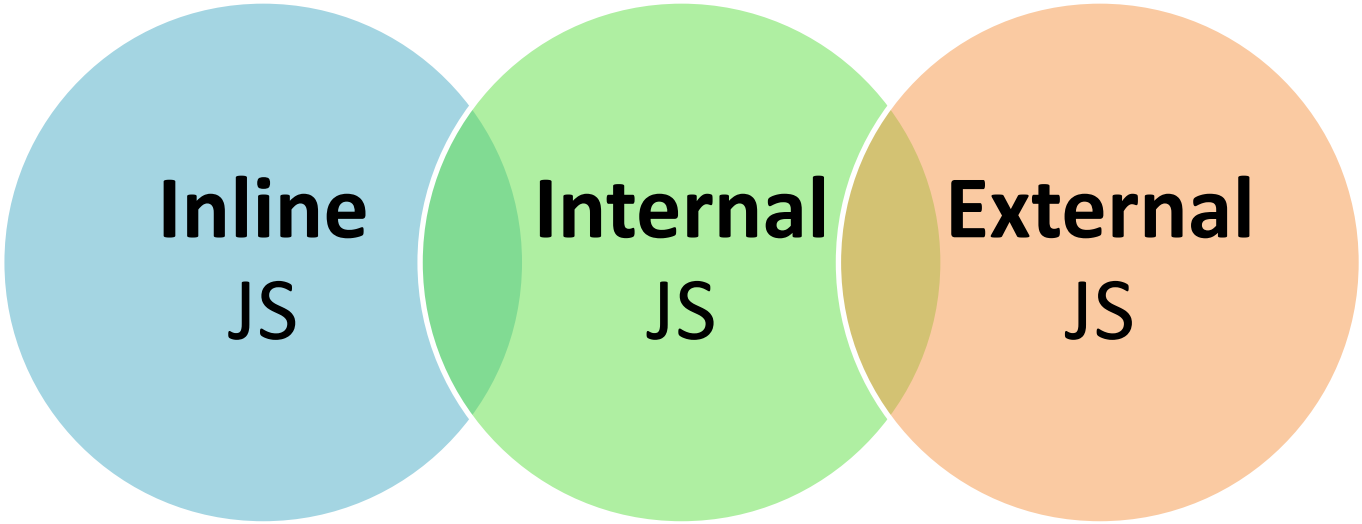
Java Script (JS)

- JS is a programming language for a web page.
- JS is created by Brendan Eich.
- “Mocha” was the first name.
- First publicly available in 1995.

Ways to use JS



JS Comment

Single line Comment:

```
// this is single line comment
```

Multi line Comment:

```
/*
This is multiple line comment
Line 2
Line 3
*/
```

Variables

Variable is just a container that contains data.

Data

Data in Variable

var name="Rahul Chauhan"; (String Data)
var id=103; (Integer or int Data)
var height=5.9; (FloatingPoint or float Data)
var isIndian=true; (Boolean Data)

Java Script is not strict typed language.

Rules for variable

It can only be Alphabets(a-z, A-Z) , Digits(0-9) , underscore(_) and dollar($).

It can not start from digit.

Keyword can not be an Identifier.

```
var data=15;
var 2data=15;
var data2=15;
var _data=15;
var var=15;
var Var=15;
var _=15;
var $=15;
var da ta=15;
var a@=15;
var _a=15;
var $a=15;
```

Change text of paragraph

If something is wrong in script tag than that script tag will not perform output, but other script tag in that page will work fine.

Printing Ways in JS

```
document.write("hi incapp");
```

```
document.getElementById("para").innerHTML="hi incapp";
```

```
window.alert("hi incapp");
```

Print the Whole Page

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

Assignment Operator (=)

```
var  a;
a = 10;
```

```
10 = a;
```

```
var  a = 10;
```

```
var  a = 10, b = 15;
```

```
var  a, b;
a = b = 10;
```

Arithmetic Operator (+ - * / %)

incapp

```
var  a;
a = 10 + 2;
```

```
var  a;
a = 10 - 2;
```

```
var  a;
a = 10 * 2;
```

```
var  a;
a = 10 / 2;
```

```
var  a;
a = 10 % 2;
```

```
var  a;
a = 10 ** 2;
```

Shorthand Operator (+= -= *= /= %=)

incapp

```
var  a=10;
a = a / 5;
```

```
var  a=10;
a /= 5;
```

```
var  a=10;
a = a + 5;
```

```
var  a=10;
a += 5;
```

```
var  a=10;
a =/ 5;
```

```
var  a=10;
a /  = 5;
```

Increment-Decrement Operator (++ --)

incapp

```
var a=10;
a = a + 1;
```

```
var a=10;
a ++;
```

```
var a=10;
a = a - 1;
```

```
var a=10;
a --;
```

Postfix

```
var a=10;
a = a + 1;
```

```
var a=10;
++ a;
```

```
var a=10;
a = a - 1;
```

```
var a=10;
-- a;
```

Prefix

```
10 ++ ;
```

```
var a=10;
a + + ;
```

Relational Operator (< <= > >= == !=)

```
var  a=10,b=5;
var c=a<b;
```

```
var  a=10,b=5;
var c=a==b;
```

```
var  a=10,b=5;
var c=a< =b;
```

```
var  a=10,b=5;
var c=a<=b;
```

```
var  a=10,b=5;
var c=a!=b;
```

```
var  a=10,b=5;
var c=a>b;
```

```
var  a=10,b=5;
var c=a>=b;
```

```
var  a="10",b=10;
var c=a==b; //true
```

```
var  a="10",b=10;
var c=a===b; //false
```

Logical Operator (&& || !)

incapp

true && true	true
true && false	false
false && true	false
false && false	false
true || true	true
true || false	true
false || true	true
false || false	false
! true	false
! false	true

```
var  a=10,b=5;
var c=a>b && b<3;
```

```
var  a=10,b=5;
var c=a<b && b<3;
```

```
var  a=10,b=5;
var c=a<b || b<3;
```

```
var  a=10,b=5;
var c=a>b || b==3;
```

```
var  a=10,b=5;
var c=!(a>b) || !(b==3);
```

Decision Making

incapp

- Decision Making is used to execute a block of code on condition, if the condition is satisfied then code will execute otherwise not.
- Types of decision making:

if	If else	else if	switch

if statement

```
if (expression) {
    // statements
}
```

Expression -> **true**

```
var age=25;
if (age>18) {
  // statements
}
// statements after if
```

Expression -> **false**

```
var age=15;
if (age>18) {
  // statements
}
// statements after if
```

If-else statement

incapp

Expression -> **true**

```
var age=25;
if (age>18) {
  // statements of if
}
else {
  // statements of else
}
// statements after if else
```

Expression -> **false**

```
var age=15;
if (age>18) {
  // statements of if
}
else {
  // statements of else
}
// statements after if else
```

If-else-if statement

```
var age=15;
if (age>60) {
   // statements of if
}
else if (age>18) {
   // statements of else  if
}
else {
  // statements of else
}
// statements after if else if
```

Switch statement

- we can use the switch statement as a substitute for long if-else-if.

```
switch (value/expression) {
  case value1 :
    // statements of case
    break;
  case value2 :
    // statements of case
    break;
  default :
    // statements of default
}
// statements after switch
```

Loop Controls

incapp

- Loop is used to execute a block of code for multiple times.
- Write the code once and execute it many times.
- Types of loop:

for **while** **do while**

For Loop

incapp

Without Loop

```
var data="";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
```

With Loop

```
var data="";
for ( var a=1 ; a<=10 ; a++ )
{
   data += "Rahul";
}
```

```
for ( Initialization ; TestExpression ; Update )
{
    // statements
}
```

While Loop

incapp

Without Loop

```
var data="";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
```

With Loop

```
var data="";
var a=1;
while ( a<=10 ) {
   data += "Rahul";
   a++;
}
```

```
while ( TestExpression) {
    // statements
}
```

Do-While Loop

incapp

Without Loop

```
var data="";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
data += "Rahul";
```

With Loop

```
var data="";
var a=1;
do {
   data += "Rahul";
   a++;
} while ( a<=10 );
```

```
do {
    // statements
} while ( TestExpression) ;
```

Important Properties/Methods of window object

incapp

Properties/Methods	Description
alert(“message”)	Display a dialog box with message you provide.
confirm(“message”)	Display a confirm dialog box with message you provide.
prompt(“message”)	Display a dialog box asking the user for input.
open(“url”)	Open new window with the URL you provided.
close()	Close the current window.
setTimeout(function, time);	Execute the function after the specified time.
setInterval(function, time);	Execute the function in loop after the specified time.

Important Properties/Methods of Math object

incapp

Properties/Methods	Description
Math.PI	Returns the value of PI.
Math.SQRT2	Returns the square root of 2.
Math.SQRT1_2	Returns the square root of 1/2.
Math.max(num1, num2,…)	Returns the highest value in the list.
Math.min(num1, num2,…)	Returns the lowest value in the list.
Math.pow(num1, num2)	Returns the value of num1 to the power of num2
Math.random()	Returns a random number in between 0.0 to 1.0.
Math.sqrt(number)	Returns the square root of the number.
Math.round(number)	Returns the nearest.
Math.ceil(number)	Returns the round up to nearest. Ex: 4.2 -> 5
Math.floor(number)	Returns the round down to nearest. Ex: 4.2 ->4

Array (**collection of data**)

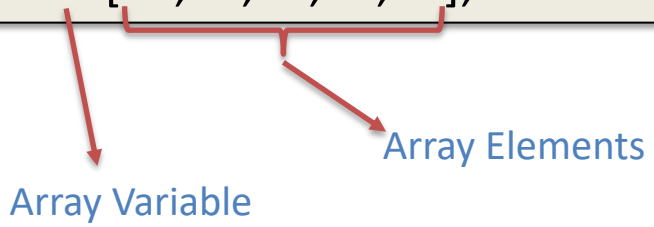
incapp

Without Array

```
var m1=89;
var m2=68;
var m3=98;
var m4=78;
var m5=45;
```

With Array

```
var m=[89,68,98,78,45];
```



Accessing Array:

```
var sum=m[0]+m[1] +m[2] +m[3] +m[4];
```

Accessing Array using loop:

```
var sum=0;
for(var x=0; x<m.length ; x++){
   sum += m[x];
}
```

Note:

Array indexing starts from 0.

'array.length' property return the number of elements in array.

Important Methods of Array object

incapp

Properties/Methods	Description
join(“characters”)	Joins all elements into a string with characters in between.
pop()	Removes the last element from an array.
push(“data")	Adds a new element to an array (at the end).
shift()	Removes the first array element and "shifts" all other elements to a lower index. And returns the value that was "shifted out"
unshift()	Adds a new element to an array (at the beginning), and "unshifts" older elements. And returns the new array length.
array1.concat(array2,…)	Creates a new array by merging (concatenating) existing arrays.
sort()	Sorts an array in increasing order.
reverse()	Reverses the elements in an array.

String

String is the collection of characters in double quotes.

```
var name="Rahul Chauhan";
```

Properties/Methods	Description
length()	Returns the length of a string.
charAt(index)	Returns the character at specified index number.
concat(data)	Returns the String by concatenating the data.
indexOf(data)	Returns the index number of specified data.
lastIndexOf(data)	Returns the index number of specified data from the last.
replace(data1, data2)	Returns the String by replacing data1 to data2.
substr(index, number)	Returns the String of specified number of character from specified index.
substring(start-index, end-index)	Returns the String in between the specified indexes. End-index is excluded.

String

incapp

Properties/Methods	Description
toLowerCase()	Returns the String in small letters.
toUpperCase()	Returns the String in capital letters.
trim()	Returns the String by removing spaces from start and end.
split("data")	Returns the array of Strings by splitting according to specified data.
padStart(targetLength, padString)	Returns the String by pad a string with another string until it reaches the given length. The padding is applied from the left end of the string. Ex: "Hi".padStart(4,0); // result is 00Hi
padEnd(targetLength, padString)	Returns the String by pad a string with another string until it reaches the given length. The padding is applied from the right end of the string. Ex: "Hi".padEnd(4,0); // result is Hi00

Change CSS using JS

To change the style of an element, use this syntax:

```
document.getElementById(id).style.propertyName =  'propertyValue' ;
```

To add the class to an element, use this syntax:

```
document.getElementById(id).className = 'name of class' ;
```

Date and Time

incapp

Get current date and time:	var d = new Date();
Get specific date and time:	var d = new Date(*year, month, day, hours, minutes, seconds, milliseconds*); Note: all parameters are not compulsory.
Get full year:	var y = new Date().getFullYear();
Get month:	var m = new Date(). getMonth();
Get Date:	var d = new Date().getDate();
Get hour:	var h = new Date(). getHours();
Get minutes:	var m = new Date().getMinutes();
Get seconds:	var s = new Date(). getSeconds();
Get day:	var d = new Date().getDay();

HTML Events

HTML event means something is happening to HTML element. Like, click, focus, mouseup, etc
In JavaScript, we can handle those events.

List of Important Events:
onClick
onMouseOver
onMouseUp / onMouseDown
onFocus / onBlur
onKeyUp / onKeyDown
onLoad
onResize

Note: HTML event names are NOT case sensitive.

GetElement ways

We can get element in different ways:

List of getElement methods:

var element = document.**getElementById**(“id”);

var elements =document.**getElementsByTagName**(“tag name”);
This method return array of elements having same tag name.

var elements =document.**getElementsByClassName**(“class name”);
This method return array of elements having same class.