id="fetchBtn" class="btn btn-primary">Fetch Data</button>
    <button type="button" id="popBtn" class="btn btn-secondary">Populate</button>

    <div class="container">
        <h1>Employee list</h1>
        <ul id="list">

        </ul>
    </div>
```

Fetch API:

```
//fetch api is a new and modern standard we will kostily use fetch API

let myBtn=document.getElementById("myBtn");
let content=document.getElementById("content");

//fetch return a promise
//fetch m hamesha 2 then lagtey h
function getData(){
    console.log("Starting")
    url="https://api.github.com/users";
    fetch(url).then((response)=>{
        console.log("Inside 1st then")
        //repsonse.text se hume text return hoga wrna agar hum json lkhdety toh hume json m milega
        //return response.text()
        return response.json()
    }).then((data)=>{
        console.log("Insdie 2nd tthen")
        console.log(data);
    })
}
function postData(){
    url="   http://dummy.restapiexample.com/api/v1/create";
    data=   '{"name":"Marium12","salary":"123","age":"23"}';
    params={
        method:'post',
        headers:{
            'Content-Type':'application/json'},
        body:data}

    fetch(url,params).then(response=>response.json())
    .then(data=>console.log(data))
}

postData();
//yeh iska syntax h abhi k ly yad rklo post m hamesha aise hi hoga
```

Async/Await in Javascript | JavaScript Tutorial In Hindi #43

```
//Asyn/Await m ahmesha 2 promises hotey h ek promise hota h jo khud promise return krta h
//asyn lagnese hamesha ek promise return hota h jis bhi func m asyn lkhahota h wo return krta h promise
function harry(){
    return "Harry"
}

console.log("Before function calling ")
```

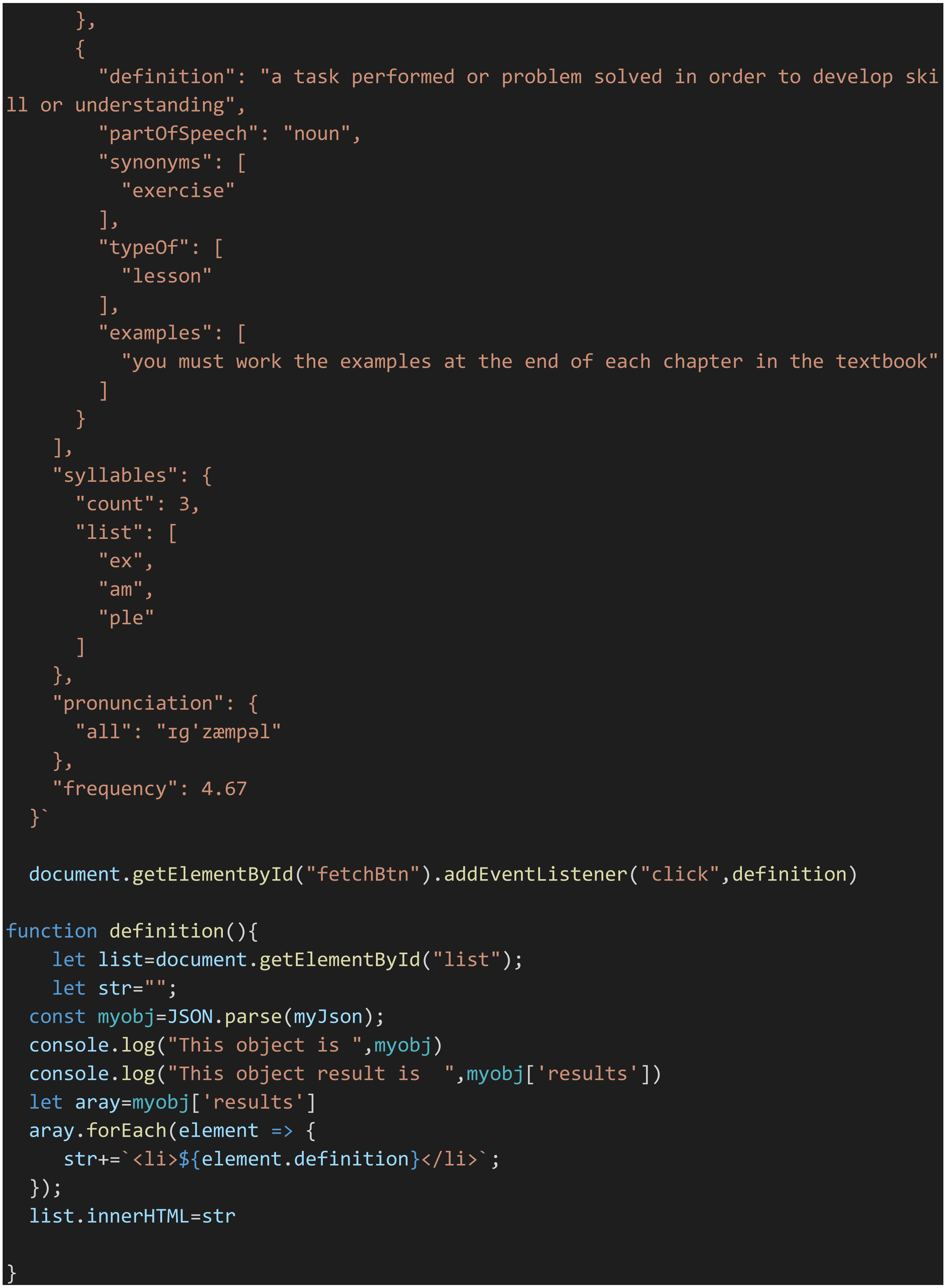
```
let a=harry();
console.log(a)
```

Error Handling:

```
//Error handling agar koi error aye
let a=undefined;
if(a != undefined){
   throw new Error("This is not undefined")
}
else{
    console.log("This is  undefined")
}
try {
    console.log("We are inside try block")
    functionHarry()
} catch (error) {
    console.log("Are you Okay")
    console.log(error);
    console.log(error.name)
    console.log(error.message)

}finally{
   console.log("chahe error aye ya na aye we will run this")
}
```

Using JavaScript Objects Exercise 5: Solution

```
let myJson=`{
    "word": "example",
    "results": [
      {
        "definition": "a representative form or pattern",
        "partOfSpeech": "noun",
        "synonyms": [
          "model"
        ],
        "typeOf": [
          "representation",
          "internal representation",
          "mental representation"
        ],
        "hasTypes": [
          "prefiguration",
          "archetype",
          "epitome",
          "guide",
          "holotype",
          "image",
          "loadstar",
          "lodestar",
          "microcosm",
          "original",
          "paradigm",
          "pilot",
          "prototype",
          "template",
          "templet",
          "type specimen"
        ],
        "derivation": [
          "exemplify"
        ],
        "examples": [
          "I profited from his example"
        ]
      },
      {
        "definition": "something to be imitated",
        "partOfSpeech": "noun",
        "synonyms": [
          "exemplar",
          "good example",
          "model"
        ],
        "typeOf": [
          "ideal"
        ],
        "hasTypes": [
          "pacemaker",
          "pattern",
          "beauty",
          "prodigy",
          "beaut",
          "pacesetter"
        ],
        "derivation": [
          "exemplify",
          "exemplary"
        ]
      },
      {
        "definition": "an occurrence of something",
        "partOfSpeech": "noun",
        "synonyms": [
          "case",
          "instance"
        ],
        "typeOf": [
          "happening",
          "natural event",
          "occurrence",
          "occurrent"
        ],
        "hasTypes": [
          "clip",
          "mortification",
          "piece",
          "time",
          "humiliation",
          "bit"
        ],
        "derivation": [
          "exemplify"
        ],
        "examples": [
          "but there is always the famous example of the Smiths"
        ]
      },
      {
        "definition": "an item of information that is typical of a class or group
",
        "partOfSpeech": "noun",
        "synonyms": [
          "illustration",
          "instance",
          "representative"
        ],
        "typeOf": [
          "information"
        ],
        "hasTypes": [
          "excuse",
          "apology",
          "specimen",
          "case in point",
          "sample",
          "exception",
          "quintessence",
          "precedent"
        ],
        "derivation": [
          "exemplify",
          "exemplary"
        ],
        "examples": [
          "this patient provides a typical example of the syndrome",
          "there is an example on page 10"
        ]
      },
      {
        "definition": "punishment intended as a warning to others",
        "partOfSpeech": "noun",
        "synonyms": [
          "deterrent example",
          "lesson",
          "object lesson"
        ],
        "typeOf": [
          "monition",
          "admonition",
          "word of advice",
          "warning"
        ],
        "derivation": [
          "exemplary"
        ],
        "examples": [
          "they decided to make an example of him"
        ]
```



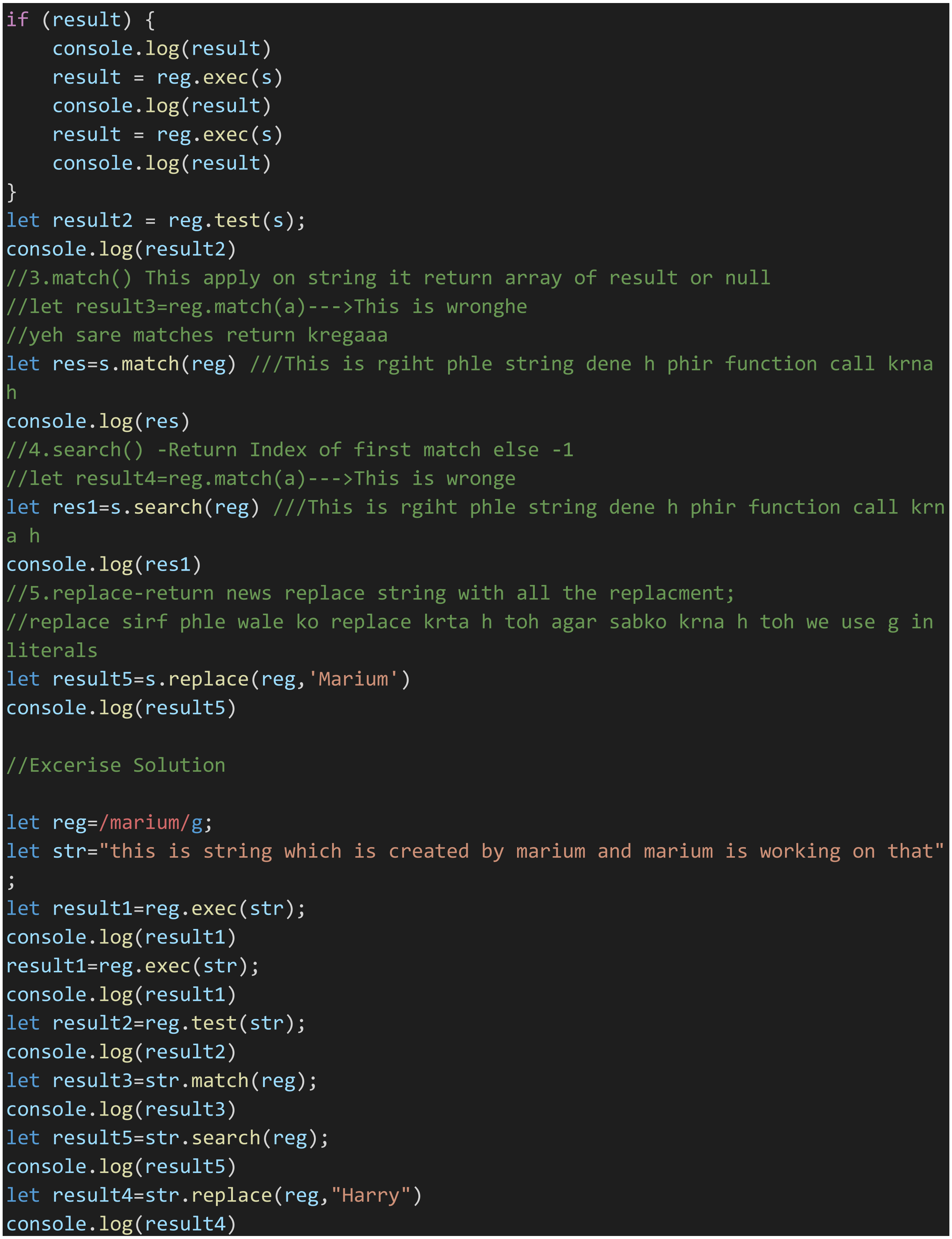
```
   },
   {
    "definition": "a task performed or problem solved in order to develop ski
ll or understanding",
    "partOfSpeech": "noun",
    "synonyms": [
     "exercise"
    ],
    "typeOf": [
     "lesson"
    ],
    "examples": [
     "you must work the examples at the end of each chapter in the textbook"
    ]
   }
  ],
  "syllables": {
   "count": 3,
   "list": [
    "ex",
    "am",
    "ple"
   ]
  },
  "pronunciation": {
   "all": "ɪg'zæmpəl"
  },
  "frequency": 4.67
 }`

 document.getElementById("fetchBtn").addEventListener("click",definition)

function definition(){
  let list=document.getElementById("list");
  let str="";
 const myobj=JSON.parse(myJson);
 console.log("This object is ",myobj)
 console.log("This object result is  ",myobj['results'])
 let aray=myobj['results']
 aray.forEach(element => {
   str+=`<li>${element.definition}</li>`;
 });
 list.innerHTML=str

}
```

Index.html

```html
<div class="container">
      <h1>Employee list</h1>
      <ul id="list">

      </ul>
  </div>

  <div class="content">

  </div>
```

JavaScript Regular Expressions & related functions

```javascript
console.log("exce agar hota h majood text toh yeh return krta h array,    test agar h toh return true else false then we have match method os tring class agar hota h toh it return array  then we have search function then replcawith")
console.log("Hello Reg");
let reg = /harry/g;//g means global
reg = /harry/i; //i flag means case-inseentive
console.log(reg)
//regular expression ko banaye k lye we need reg literals
//source hota h k reg eexp ki literal m kia lkha tha uska content
console.log(reg.source)
//Function to Match Expression;
//hum direclty wagera bhi search krsktry h basically reg searching m hi use hota h
let s = "This is a great code with Harry and also harry bhai ";
//1.exe() This return array for match and null for no-match
//yeh match ka index btata h
//mere pass 2 harry h but ek hi show horhaa h qk mne apne string ko globalThis; flag nh dia reg define krtey wqt agar m g dedo toh wo mjhe 2no locations dega
//2-test funcction means return true or false:agar wo reg expression h majood toh true milega wrna false milega
let result = reg.exec(s)
```



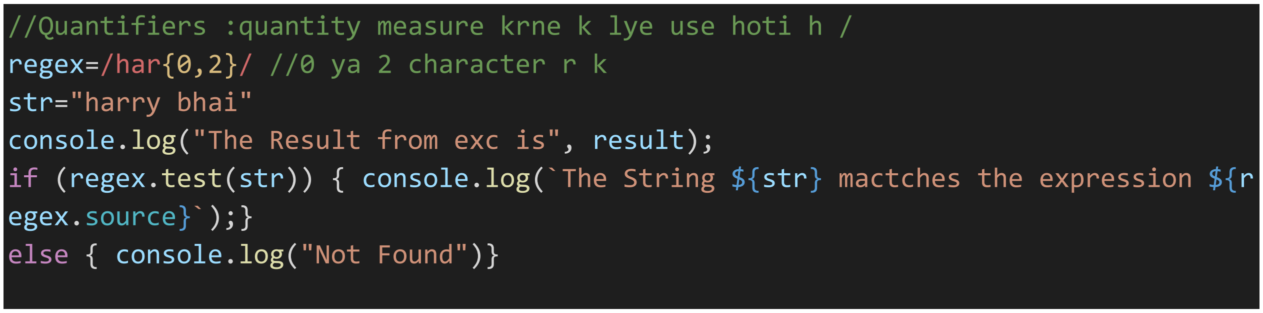
```
if (result) {
    console.log(result)
    result = reg.exec(s)
    console.log(result)
    result = reg.exec(s)
    console.log(result)
}
let result2 = reg.test(s);
console.log(result2)
//3.match() This apply on string it return array of result or null
//let result3=reg.match(a)--->This is wronghe
//yeh sare matches return kregaaa
let res=s.match(reg) ///This is rgiht phle string dene h phir function call krna
h
console.log(res)
//4.search() -Return Index of first match else -1
//let result4=reg.match(a)--->This is wronge
let res1=s.search(reg) ///This is rgiht phle string dene h phir function call krn
a h
console.log(res1)
//5.replace-return news replace string with all the replacment;
//replace sirf phle wale ko replace krta h toh agar sabko krna h toh we use g in
literals
let result5=s.replace(reg,'Marium')
console.log(result5)

//Excerise Solution

let reg=/marium/g;
let str="this is string which is created by marium and marium is working on that"
;
let result1=reg.exec(str);
console.log(result1)
result1=reg.exec(str);
console.log(result1)
let result2=reg.test(str);
console.log(result2)
let result3=str.match(reg);
console.log(result3)
let result5=str.search(reg);
console.log(result5)
let result4=str.replace(reg,"Harry")
console.log(result4)
```

Regular Expressions - Metacharacters in JavaScript

```
//metacharacter symbol
let regx=/harry/;
regx =/^h/; //^ is work like a * agar koi h se shuru horhe h toh show krdo
regx=/ry$/; //akhir wale ko match krrhe h
regx=/h.rry/; //koi bhi ek character(dot) .
regx=/h*rry/; //matches any 0 or more characters
let str="harry is a good boy and harry is code with harry";
let result=regx.exec(str)
console.log("The result from exce is",result);
if(regx.test(str)){
    console.log(`The string ${str} ${regx.source} matches the expression `)
}
else{
    console.log("No match found")
}
```

Regular Expressions - Character sets

```
//character sets
let regex = /h[a-z]rry/i;
regex = /h[0-9]rry/i; //can be -9
regex = /h[aty]rry/i; //can be aty
regex = /h[a-z]rry/i; //
//yeh [] character set k lye h k is jaagh a-z m kuch hotaa ya agar m yahan lkhdo
//[0-9 tih bhi match nh hoga
regex = /ha[^aty]ry/ // chahti k k a-
x k beach m koi valu na ho agar mjhe aty ko match nh krwanaa
//square brcket laga kr us k beach m kuch dalna or agar wo  match hojata h toh aj
ayee this is called characters set
regex = /h[^aty]rr[yu]/
regex = /h[a-zA-Z]rr[yu]/
let str = "harry bhai";
str = "hirrt bhai" //This will not match
str = "hirry bhai" //This will match
str="hArry Bhai"
let result = regex.exec(str);
```



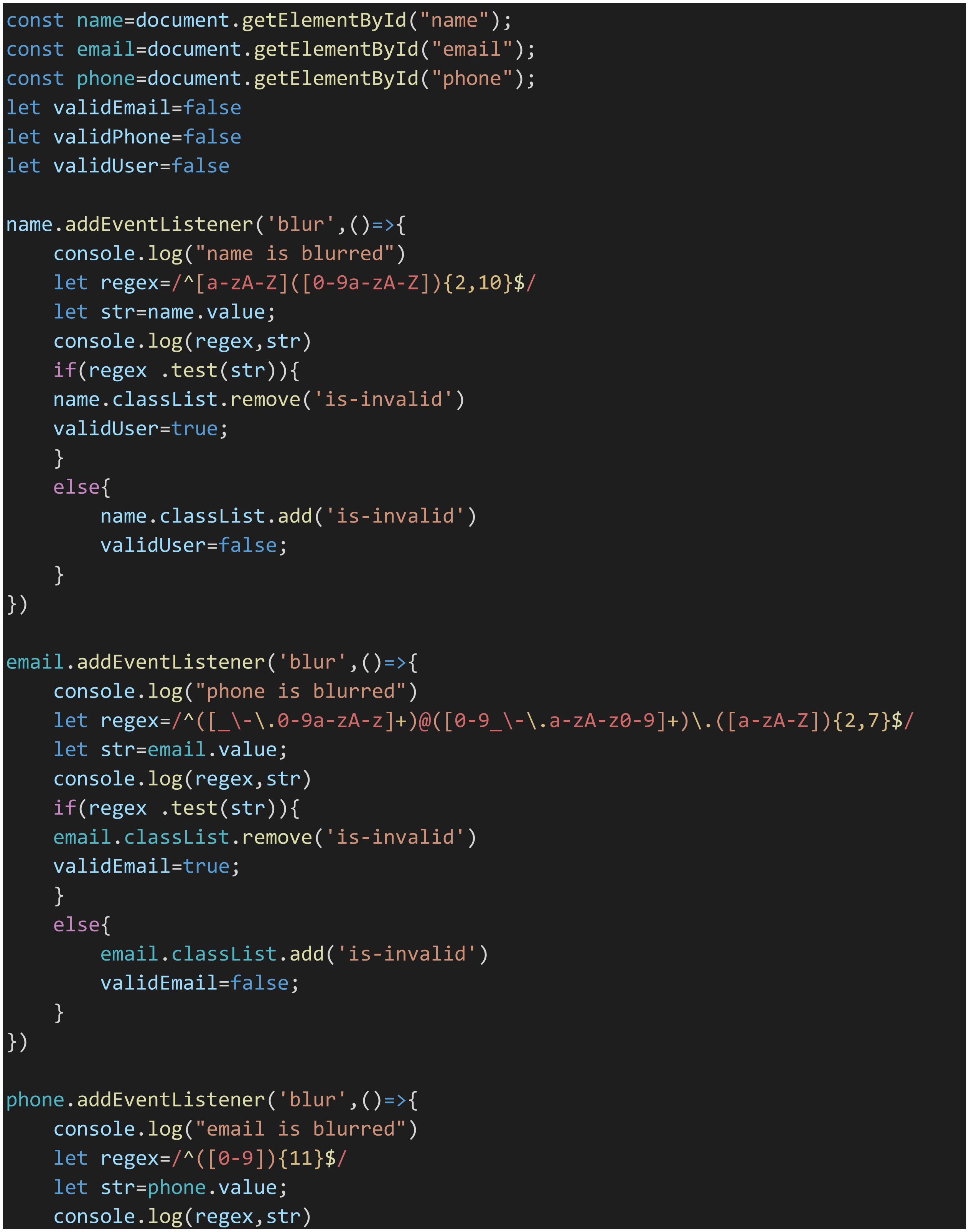
```
//Quantifiers :quantity measure krne k lye use hoti h /
regex=/har{0,2}/ //0 ya 2 character r k
str="harry bhai"
console.log("The Result from exc is", result);
if (regex.test(str)) { console.log(`The String ${str} mactches the expression ${regex.source}`);}
else { console.log("Not Found")}
```

baki code with harry k website m dkhne h qk mjhe ye itna smjh nh arha tha

Shorthand character classes:

Project 4:Form-validation Regular Expression:

Js file:



```
const name=document.getElementById("name");
const email=document.getElementById("email");
const phone=document.getElementById("phone");
let validEmail=false
let validPhone=false
let validUser=false

name.addEventListener('blur',()=>{
    console.log("name is blurred")
    let regex=/^[a-zA-Z]([0-9a-zA-Z]){2,10}$/
    let str=name.value;
    console.log(regex,str)
    if(regex .test(str)){
    name.classList.remove('is-invalid')
    validUser=true;
    }
    else{
        name.classList.add('is-invalid')
        validUser=false;
    }
})

email.addEventListener('blur',()=>{
    console.log("phone is blurred")
    let regex=/^([_\-\.0-9a-zA-z]+)@([0-9_\-\.a-zA-z0-9]+)\.([a-zA-Z]){2,7}$/
    let str=email.value;
    console.log(regex,str)
    if(regex .test(str)){
    email.classList.remove('is-invalid')
    validEmail=true;
    }
    else{
        email.classList.add('is-invalid')
        validEmail=false;
    }
})

phone.addEventListener('blur',()=>{
    console.log("email is blurred")
    let regex=/^([0-9]){11}$/
    let str=phone.value;
    console.log(regex,str)
    if(regex .test(str)){
        phone.classList.remove('is-invalid')
        validPhone=true;
    }
    else{
        phone.classList.add('is-invalid')
        validPhone=false;
    }
})

let submit=document.getElementById("submit")
submit.addEventListener("click",(e)=>{
    e.preventDefault();
    let failure=document.getElementById("failure")
    let success=document.getElementById("success")
    if(validEmail && validPhone && validUser){
        console.log("All Vaid")
        success.classList.add("show")
        $('#failure').alert('close');

    }
    else{
        console.log("one may be invalid")

       failure.classList.add("show")
       $('#success').alert('close');
    }



})

console.log(validEmail,validPhone,validUser)
```

Index.html:

```
<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1
/css/bootstrap.min.css"
    integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="an
onymous">

  <title>Pakistan's Travel Desk!</title>
</head>
<body>
  <div id="success" class="alert alert-success alert-
dismissible fade" role="alert">
    <strong>Success!</strong> Your travel request has been successfully submitted
    <button type="button" class="close" data-dismiss="alert" aria-label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
  </div>
  <div id="failure" class="alert alert-danger alert-
dismissible fade" role="alert">
    <strong>Error!</strong> Your travel request has not been send due to Error
    <button type="button" class="close" data-dismiss="alert" aria-label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
  </div>
  <div class="container">
    <h1>Pakistan's Travel Desk</h1>
    <form>
      <div class="form-group">
        <label for="name">userame</label>
        <input type="text" class="form-
control" id="name" placeholder="Enter your name">
        <small id="nameValid" class="form-text text-muted invalid-feedback">
          Your username must be 2-10 long.Must Start with aplhabet
        </small>
      </div>
      <div class="form-group">
        <label for="email">Email Address</label>
        <input type="text" class="form-
control" id="email" placeholder="Enter your email">
        <small id="phoneValid" class="form-text text-muted invalid-feedback">
          Your email is invalid
        </small>
      </div>
      <div class="form-group">
```

```
        <label for="car">Select your car</label>
        <select class="form-control" id="car">
          <option>Omni</option>
          <option>Maruti 800</option>
          <option>Ford Titanium</option>
          <option>Audi A4</option>
        </select>
      </div>
      <div class="form-group">
        <label for="address">Enter your Address</label>
        <textarea class="form-control" id="address" rows="3"></textarea>
      </div>
      <div class="form-group">
        <label for="phone">Enter your phone number</label>
        <input type="phone" class="form-
control" id="phone" placeholder="Enter your Phone Number">
        <small id="phoneValid" class="form-text text-muted invalid-feedback">
          Number must be in 11 digit long
        </small>
      </div>
      <div class="form-group">
        <label for="exampleFormControlTextarea1">Any other Message</label>
        <textarea class="form-control" id="msg" rows="3"></textarea>
      </div>
      <button id="submit" class="btn btn-primary">
        Submit
      </button>
    </form>
  </div>
  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="js/apps.js"></script>
  <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-
J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
    crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.
js"
    integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
    crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.mi
n.js"
    integrity="sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
```

```
    crossorigin="anonymous"></script>
</body>
</html>
```

Iterators in JavaScript

```
const myArray = ['Apples', 'Grapes', 'Oranges', 'Bhindi'];
console.log(myArray)

function fruitsIterators(array) {
    let nextIndex = 0;
    //we will return an object
    return {
        next: function () {
            if (nextIndex < array.length) {
                return {//we will return this object:
                    value: array[nextIndex++],
                    done: false}
                 }
            else {
                return{ done: true }
            }
        }

    }
}
const fruits=fruitsIterators(myArray)
console.log(fruits.next().value)
console.log(fruits.next().value)
console.log(fruits.next().value)
console.log(fruits.next().value)
console.log(fruits.next().value)
```

AlarmClock

Js:

```
console.log("HH")
let alarmSubmit=document.getElementById("alarmSubmit");
```

```
alarmSubmit.addEventListener("click",setAlarm)
function setAlarm(e){
    e.preventDefault();
    let alarm=document.getElementById("alarm");
    console.log(`Setting Alarm...${alarm.value}`);
    alarmDate=new Date(alarm.value);
    console.log(alarmDate)
    let now=new Date();
    console.log(alarmDate-now);
    let timeToAlarm=alarmDate-now;
    if(timeToAlarm >= 0){
      setTimeout(() => {
          ringBell();
      }, timeToAlarm);
    }
    else{
        console.log("Will not bell")
    }
}
function ringBell(){
var audio = new Audio('https://interactive-examples.mdn.mozilla.net/media/examples/t-rex-roar.mp3');
audio.play();
}
```

Index:

```
  <form>
    <div class="form-group">
      <label for="alaram">Set Alarm </label>
      <input type="text" class="form-control" id="alarm" name="alarm" placeholder="Enter alaram in yyyy-mm-dd hr:mn:ss">
        </div>

    <button type="text" id="alarmSubmit" class="btn btn-primary">Set Alarm</button>
  </form>

</div>
```

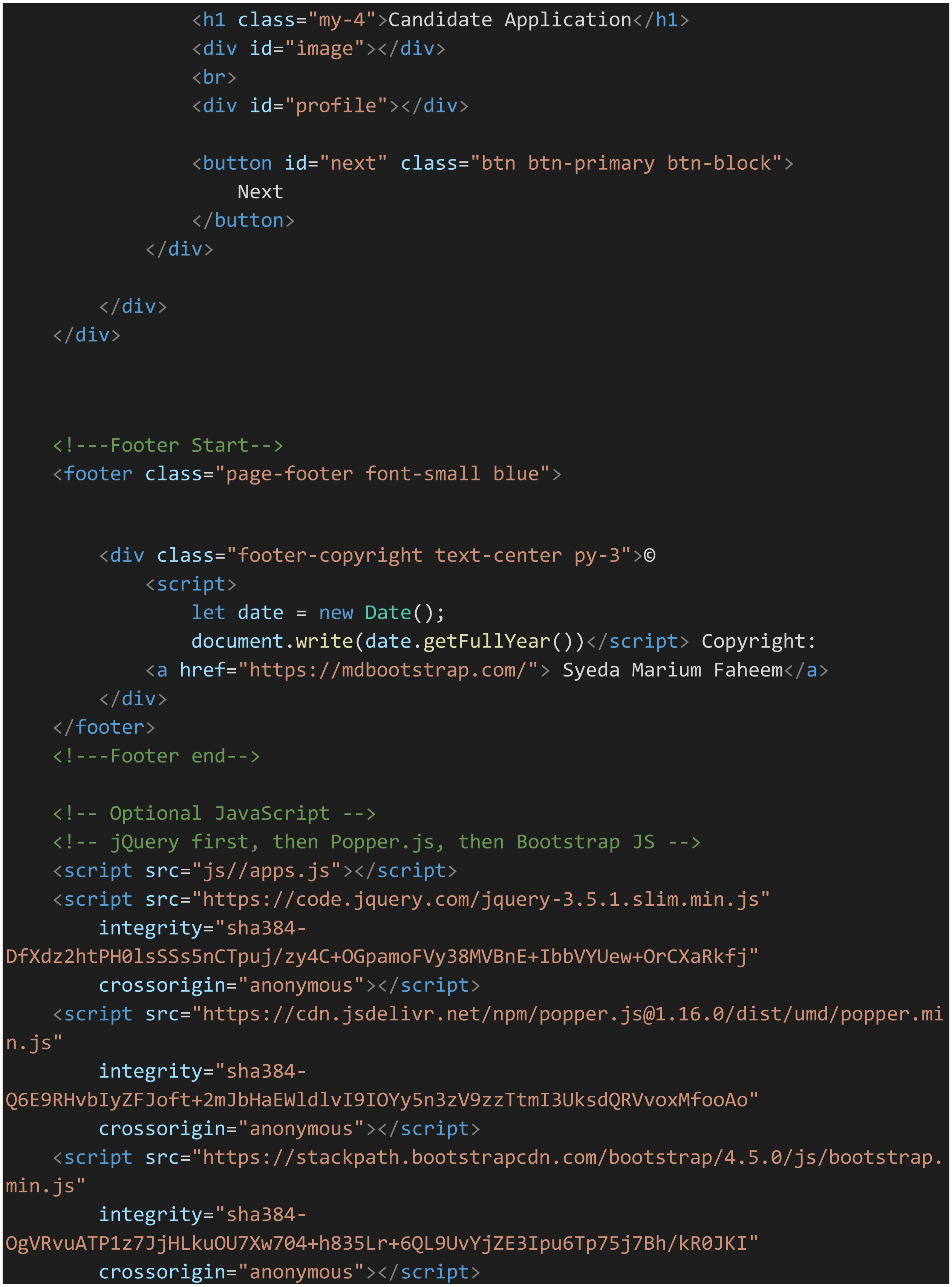
Generators :

```
//runtime pe value generate kregaa similar that we use in python
//function k bd * laga dena is syntax of generator
//yeild is 1 key workd yeild vale generat karega
function* numbersGen(){
    let i=0;
   while(true){
       yield(i++).toString();
   }}
const gen=numbersGen()
console.log(gen.next().value)
console.log(gen.next().value)
console.log(gen.next().value)
```

Creating a CV Screener :

Index.html:

```
<!doctype html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
        integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">
    <link rel="stylesheet" href="style.css">
    <title>Hello, world!</title>
</head>

<body>
    <div class="container text-center">
        <div class="row">
            <div class="col-md-6 mx-auto ">
```

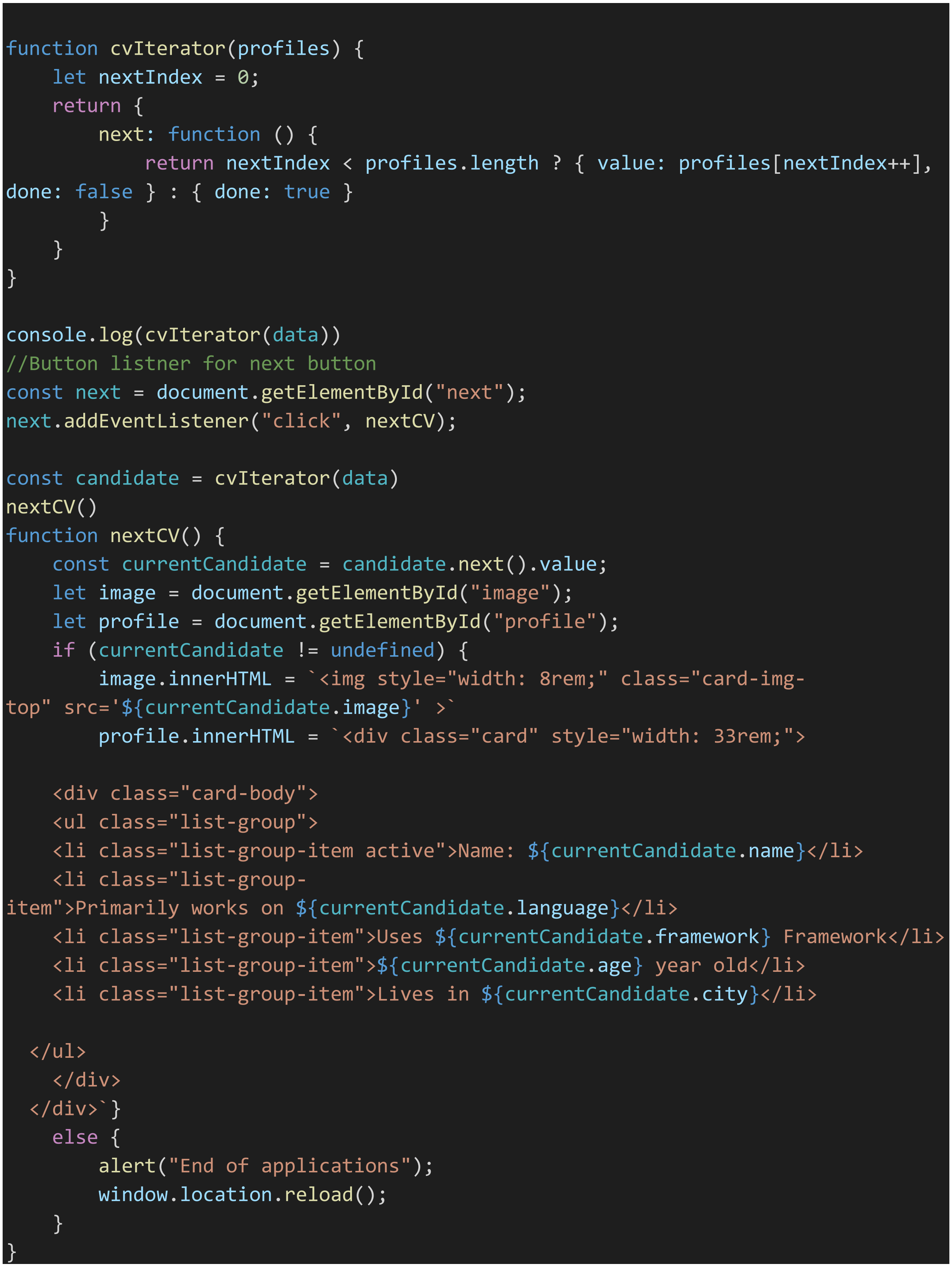


```
                <h1 class="my-4">Candidate Application</h1>
                <div id="image"></div>
                <br>
                <div id="profile"></div>

                <button id="next" class="btn btn-primary btn-block">
                    Next
                </button>
            </div>

        </div>
    </div>



    <!---Footer Start-->
    <footer class="page-footer font-small blue">


        <div class="footer-copyright text-center py-3">©
            <script>
                let date = new Date();
                document.write(date.getFullYear())</script> Copyright:
            <a href="https://mdbootstrap.com/"> Syeda Marium Faheem</a>
        </div>
    </footer>
    <!---Footer end-->

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="js//apps.js"></script>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
        integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
        crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.mi
n.js"
        integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
        crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.
min.js"
        integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
        crossorigin="anonymous"></script>
```

```
</body>

</html>
```

Js:

```
const data = [
    {
        name: "Ali Ahmed",
        age: 20,
        city: "karachi",
        language: "Python",
        framework: "Django",
        image: "https://randomuser.me/api/portraits/men/47.jpg"
    },
    {
        name: "Bakhtwar Faheem",
        age: 20,
        city: "karachi",
        language: "Javascript",
        framework: "Angular",
        image: "https://randomuser.me/api/portraits/men/45.jpg"
    },
    {
        name: "Ebad Nadeem",
        age: 18,
        city: "Islamabad",
        language: "Python",
        framework: "Flask",
        image: "https://randomuser.me/api/portraits/men/47.jpg"
    },
    {
        name: "Syeda Hareem",
        age: 20,
        city: "karachi",
        language: "C++",
        framework: "sfml",
        image: "https://randomuser.me/api/portraits/women/47.jpg"
    },]

//Cv Iterator
```



```
function cvIterator(profiles) {
    let nextIndex = 0;
    return {
        next: function () {
            return nextIndex < profiles.length ? { value: profiles[nextIndex++],
done: false } : { done: true }
        }
    }
}

console.log(cvIterator(data))
//Button listner for next button
const next = document.getElementById("next");
next.addEventListener("click", nextCV);

const candidate = cvIterator(data)
nextCV()
function nextCV() {
    const currentCandidate = candidate.next().value;
    let image = document.getElementById("image");
    let profile = document.getElementById("profile");
    if (currentCandidate != undefined) {
        image.innerHTML = `<img style="width: 8rem;" class="card-img-
top" src='${currentCandidate.image}' >`
        profile.innerHTML = `<div class="card" style="width: 33rem;">

    <div class="card-body">
    <ul class="list-group">
    <li class="list-group-item active">Name: ${currentCandidate.name}</li>
    <li class="list-group-
item">Primarily works on ${currentCandidate.language}</li>
    <li class="list-group-item">Uses ${currentCandidate.framework} Framework</li>
    <li class="list-group-item">${currentCandidate.age} year old</li>
    <li class="list-group-item">Lives in ${currentCandidate.city}</li>

  </ul>
    </div>
  </div>`}
    else {
        alert("End of applications");
        window.location.reload();
    }
}
```

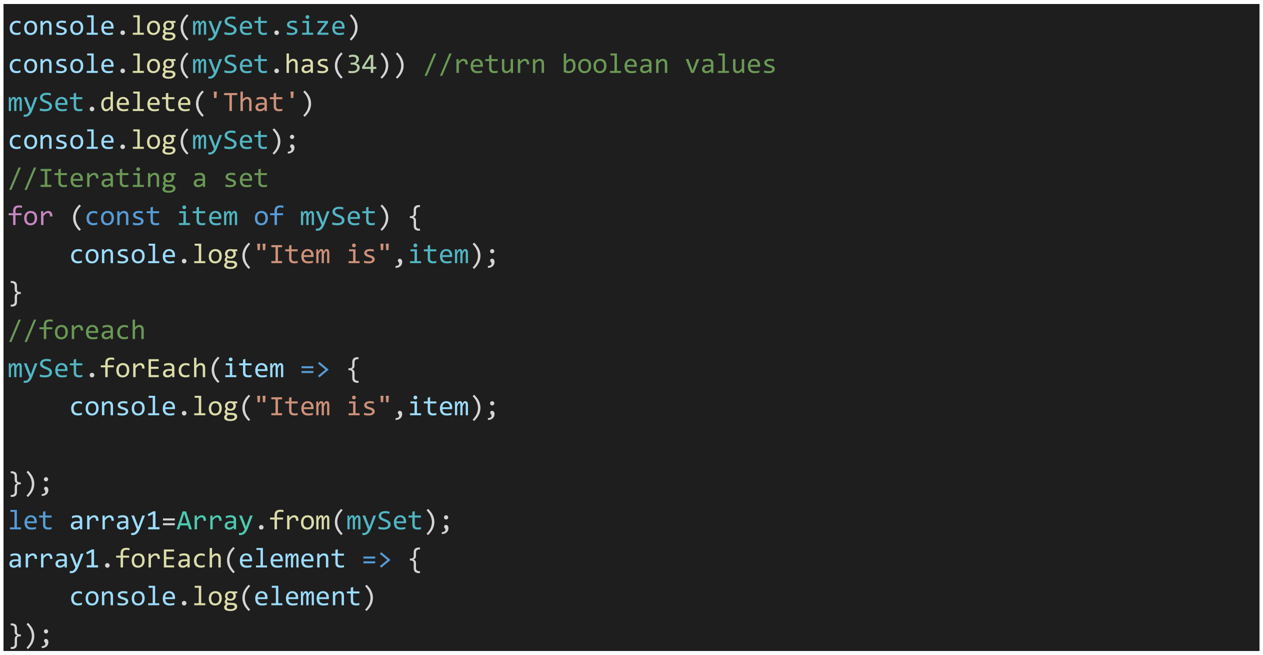
For.. of Loop vs For.. in Loop:

JavaScript Maps :

```
//we can use any key and value
let myMap=new Map();
//in this way we created an empty map
console.log(myMap);
let key1="myStr",
key2={"name":"Marium"},
key3=function(){}
//setting map values
myMap.set(key1,'This is a string');
myMap.set(key2,'This is blank obj');
myMap.set(key3,'This is an empty function');
console.log(myMap)
//Getting values from the map
let value1=myMap.get(key1);
//key1 k coreedong value
//console.log(value1)
let value2=myMap.get(key2);
//key2 k coreedong value
console.log(value2)
//getting size pf map
console.log(myMap.size)
//you can loop using for ..of to gey keys and values
for(let [key,values] of myMap){
    console.log(key,values);
}

for(let key of myMap.keys()){
    console.log("Keys are",key)
}
for(let value of myMap.values()){
    console.log("Values are",value)
}
//you can loop through a map using for each lop
myMap.forEach((value,key) => {
    console.log(`key is ${key} and value is ${value}`)
    });
//we can also convert map into array
let myArray=Array.from(myMap);
console.log("Map to array is ",myArray)
//hum sirf keys ya value ko bhi array m convert krsktey h like
let myKeysArray=Array.from(myMap.keys());
console.log(`Map to array keys is `,myKeysArray);
let myvalueArray=Array.from(myMap.values());
console.log(`Map to array values is `,myvalueArray);
```

JavaScript Sets

```
console.log("sets")
//jse map kiye thy wese hi sets h set is a collcetion of well defined values unique values store hoktry h
const mySet=new Set();
//Adding values to the set
console.log("The set looks like",mySet)
mySet.add("This");
mySet.add("My name");
mySet.add("That");
mySet.add(345);
mySet.add(true)
console.log("The set looks like",mySet)
//yai km contructor ko de kr bhi krskty thy
let set1=new Set([1,4,5,'this',false,{'a':111}]);
console.log("The set looks like",set1)
//agar koi aise app ho jaahn array se uniquw value chahye ho toh us array ko daldo set m
```



```
console.log(mySet.size)
console.log(mySet.has(34)) //return boolean values
mySet.delete('That')
console.log(mySet);
//Iterating a set
for (const item of mySet) {
    console.log("Item is",item);
}
//foreach
mySet.forEach(item => {
    console.log("Item is",item);

});
let array1=Array.from(mySet);
array1.forEach(element => {
    console.log(element)
});
```