Date.....

Javascript

- Javascript is a very powerful client-side Scripting language.
- Javascript is used mainly for enhancing the interaction of a user with the webpage.
- Javascript offers lots of flexibility.
- Mobile app development, desktop app development and game development.

What is Javascript Used for?

Javascript is used in various fields from the web to Servers:

- Web Applications
- Mobile Applications
- Web-based Games
- Back-end Web Development

Features of Java Script

- Light Weighted
- Case Sensitive
- Control Statements
- in-built Functions
- looping Statements
- Client Side Technology
- Object based Scripting language
- Validating User's Input
- if...else Statement
- Scripting language
- Interpreter Based
- Event handling

Date.....

Application of Javascript

Javascript is used to create interactive websites. It is mainly used for:

- Client-side validation,
- Dynamic drop-down menus,
- Displaying date & time,
- Displaying pop-up windows & dialog boxes
- Displaying clocks etc.

Example

```
<Script type = "text/javascript" >
document.write ("Javascript is a simple language for javatpoint learners");
</Script>
```

Javascript provides 3 places to put the Javascript Code:

- Within body tag
- Within head tag and
- external Javascript file

] Inpage

Javascript Syntax

- Within body tag
- Within head tag
- external Javascript file

```
<html>
<head>
<Script></Script>
</head>
OR <body>
<Script></Script>
</body>
</html>
```

<Script></Script>

Sbiral

Date.....

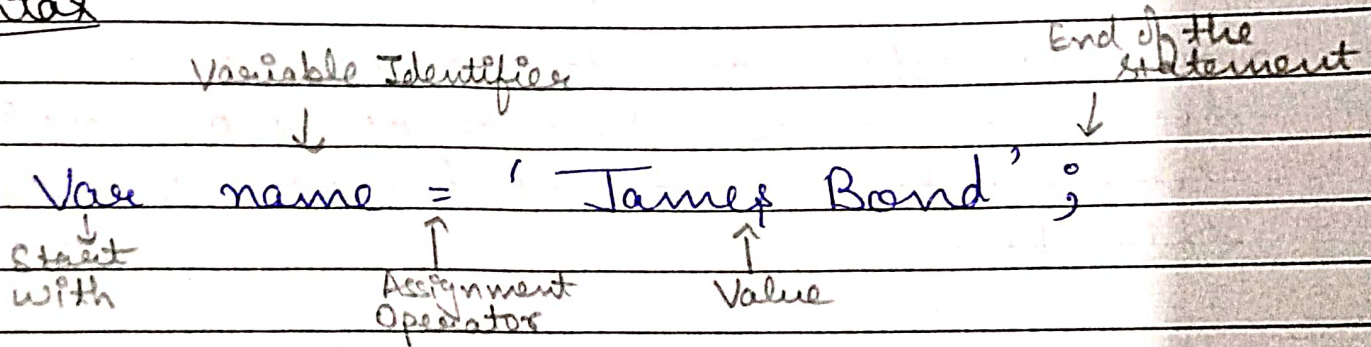
What is Variable?

Variables are used to store data, like String of text, numbers, etc. The data or value stored in the variables can be set, updated, and retrieved whenever needed.

Example

```
Var name = "Peter Parker";
Var age = 21;
Var isMarried = false;
```

Syntax



Rules for Declaring a Variable

- A variable name must start with a letter, underscore (_), or dollar sign ($).
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters (A-z, 0-9) and underscores.
- A variable name cannot contain spaces.

Date.....

- A variable name cannot be a JavaScript keyword or a JavaScript reserved word.

Javascript Comments

Javascript comments are meaningful way to deliver the message. It is used to add information about the code, warnings or suggestions so that end user can easily interpret the code.

The Js comments are ignored by the Js engine which is already embedded in the browser.

By these comments we have two advantages:-

1. To make code easy to understand.
2. To avoid the unnecessary code.

Types of Comments

1. Single line Comment
2. Multi line Comment

1. Single line Comment => It is represented by double forward slashes (//). It can be used before & after the Statement.

Example:-
```
<Script>
// It is Single line comment
document.write "Hello Javascript"
</Script>
```

Date.....

By the above example we added comment before the statement.

Example :-
```
<Script>
var a = 10;
var b = 20;
var c = a+b; // We are adding var a & b
document.write(c); // Here we are printing
                      the c value
</Script>
```

By this example a comment is added after the statement.

2. Multi line Comment => It can be used to add a single line as well as multiline comments. It is represented by forward slash with astrick Symbol (*) and comment is end with astrick Symbol with forward slash.

Example :-
```
<Script>
/* It is a single line representation of
   multiline comment */
document.write("Hello it is a representation
                of multiline comment");
/* by this Eg we can
   Understand the multiline comments
   representation in a single line & also in
   multiline */
</Script>
```

Identifier
var a = 5 ; → Terminator
keyword Assign. Value, Datatype

Date.....

Javascript Datatype

Javascript provides different data types to hold different types of values. There are two types of data types in Javascript :-

1. Primitive Data Type
2. Non-primitive Data Type (Reference)

Javascript is a dynamic type language means we need not to specify type of the variable because it is dynamically used by Js engine. We need to use Var here to specify the data type. It can hold any type of values such as numbers, Strings etc. that means dynamically type.

Example
```
var a = 20; // number
var b = "Drishti"; // String
```

1 Primitive Data Type

There are five types of Primitive data types are available.

	Data Type	Description
1	Number	which represents numeric values. Eg = 200

Date.....

2 String — Which represents sequence of characters. Eg "Drishti"

3 Boolean — Which represents boolean value either True / False

4 Undefined — Undefined is the default value of a variable that has not been assigned any value.

5 Null — A null value denotes an absence of value i.e. no value at all.

```
var ster = null;
```

Symbol

2 Non - Primitive Data type

1 Object — An object holds multiple values in terms of properties and methods.

Example :-

```
var person = {
    first name : "Drishti",
    last name : "Chakraverty",
    age : 22
};
```

2 Date — Date object represents date & time including days, months, years, hours,

Date.....

minutes, seconds & milliseconds.

Example: var today = new Date("24 September 2000");

3 Array

An array stores multiple values using special syntax.

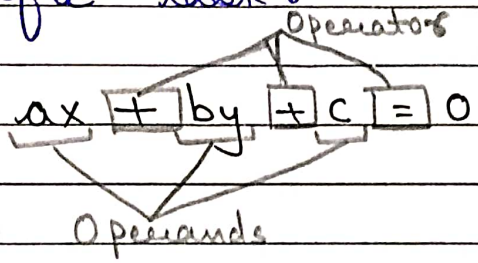
Example: var nums = [1,2,3,4];

Regular Expression

Function

Javascript Operators

It is a symbol which is going to perform a specific task.



Here '+' is an arithmetic operator & '=' is an assignment operator. There are following types of operators available in Java Script.

1 Arithmetic Operators
2 Comparison Operators (Relational)
3 Bitwise Operators
4 Logical Operators
5 Assignment Operators
6 Special Operators

Date.....

1. Arithmetic Operators -> Arithmetic Operators are used to perform arithmetic (mathematical) operations on the operands. They are given as follows :-

Operator	Description	Example
+	To perform addition	30+20 = 50
−	To perform Subtraction	30 − 20 = 10
*	To perform multiplication	3 * 2 = 6
/	To perform Division	10/2 = 5
%	To perform Modular Division	20 % 2 = 0

++	Increment	var c = 20;
		c++ ;
		d.w(c);
		Output = 21

−−	Decrement	var c = 20;
		c−− ;
		d.w(c);
		output = 19

2. Comparison Operators => A comparison operator compares two operands.

Date.....

Operator	Description	Example
==	is equal to	10 == 20 → False
===	Identical (= & of same type)	10 === 20 → False
!=	Not equal to	10 != 20 True
!==	Not identical	20 !== 20 False
>	Greater than	20 > 10 True
>=	Greater than or equal to	20 >= 10 True
<	& Less than	20 < 10 False
<=	Less than or equal to	20 <= 10 False

3. Bitwise Operators => These operators perform Bitwise operations on operands.

Operator	Description	Example
&	Bitwise &	(10 == 20 & 20 == 33) False
|	Bitwise OR	(10 == 20 | 20 == 33) False
^	Bitwise XOR	(10 == 20 ^ 20 == 33) False
~	Bitwise NOT	(~10) = -10

Date.....

<<	Bitwise left Shift	(10<<2) = 40
>>	Bitwise Right Shift	(10>>2) = 2
>>>	Bitwise right Shift with 0	(10>>>2) = 2

4 Logical Operators

Operator	Description	Example
&&	Logical AND	(10==20 && 20==23) False
||	Logical OR	(10==20 || 20==23) False
!	Logical NOT	!(10==20) True

5 Assignment Operators

Operator	Description	Example
=	Assign	10+10 = 20
+=	Add & assign	var a = 10; a += 20; a = 30
-=	Subtract & assign	var a = 20; a -= 10; a = 10

Date.....

*=	Multiply & assign	var a = 10;
		a* = 20;
		a = 200

/=	Divide & assign	var a = 10;
		a /= 2;
		a = 5

%=	Modulus & assign	var a = 10;
		a %= 2;
		a = 0

6 Special Operators

Operator	Description
? :	It is a conditional Operator, returns value based on the condition. It is like a if else.
,	Comma operator allows multiple expression to be evaluated as single statement. It is used to divide the two statements.
Delete	Delete operator deletes the property of the object.
IN	IN operator checks if object has the given property.

Date.....

Instance of	which checks if the object is an instance of given type.
New	To create an object (instance)
Type Of	Which checks the type of object
Void	It discards the expression return value.
Yield	Which checks what is returned in a generator by the generators iterator.

Conditional Control Structure

Allows a group of statements from multiple group of statements.

1. If() (Decision Maker) ⇒ When you want to execute a group of one or more Script Statements only when a particular condition is met.

OR

It evaluates the content only if expression is true.

Syntax :-

```
if(expression)
{
// content to be evaluated
}
```

Date.....

Example :-

```
<Script>
var a = 20;
if (a > 10) {
document.write ("Value of a is greater than 10");
}
</Script>
```

2. If else() => It evaluates the content whether condition is true or false.

Syntax :-

```
if (expression) {
// content to be evaluated if condition is true
}
else {
// content to be evaluated if condition is false
}
```

Example :-

```
<Script>
var a = 20;
if (a % 2 == 0) {
document.write("a is even number"); }
else {
document.write ("a is odd number"); }
</Script>
```

Date.....

3. if else if() => It evaluates the content only if expression is true from several expressions.

Syntax :-

```
if(expression 1) {
//Content to be evaluated if expression 1 is true}
else if (expression 2)
{//Content to be evaluated if expression 2 is true}
elseif (expression 3)
{ //Content to be evaluated if expression 3 is
  true }
else {//Content to be evaluated if no
      expression is true }
```

Example :-

```
<Script>
var a = 20;
if (a == 10) {
document.write ("a is equal to 10");
}
elseif (a == 15) {
document.write ("a is equal to 15");
}
elseif (a == 20) {
document.write ("a is equal to 20");
}
else {
document.write ("a is not equal to 10, 15 or
                 20");
}
</Script>
```

Date.....

4. Nested if() => Ladder if

Syntax :-

```
if (condition)
{
Statement 1;
}
else if (condition)
{
Statement 2;
}
else if (condition)
{
Statement 3;
}
else
{
Statement 4;
}
```

Switch Statement

The JavaScript Switch Statement is used to execute one code from multiple expressions.

Syntax :-

```
switch (expression) {
case value 1:
Code to be executed;
break;
case value 2;
Code to be executed;
break;
- - - -
default;
```

Date.....

Example:-

```
<Script>
var grade = 'B';
var result;
Switch (grade) {
case 'A':
result += "A Grade";
Case 'B':
result += "B Grade";
Case 'C'
result += "C Grade";
default:
result += "No Grade";
}
document.write (result);
</Script>
```

Loop Control Statement

The Java Script loops are used to iterate the piece of code using for, while, do while or for-in loops.

There are four types of loops in JavaScript.

(i) For Loop
(ii) While Loop
(iii) do-while loop
(iv) for-in Loop

Date.....

(i) For() Loop => The For loop allows you to execute a block of statements for a predefined number of times.

Syntax :-

```
for (initialization; condition;
    Increment/Decrement)
{
Code to be executed
}
```

(ii) While () Loop => The JavaScript while loop iterates the elements for the infinite number of times. It should be used if number of iteration is not known.

Syntax :-

```
Initialization
while (Condition)
{
Code to be executed
    Increment / Decrement
}
```

(iii) do while() loop => do while loop iterates the elements for the infinite number of times like while loop. but, code is executed at least once whether condition is true or false.

Date.....

Syntax :- do {
code to be executed
}
while (condition);

(iv) for-in() Loop => The for...in loop provides a simpler way to iterate through the properties of an object.

Syntax :-
for (VariableName in Object)
{
Statement (s)
} (condition);

JavaScript Array

JavaScript array is an object that represents a collection of similar type of elements.

There are 3 ways to construct array in JavaScript :-

(i) By array literal

(ii) By creating instance of Array directly (using new keyword)

(iii) By using an Array constructor (using new keyword)

Date.....

(i) JavaScript Array Literal

Syntax :-

```
Var arrayname = [value1, value2 ..... valuen];
```

We can see values are contained inside [and separated by , (comma)

(ii) JavaScript Array Directly (new keyword)

We can create an object by using new keyword with homogeneous data elements.

Syntax :-

```
Var arrayname = new Array();
```

(iii) JavaScript Array Constructor (new keyword)

Here, we need to create instance of array by passing arguments in constructor so that we don't have to provide value explicitly.

Example :-

```
<Script>
    Var emp = new Array("Drishti", "Pratibha",
                        "Anshika");
    for (i=0; i<emp.length; i++) {
    document.write(emp[i] + "<br>");
    }
</Script>
```

Date.....

Java Script Array Methods

Methods	Description
1 Concat()	It returns a new array object that contains two or more merged arrays.
2 Copy with in()	It copies the part of the given array with its own elements & returns the modified array.
3 Every()	It determines whether all the elements of an array are satisfying the provided function conditions.
4 Fill()	It fills elements into an array with static values.
5 Filter()	It returns the new Array containing the elements that pass the provided function conditions.
6 Find()	It returns the value of the first elements in the given array that satisfies the specified condition.

Date.....

7	Find Index()	It returns the index value of the first element in the given array that satisfies the specified condition.
8	For Each()	It invokes the provided function once for each element of an Array.
9	Includes()	It checks whether the given array contains the specified elements.
10	Index Of()	It searches the specified element in the given array & returns the index of the first match.
11	Join()	It joins the elements of an array as a string.
12	Last Index Of()	It searches the specified elements in the given array & returns the index of the last match.
13	Map()	It calls the specified function for every array element and returns the new Array.

Date.....

14	Pop()	It removes & returns the last elements of an array.
15	Push()	It adds one or more elements to the end of an Array.
16	Reverse()	It reverses the elements of given array.
17	Shift()	It removes & returns the first element of an array.
18	Slice()	It returns a new array containing the copy of the given array.
19	Sort()	It returns the elements of the given array in a sorted order.
20	Splice()	It add / remove elements to / from the given array.
21	Unshift()	It adds one or more elements in the beginning of the given array.

Date.....

JavaScript Objects

- A JavaScript object is an entity having State & behaviour. For example: car, pen, bike, chair, glass, keyboard, monitor etc.
- JavaScript is an object-based language. Everything is an object in JavaScript.
- JavaScript is templated based not class based.

Creating Objects in JavaScript

There are 3 ways to create objects.

(i) By object literal

(ii) By creating instance of Object directly (using new keyword)

(iii) By using an object constructor (using this keyword)

(i) JavaScript object by Object literal

Syntax :-

```
Object = { property 1 : value 1, property 2 : value 2 ..... property n : value n }
```

In this, property & the values are separated by :(colon).

Date.....

Example:-

```
<Script>
emp = {id: 1, Name: "Drishti", Salary: 50000}
document.write(emp.id + " " + emp.name + " "
                + emp.Salary)
</Script>
```

(ii) By creating Instance of the object

Syntax:-

```
Var object name = New object ();
```

Here, we are using New keyword to create the object.

Example:-

```
<Script>
var emp = New object ();
emp.id = 1;
emp.name = "Drishti";
emp.Salary = 5000;
document.write(emp.id + " " + emp.name +
                " " + emp.Salary);
</Script>
```

(iii) By using an Object Constructor

Here, you need to create function with argument.

Each argument value can be assigned in the current object by using this keyword.

Shiral

Date.....

This keyword refers to the current object.

Creating object constructor is given below :-

Example :-

```
<Script>
function emp (id, name, Salary)
{
this.id = id;
this.name = name;
this.Salary = Salary;
}
</Script>
```

Javascript Function

A function is a block of code that performs a specific task.

Function is used to perform operations. We can call Java Script function many time to reuse the code.

Advantages :-

1. Code Reusability => We can call a function several times. So it save Coding.

2. Less Coding => It makes our program concept. We don't need to write many lines of

Date.....

Code each time to perform a common task.

Declaring a Function

- A function is declared using the function keyword.
- The basic rules of naming a function are similar to naming a variable.
- The body of function is written within { }.

Syntax :-

```
function nameOfFunction() {
    // function body
}
```

Calling a Function

- Example :-

```
function greet() {
    // code
}

greet();

// code
```

Date.....

Function Parameters

A function can also be declared with parameters. A parameter is a value that is passed when declaring a function.

Example

```
function greet (name) {
    // Code
}

greet (name);

// Code
```

Function Return

- The return Statement can be used to return the value to a function call.
- The return Statement denotes that the function has ended. Any code after return is not executed.
- If nothing is returned, the function returns an undefined value.

Example

```
function add (num1, num2) {
    // Code
    return result;
}
let x = add (a, b);

// Code
```

Date.....

Benefits of Using a function

- Function makes the code reusable. You can declare it once & use it multiple times.
- Function makes the program easier as each small task is divided into a function.
- Function increases readability.

Function Expressions

In JavaScript, functions can also be defined as expressions.

```
// program to find the Square of a number
let x = function (num) { return num * num };
console.log(x(4));

// can be used as variable value for other variables
let y = x(3);
console.log(y);
```

Output

```
16
9
```

Date.....

Javascript Programs

- Inpage Javascript Example

```
<Script>
document.write("Hello World !!");
</Script>
```

- External Javascript Example

```
<body>
<Script src = "external.js">
</Script>
</body>
```

- Variable Declaration

```
<Script>
let fname = "Drishti";
var fname = "Chakravarty";

document.write (fname);
</Script>
```

- Javascript Datatype (null)

```
<Script>
var x = ["apple", "banana"];
document.write (x + "<br><br>");
document.write (typeof x);
</Script>
```

Date.....

Javascript Datatype

```
<Script>

var fname = "Drishti Chakraverty"
var fname = "Drishti Chakraverty"

document.write(fname + "<br><br>");
document.write(typeof fname + "<br><br>");

var number = 300;

document.write(number + "<br><br>");
document.write(typeof number + "<br><br>");

var boolean = False;
var boolean = True;

document.write(boolean + "<br><br>");
document.write(boolean + "<br><br>");
document.write(typeof boolean);

</Script>
```

Javascript String

```
<body>
<p id = "p1"> </p>
<p id = "p2"> </p>
<p id = "p3"> </p>
<Script>
```

Date.....

```
let str1 = "This is double quoted String";
let str2 = 'This is single quoted String';
let str3 = `This is backtick String`;

document.getElementById("p1").innerHTML = str1;
document.getElementById("p2").innerHTML = str2;
document.getElementById("p3").innerHTML = str3;
</Script>
```

Javascript String 2

```
<body>

<p id = "p1"> </p>
<p id = "p2"> </p>
<p id = "p3"> </p>
<p id = "p4"> </p>

<Script>

let str1 = "Javascript";
let ch1 = str[0]        // J

document.getElementById("p1").innerHTML = ch1;

</Script>
</body>
```

Output :- J

Date.....

Javascript String

```
<body>

<p id = "p1"> </p>
<p id = "p2"> </p>
<p id = "p3"> </p>
<p id = "p4"> </p>
<p id = "p5"> </p>
<p id = "p6"> </p>
<p id = "p7"> </p>

<Script>

let str = "Drishti";

let ch1 = str [0]  // D
let ch2 = str [1]  // r
let ch3 = str [2]  // i
let ch4 = str [3]  // s
let ch5 = str [4]  // h
let ch6 = str [5]  // t
let ch7 = str [6]  // i
let ch7 = str [7]

document.getElementById("p1").innerHTML = ch1;
document.getElementById("p2").innerHTML = ch2;
document.getElementById("p3").innerHTML = ch3;
document.getElementById("p4").innerHTML = ch4;
document.getElementById("p5").innerHTML = ch5;
document.getElementById("p6").innerHTML = ch6;
document.getElementById("p7").innerHTML = ch7;

</Script>
</body>
```

Date.....

Javascript String 3

```
<body>
<p id = "p1"></p>
<p id = "p2"></p>
<p id = "p3"></p>

<Script>
    let Str1= "This is 'String 1' in Java";
    let Str 2 = "This is "String2" in Java';
    let Str 3 = 'This is 'String1' in Java';

document.getElementById("p1").innerHTML = Str 1;
document.getElementById("p2").innerHTML = Str 2;
document.getElementById("p3").innerHTML = Str 3;

</Script>
</body>
```

Javascript Concatenation

```
<body>
<p id = "p1"></p>
<p id = "p2"></p>
<Script>
let Str1 = "Edunet";
let Str 2 = 'Foundation';
let Str 3 = Str 1 + Str 2;
let Str4 = Str 1.Concat (Str 2);
document.getElementById("p1").innerHTML = Str 3;
document.getElementById("p2").innerHTML = Str 4;
</Script>
</body>
```

Date.....

Javascript String object

```
<body>
    <p id = "p1"> </p>
    <p id = "p2"> </p>
    <p id = "p3"> </p>
    <p id = "p4"> </p>
<script>

    let str1 = new String();

    //or

    let str2 = new String ("Hello World");
    let str3 = "Hello";

document.getElementById("p1").innerHTML = str1;
document.getElementById("p2").innerHTML = str2;
document.getElementById("p3").innerHTML = typeof(str1);
document.getElementById("p4").innerHTML = typeof(str3);

</script>
</body>
```

Arithmetic Operators

```
<script>
    let x = 8;
    let y = 6;

    // Addition
```

Date.....

```
Console.log ('x+y = ', x+y);
// Subtraction
Console.log ('x-y = ', x-y);
// Multiplication
Console.log ('x*y = ', x*y);
// Division
Console.log ('x/y = ', x/y);
// remainder or Modulus
Console.log ('x%y = ', x%y);
// Increment
Console.log ('x++ = ', ++x);
// Decrement
Console.log ('y-- = ', --y);
</Script>
```

Comparison Operators

```
<Script>
Const a=5, b=6, c="Javascript";
// equal to Operator
Console.log (a==5);          // true
Console.log (b==3);          // true
Console.log (c=="Javascript"); // False
```

Date.....

// Not equal to Operators

```
Const a = 3, b = "Javascript";

Console.log(a != 2);
Console.log(b != 'Javascript');
```

// Strict equal to Operators

```
Const a = 2;

Console.log(a === 2);
Console.log(a === '2');
```

// Strict not equal to Operators

```
Const a = 5;  b = "deishti";
Console.log(a !== 8);          // true
Console.log(b !== 'Drishti');  // False
```

// Greater than Operators

```
Const a = 3;
Console.log(a > 4);   // False
Console.log(a > 2);   // True
```

// Greater than or equal to Operators

```
Const a = 3;
Console.log(a >= 2);   // True
Console.log(a >= 3);   // True
Console.log(a >= 4);   // False
```

Surai

Date.....

// Less than Operators

```
Const a = 5, b = 3;

Console.log (a<2);          // False
Console.log (b<9);          // True
```

// less than or equal to Operators

```
Const a = 4, b = 6;

Console.log (a<=8);         // True
Console.log (b<=3);         // False

</Script>
```

Logical Operator

```
<body>
  <p id = "P1"> </p>
  <p id = "P2"> </p>
  <p id = "P3"> </p>
  <p id = "P4"> </p>
  <p id = "P5"> </p>

<Script>

let a = 5, b = 10;

document.getElementById("P1").innerHTML = (a<b)
                && (a<b);          // logical AND
```

Date.....

```
document.getElementById("P2").innerHTML = (a>b) || (a==b);
                                          // logical OR

document.getElementById("P3").innerHTML = (a<b) || (a==b);
                                          // logical OR

document.getElementById("P4").innerHTML = !(a<b);
                                          // logical NOT

document.getElementById("P5").innerHTML = !(a>b);
                                          // logical NOT
```

```
</script>
</body>
```

Assignment Operator

```
<body>
  <p id = "p1"></p>
  <p id = "p2"></p>
  <p id = "p3"></p>
  <p id = "p4"></p>
  <p id = "p5"></p>
  <p id = "p6"></p>

<script>

let x = 5, y = 10;

x = y;
```

Date.....

```
document.getElementById("P1").innerHTML = x;
x += 1;
document.getElementById("P2").innerHTML = x;
x -= 1;
document.getElementById("P3").innerHTML = x;
x *= 5;
document.getElementById("P4").innerHTML = x;
x /= 5;
document.getElementById("P5").innerHTML = x;
x %= 2;
document.getElementById("P6").innerHTML = x;
</script>
</body>
```

Add_User_Input

```
<script>
// Store input numbers
const num1 = parseInt(prompt('Enter first no.'));
const num2 = parseInt(prompt('Enter Second no.'));

// add two numbers
const Sum = num1 + num2
```

Date.....

// display the Sum

Console.log('The Sum of ${num1} & ${num2} is ${Sum}')

</Script>

Alert

<body>

<Script>

/*

window.alert(9+10);

*/

alert(90+10);

Console.log(10+15);

document.getElementById("P1").innerHTML = 3+5;

</Script>

</body>

Swapping

<Script>

let a = 3;

let b = 4;

[a,b] = [b, a]

Console.log('a=3', 'b=4');

Console.log('after Swaping the no.');

Console.log(a);

Spiral Console.log(b);

</Script>

Date.....

Swapping two numbers

Take input from user

```
<Script>
let a = parseInt(prompt('Enter first no.'));
let b = parseInt(prompt('Enter Second no.'));
  [a, b] = [b, a]

document.write('The first no. after Swap: ${a}');
document.write('The Second no. after Swap: ${b}');
</Script>
```

If Condition

```
<Script>

  var a = 20;
  if(a > 10);
  {
document.write("Value of a is greater than 10");
  }
</Script>
```

2 If Condition

```
<Script>
  if(1>0)
{alert("1 is greater than 0"); }
  if(1<0)
{alert("1 is less than 0"); }
</Script>
```

Date.....

Comparison between Two Salary

```
<Script>
var MySalary = 20000;
var YourSalary = 10000;

if (MySalary > YourSalary)
{
alert("My Salary is greater than Your Salary");
}
</Script>
```

To check whether a number is positive or Negative

```
<Script>
Const number = prompt("Enter a number");

if (number > 0)
{
document.write("The number is positive");
}
else
{
document.write("The number is negative");
}
</Script>
```

To check whether the number is greater or less than 18

```
<Script>
```

Date.....

```
const age = prompt ( "Enter a number : ");

    if ( age > 18)
      {
document.write ("You are eligible for Voting");
      }
      else {
document.write ("You are not eligible for Voting");
      }
</Script>
```

Write a program to check if the number is positive or Negative/zero using if else condition

```
<Script>
var num = prompt ("Enter a number : ");

    if( num > 0)
      {
document.write (" The number is positive");
      }
      else
      {
document.write (" The number is negative or zero");
      }
</Script>
```

Date.....

if else if Condition

Program to check whether a number is Negative, Positive or Zero

```
<Script>
const number = prompt ("Enter a number: ");

// Check if the number is greater than 0
    if (number > 0)
    {
document.write ("The number is positive");
    }
// Check if number is zero

    elseif (number == 0)
    {
document.write ("The number is Zero");
    }
// if number is neither greater than 0, nor zero
    else
    {
document.write ("The number is negative");
</Script>
```

Date.....

Function Parameter

```
<Script>
function greet(name) {
console.log("Hello" + name);
}
let name = prompt("Enter Your Name");
// Calling a function
greet(name);
</Script>
```

Add Two number

```
<Script>
// declaring a function
function add(a,b) {
console.log(a+b);
}
// Calling Function
add(3,6);
add(5,6);
add(8,9);
add(1,60);
</Script>
```

Date.....

Add five numbers

```
<Script>
// declaring a function
function add (a, b, c, d, e) {
document.write (a+b+c+d+e);
}
// Calling function
add (3, 6, 1, 3, 4);
</Script>
```

Arithmetic Operators using a function

```
<Script>
// declaring a function
function add (a, b) {
        Console.log (a+b);
}
function sub (a, b) {
        Console.log (a-b);
}
function mult (a, b) {
        Console.log (a* b);
}
function dive (a, b) {
        Console.log (a/b); }
```

Date.....

// calling function

```
add(4, 3);
Sub(9, 6);
mult(9, 4);
div(4, 2);
```

</Script>

Sum of two number using Javascript return function

```
<Script>
    function add(a, b) {
        return a+b;
    }
// take input from user
let number1 = ParseFloat(prompt("Enter Ist No.: "));
let number2 = ParseFloat(prompt("Enter IInd No.: "));
// Calling the function
let result = add(number1, number2);
// Display the result
console.log("The Sum is: " + result);
</Script>
```

Date.....

Function return

```
<Script>
    function add(a,b){
                return a+b;
            }
    function Sub(a,b){
                return a-b;
            }
    function mul(a,b){
                return a*b;
            }
    function div(a,b){
                return a/b;
            }
let number1 = parseInt(prompt("Enter First Number:"));
let number2 = parseInt(prompt("Enter Second Number:"));

let result1 = add(number1, number2);
let result2 = Sub(number1, number2);
let result3 = mul(number1, number2);
let result4 = div(number1, number2);

console.log("The Sum is: " + result1);
console.log("The Sub is: " + result2);
console.log("The mul is: " + result3);
console.log("The div is: " + result4);

</Script>
```

Date.....

Write a program which calls a function that perform a calculation & return the result

```
<body>
    <p id = "Js"></p>
    <p id = "Js1"></p>
    <p id = "Js2"></p>
<Script>
    function myfunction (P1, P2) {
                return P1 * P2;
                }
document.getElementById("Js").innerHTML = Myfunction(3,6);
document.getElementById("Js1").innerHTML = Myfunction(8,5);
document.getElementById("Js2").innerHTML = Myfunction(2,5);

</Script>
</body>
```

Javascript Object using Object literal

```
<Script>
    emp = {id: 101, name: "Drishti", Salary: 50000}

document.write(emp.id+" "+emp.name+" "+emp.Salary);

</Script>
```

Date.....

Javascript Object by creating instance of object

```
<Script>
    var emp = new object();

        emp.id = 101;
        emp.name = "Drishti";
        emp.Salary = 50000;

document.write(emp.id+" "+emp.name+" "+emp.
                                    Salary);
</Script>
```

Javascript Object by using an object Constructor

```
<Script>
    function emp(id, name, Salary) {

        this.id = id;
        this.name = name;
        this.Salary = Salary;
                }
e = new emp(101, "Drishti", 50000);

document.write(e.id+" "+e.name+" "+e.Salary);

</Script>
```

Date.....

For Loop

```
<Script>
    for(i=1; i<=5; i++)
    {
document.write(i+"<br/>")
    }
</Script>
```

Display a text five times using for loop

```
<Script>
    Const n=5;
// looping from i=1 to 5
    for(let i=1; i<=n; i++)
    {
document.write('Welcome to Javascript <br>');
    }
</Script>
```

Display the sum of 100 natural number using for loop.

```
<Script>
        let Sum=0;
        Const n=100;
// looping from i=1 to n
// in each iteration, i is increased by 1

    for(let i=1; i<=n; i++)
    {
```

Date.....

```
    sum += i;      // sum = sum + i (1+2+3+4+.....+100)
  }
console.log('Sum: ', sum);
</Script>
```

While loop

```
<Script>

  var i = 11;
   while (i<=15)
   {
document.write (i + "<br>");
    i++;
   }
</Script>
```

Sum of Positive Number only using Javascript While loop

```
<Script>

// If the user enters a negative number, the loop ends.
// The negative number entered is not added to the
   Sum variable

  let sum = 0;

// take the input from the user
```

Spiral

Date.....

```
let number = parseInt(prompt('Enter a Number: '));
while (number >= 0) {
    //add all the positive no.
        Sum += number;
// take input again if the number is positive.
number = parseInt(prompt('Enter a number: '));
    }
// display the Sum
document.write(Sum);
</Script>
```

Do While loop

```
<Script>
        var i = 1;
    do {
    document.write(i + "<br/>");
        i++;
        } while (i<=5);
</Script>
```

For in loop

```
<Script>
        var person = {
            first name : 'Drishti',
            last name : 'Chakarwarty',
            DOB : '24/09/2000'
            };
```

Date.....

```
for( var prop in person) {
console.log( prop + ' : ' + person [prop] );
```

Break Statement

```
<Script>
    for( let i=1; i<=5; i++) {

        // break condition
        if(i==3) {
            break;
        };
        console.log (i);
    }
</Script>
```

Continue Statement

```
<Script>
for( let i=1; i<=5; i++) {
        // continue condition
        if(i==3) {
            continue;
        };
        console.log (i);
    }
</Script>
```

Date.....

Switch Case

```
<Script>
    let month = 1;
    let monthname;
Switch(month) {

    Case 1:
        month name = "January";
        break;
    Case 2:
        month name = "Febeuary";
        break;
    Case 3:
        month name = "March";
        break;
    Case 4:
        month name = "April";
        break;
        .
        .
        .
    Case 12:
        month name = "December";
        break;
    default:
        month name = "Invalid month Number";
        }
    alert(month name);
</Script>
```

Date.....

Javascript Array

```
<body>
<p id = "p1"></p>
<p id = "p2"></p>
<p id = "p3"></p>
<p id = "p4"></p>
<p id = "p5"></p>

<Script>

let StringArray = ["one", "two"];
let numericArray = [1, 2, 3];
let decimalArray = [1.1, 1.2, 1.3];
let booleanArray = [True, False];
let dataArray = [1, "Drishti", "DC", True, 50000, 5.5];

document.getElementById("p1").innerHTML = StringArray;
document.getElementById("p2").innerHTML = numericArray;
document.getElementById("p3").innerHTML = decimalArray;
document.getElementById("p4").innerHTML = booleanArray;
document.getElementById("p5").innerHTML = dataArray;

</Script>
</body>
```

Date.....

Write a Table using Js by taking ~~user~~ input from user.

```
<Script>
Var a = parseInt(prompt('Enter a number: '));
for(let i = 1; i<=10; i++) {

    Const result = i*a;
document.write('${a} * ${i} = ${result}'+"<br/>");
}
</Script>
```

Date : / /
Page No. 1

Javascript Array

0 1 2 3 4 → Index Values

```
var ary = [10, 20, 30, 40, 50];

document.write (ary [3]);
```

Output => 40

If we want to print every array in different line

```
var ary = [10, 20, 30, 40, 50];

    for ( var a=0; a<=4; a++) { 

document.write (ary[a] + "<br>");
    }
```

Output => 10
20
30
40
50

If we want to do the Sum of every array value

Date : / /
Page No.

```
<Script>

var arry = [10, 20, 30, 40, 50];
    var Sum = 0
    for ( a=0, a<=4, a++) {

document.write ( arry[a] )

    Sum = Sum + arry[a];

    }

    document.write ("Total Sum: " +Sum);

</Script>
```

We can also print different ~~type~~ data type values in a Single array.

```
<Script>

    var arry = [10, "Drishti", "Chakravarty", true, null];
    for(a=0, a<=4, a++) {
    document.write (arry[a] + "<br>");
    }
```

Another way to create Array

Constructor Method

```
var arr = new Array();

    arr[0] = 10;
    arr[1] = "Drishti";
    arr[3] = true;

for( var a=0; a<5; a++) {

document.write( arr[a] + "<br>");
```

To Take the input from the user

```
var arr = new Array (3);

    for( var g = 0; g<3; g++) {
      arr[g] = prompt("Enter the Value : ");
    }
document.write ( arr[a]
  for (var a = 0; a<3; a++) {
    document.write ( arr[a] + "<br>");
    }
```

Date : / /
Page No.

Multidimensional Array

```
var ary = [
          0          1          2          3
0 [ "Harry", 18, "Male", "B.Com" ],
1 [ "Sunny", 19, "Male", "BCA" ],
2 [ "Sarah", 18, "Female", "BCA" ],
3 [ "Tom", 17, "Male", "BA" ]
  ];
for(var a=0; a<4; a++) {
 document.write(ary[a] + "<br>");
   }
```

(Outer Loop)

(Inner Loop)

```
for(var a=0; a<4; a++) {
  for(var b=0; b<4; b++) {
  document.write(ary[a][b]);
    }
  document.write("<br>");
   }
```

To print Single value

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

To print the data in a table

```
document.write("<table border='1px' cellspacing='0'>");
    for (var a=0; a<4; a++) {        // arr.length
document.write("<tr>");
    for (var b=0; b<4; b++) {
document.write("<td>" + arr[a][b] + "</td>");
    }
document.write("</tr>");
    }
document.write("</table>");
```

To find the length

```
document.write(arr.length);
```

Modify Array Element

```
var a = ["Drishti", 21, "ADIT"];
    a[1] = 22;

document.write(a);
```

Remove Array Element

```
delete a[1];
```

Date : / /
Page No.

Javascript Array Methods

Sort()

```
var a = ["Drishti", "Pratibha", "Anshika", "kumkum"];
a.sort();
document.write(a + "<br>");
```

Reverse()

```
a.reverse();
doc.write(a + "<br>");
```

pop() ⇒ Delete last value
push() ⇒ Add ~~last~~ new Value at last

```
a.pop();
doc.write(a);
```

newValue

```
a.push();
doc.write(a);
```

Date : / /
Page No. 4

Shift & Unshift Method

Shift() ⇒ Delete first value

Unshift() ⇒ Add value to the fist index

```
a.shift();
doc.write(a);
                ⇒ New value
a.unshift( );
doc.write(a);
```

Concat() => To merge (जोडना)

Join() => Array की जितनी भी Values है उन सबको मिलाके एक Value बना देता है। (String बना देता है।)

```
var a = ["Drishti", "Pratibha", "Anshika"];

var b = a.concat("Kumkum", "Anju");

doc.write(b);

var a = [                    ]
var b = [                    ]
var c = a.concat(b);
doc.write(c);
```

If we want to concat more than one array =>

```
var c = a.concat(b,d);
```

Date : / /
Page No.

```
var a = [                    ]
var b = [                    ]
var c = a.concat(b);
doc.write(c);
var d = c.join(" - ");
doc.write(d);
```

(start, end) Slice() ⇒ Existing array में से एक या एक से ज़्यादा value निकाल सकते है।

Splice() ⇒ Value add कराने के लिए

```
var a = [                    ];

var b = a.slice(1, 4);

var a = [                    ];
```

→ हमे कुछ Delete नही करना।

```
a.splice(2, 0, "Neha", "Drishti");
```

कौनसे Index पर Value Insert करनी है।

जो Value Insert करनी है।

isArray()

```
var a = ["Drishti", "Pratibha", "Anshika"];

var b = Array.isArray(a);

doc.write(b);
```

#

```
var a = "Drishti";

if( Array.isArray(a)){
document.write("This is an Array");
} else {
document.write("This is not an Array");
}
```

indexOf() ⇒ Existing Value का Index check करने के लिए

(searchitem, Start)

```
var a = ["Sanjay", "Aman", "Rehman", "Aman",
         "Rahul"];

var b = a.indexOf("Aman");

var b = a.indexOf("Aman", 2);
```

Date : / /
Page No.

lastIndexOf() ⇒ Last से Index Value Return करता है।

```
var c = a.lastIndexOf("Aman");
var c = a.lastIndexOf("Aman", 2);
```

Includes()

```
var a = [                                    ];
                                  → Existing Value name.
var b = a.includes("      ");
doc.write(b);
```

Some() ⇒ Checks if any of the elements in an array pass a test.

```
var ages = [10, 20, 18, 13];

var b = ages.some(check);
doc.write(b);

function check(age){
    return age >=18;
```

every() => सभी values Test में Pass करनी जरूरी है।

```
var ages = [10, 18, 20, 25];

var b = ages.every(check);
doc.write(b);

function check(age) {
    return age >= 18;
}
```

find() => returns the value of the first element in an array that pass a test.

findIndex() => returns the index of the first element in an array that pass a test.

```
var ages = [10, 23, 19, 20];

var b = ages.find(check);
doc.write(b);

function check(age) {
    return age >= 18;
}
```

For findIndex => Just write findIndex in place of find.

Date : / /
Page No.

filter() => method creates an array filled with all array elements that pass a test

```
var ages = [10, 12, 19, 20];

var b = ages.filter(check);
doc.write(b);

function check(age) {
    return age >= 18;
}
```

toString() => Converts an array into a string

```
var a = ["Drishti", "Jyotibha", "Anshika"];

a.toString();

doc.write(a);
```

valueOf() => Returns the array

fill() => method fills all the elements in an array with a static value.

```
var a = [                    ];
a.fill("Ram");
doc.write(a);
```

forEach()

```
var a = [                    ];

a.forEach(function(value, index){
    doc.write(index + " : " + value + "<br>");
    })
```

Javascript Object

```
var a = {
        fname : "Drishti",
        lname : "Chakarvarty",
        age   : 18
        Country : "India"
    };
document.write(a.fname);
```

Properties

When you want to print array value in Object

```
var a = {
    fname : 'Drishti',
    lname : 'Chakarvarty',
    favMovies : ['Dhoom', 'Sholay', 'DDLJ']
        }
document.write(a.favMovies[0]);
```

Date : / /
Page No.

We can also call function in Object

```
Salary : function () {
    return 25000;
}
```
Salary → Method

```
doc.write (a.Salary());
```

Calling another f^n (Concatenation)

```
full name : function () {
    return this.fname + " " + this.lname;
}
```
this → keyword to call fname

```
doc.write (a.fullname());
```

Object within an Object

```
var a = {
    fname : 'Drishti',
    lname : 'Chakraverty',
    age : 22,
    living : {
        'city' : 'Delhi',
        'country' : 'India'
    },
    full name : function () {
```

```
return this.fname + " " + this.lname;
    }
}
doc.write(a.living.city);
```

Creating Object using new Keyword

```
var person = new Object();

person.firstname = 'Ram';
person.lastname = 'Kumar';
person.age = 25;
console.log(person);
doc.write(person);
          person.age
doc.write(person['firstname']);
```

Array of Objects

```
var student = [
    {Name: "Ram", age: 20},
    {Name: 'Sham', age: 18},
    {Name: 'Rahul', age: 15},
];
console.log(student);

for(var a=0; a<student.length; a++){
```

Date : / /
Page No.

~~Using fo~~

```
document.write(Student[a].name + "  " + Student[a].
        age + "<br>");
}
```

Const variable with Array & Objects

```
Const a = [10, 20, 30];
a[1] = 25;
a[2] = 35;

console.log(a);
```

```
Const a = {
    name : 'Ram',
    age : 25
};

a.name = 'Drishti';
a.age = 22;
console.log(a);
```

For In loop

```
for (var i in a) {
        Statement
}
```

```
var obj = {
    fname : 'Drishti',
    lname : 'Chakravarty',
    Age   : 22,
    City  : 'Delhi'
    };
for (var key in obj){
    doc.write(obj[key] + "<br>");
doc.write(key + " : " + obj[key] + "<br>");
```

Array - Map() function

```
a.map(function () {
        Statement
    });

var ary = [11, 4, 9, 16];

var b = ary.map(test);
doc.write(b);
```

Date : / /
Page No. 56 to 60

```
function test (x) {
    return x * 10;
}

var ary = [
{ fname: "A" , lname: "B" },
{                          }
{                          }
];

var b = ary.map(test);
document.write(b);

function test (x) {
    return x.fname + "  " + x.lname;
}
```

~~String Methods~~

Date Object

```
var now = new Date();

doc.write(now);
```

toDateString => Only to print day & Date i.e. Readable format

```
var now = new Date();
document.write(now.toDateString());
```

getDate() ⇒ Simple Date

```
doc.write(now.getDate());
```

getFullYear() ⇒ Print full year

```
doc.write(now.getFullYear());
```

getMonth() ⇒ Value 0 (January)

```
doc.write(now.getMonth());
```

getDay() ⇒ Return week Day 0 (Sunday)

```
doc.write(now.getDay());
```


```
var now = new Date('January 5 2010');
doc.write(now.getDay());
```

getMonth
getFullYear

Date : / /
Page No.

getHours =>

```
var now = new Date();

doc.write(now.getHours());
          getMinutes
          getSeconds
          getMilliseconds
```

SetDate() => To See the future value

```
var now = new Date();
now.setDate(20);

document.write(now);
```

SetFullYear(2020);

```
now.setFullYear(2020);
doc.write(now.getMonth);
```

Set Month

```
now.setMonth(4);
doc.write(now);
```

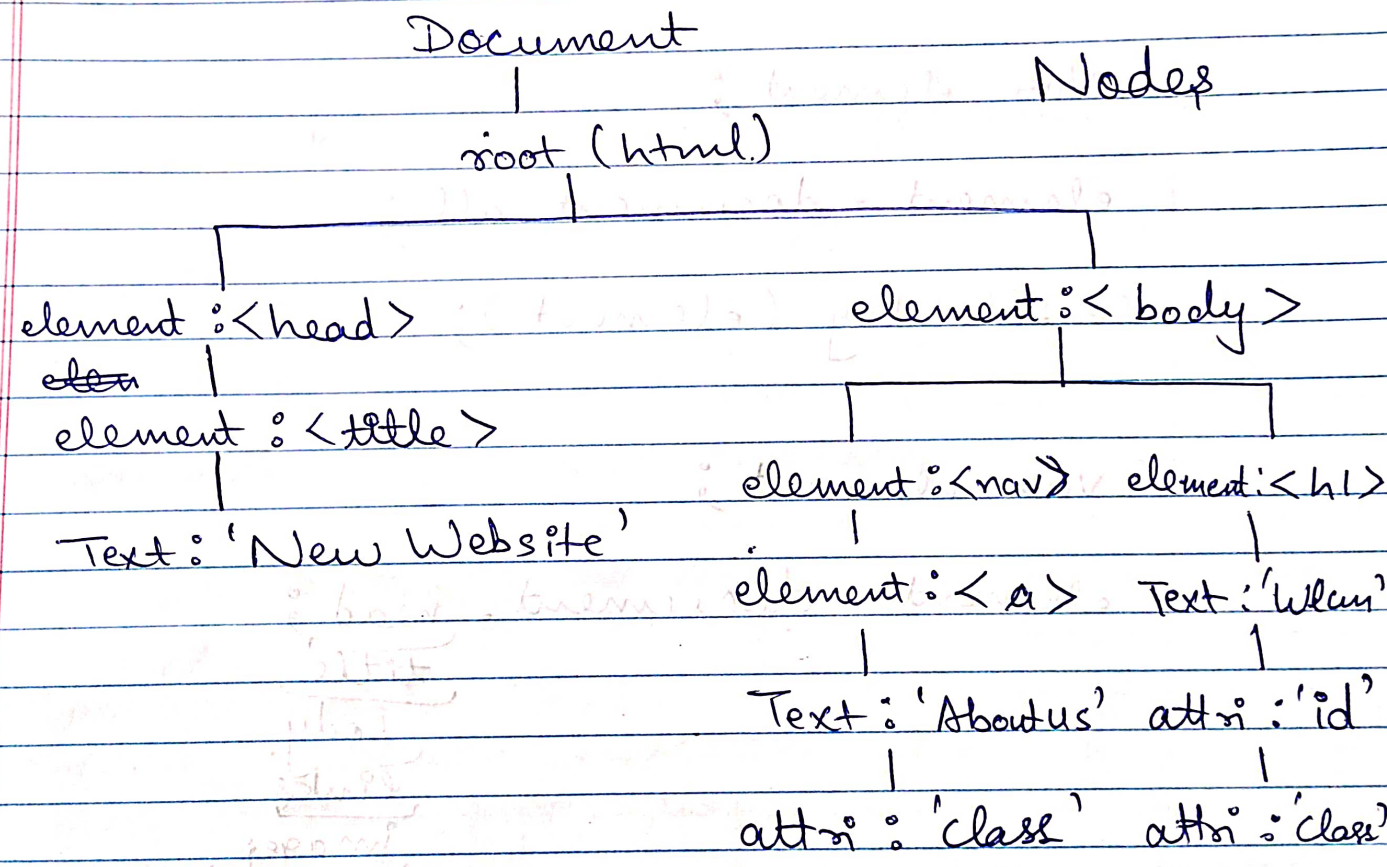
Set Hours

```
now.setHours(14);
doc.write(now);
```

```
var now = new Date();

document.write(now.getDate() + "/" + now.getMonth()
    + "/" + now.getFullYear());
```

Document Object Model



Date : / /
Page No.

How to Target DOM Object :

- Id → document.getElementById(id)
- Class Name → document.getElementsByClassName(name)
- Tag Name → document.getElementsByTagName(name)

```
var element;
element = document.all;
console.log(element);
```

```
var element;
element = document.head;
```

title
body
links
images
forms
doctype
URL
domain
baseURI

```
var element;

element = document.getElementById("header");
                                   Id name
console.log(element)

element = document.getElementsByClassName("list")[0];

element = document.getElementsByTagName("ul");
```