JAVASCRIPT

HTML is used to create a web page.

CSS is used to define styles to a web page.

JavaScript is used to add dynamic behaviour to a web page.

HTML & CSS can be used at client side only where as JavaScript can be used both at client side & server side.

Initially JavaScript was created to use client side only. Now it can be used at both client side & server side.

Java Script initial name was LiveScript.

It was created by Brendan Eich in 1995.

It is became an ECMA standard in 1997.

ECMA-262 is the official name of the standard.

ECMAScript is the official name of the language.

ECMA stands for European Computer Manufacturers Association.

Java script is the world's most popular programming language.

Java script is the programming language of the web.

Java script can change HTML content.

Java script can change HTML attribute values.

Java script can change HTML styles (CSS).

JavaScript Language Basics:

1) Identifiers
2) Keywords
3) Literals
4) Data Types
5) Variables
6) Type Conversions
7) Operations On Data
8) Program Structure
9) Operators
10) Control Statements

Identifiers:

An identifier is a word and it is used to identify variable, method, function, .. etc.,

It can be a variable name, method name, function name,.. etc.,

Rules to declare an identifier:

1) It can be formed by using alphabets(A to Z and a to z), digits(0 to 9), underscore symbol(_) and dollar symbol($).
2) It must begins with alphabet, underscore symbol or dollar symbol.
3) The length of the identifier is not limited.
4) It should not contain special symbols other than underscore and dollar symbols.
5) It should not contain white space characters(Space bar, tab and enter keys).

Keywords:

A set of words reserved by language itself and those words are called keywords.

Examples:

if, else, switch, for, while, do, in, of, break, continue, null, true, false, function, .. etc.,

Initially 56 keywords. 8 keywords newly added & 16 keywords removed. At present total 48 keywords in JavaScript.

New keywords are await, class, enum, export, extends, import, let & super. Removed keywords are byte, short, int, long, float, double, char, boolean, abstract, final, goto, native, synchronized, throws, transient & volatile.

Note: Keyword cannot be used as an identifier.

Literals:

A literal is a source code representation of a fixed value.

In JavaScript, literals are divided into the following categories:

1) Numeric Literals(Decimal, Binary, Octal, Hexadecimal, Floating point, Exponential, BigInt)
2) String Literals
3) Boolean Literals
4) Object Literals

Data Types:

1) Undefined

2) Null

3) Boolean

4) String

5) Symbol

6) Number

7) BigInt

8) Object

Variables:

A JavaScript variable can hold any type of data. Variable can be declared by using let, const or var keywords.

Basics Programs:

Example1:

```
<html><body><script>
document.write("Hello World");
</script></body></html>
```

Example2:

```
<html><body><script>
console.log("Welcome");
</script></body></html>
```

Example3:

```
<html><body><script>
alert("Hello");
</script></body></html>
```

Example4:

```
<html><body><script>
let a;
a=500;
a=600;
document.write(a);
const b; => Invalid
b=700; => Invalid
document.write(b)
const c=800;
document.write(c)
let x=100;
document.write(x)
var y=200;
document.write(y);
z=300;
document.write(z)
</script></body></html>
```

Type Conversions:

Value	Number	String	Boolean
0	0	"0"	false
1	1	"1"	true

5	5	"5"	true
-5	-5	"-5"	true
false	0	"false"	false
true	1	"true"	true
""	0	""	false
"0"	0	"0"	true
"abc"	NaN	"abc"	true
"1a"	NaN	"1a"	true

Operations On Data:

Number + Number = Number

Number + String = String

Number + Boolean = Number

String + Number = String

String + String = String

String + Boolean = String

Boolean + Number = Number

Boolean + String = String

Boolean + Boolean = Number

Operators:

1) Arithmetic Operators(+,-,*,/,%,**)

2) Relational Operators(<,>,<=,>=,==,!=,===,!==)

3) Logical Operators(&&,||,!)

4) Increment & Decrement Operators(++,--)

5) Bitwise Operators(&,|,^,<<,>>,>>>,~)

6) Assignment Operators(=,+=,-=,*=,/=,%=,**=,&=,|=,^=,<<=,>>=,>>>=)

7) Ternary Operator(?:)

8) Unary Plus & Unary Minus Operators(+,-)

9) Other Operators(typeof, instanceof, .. etc.,)

Control Flow Statements:

1) Selection Statements
   i) if Statement
   ii) if else Statement
   iii) if else if .. else Statement
   iv) Nested if Statement
   v) switch Statement
2) Iteration Statements
   i) while loop
   ii) do while loop
   iii) for loop
   iv) for in loop
   v) for of loop
   vi) Nested loops
3) Jump Statements
   i) break Statement
   ii) break LABEL Statement
   iii) continue Statement
   iv) continue LABEL Statement
   v) return Statement

Inline JavaScript Example:

```
<!DOCTYPE html>
<html>
<body>
<input type="button" value="Click Me"
onclick=document.write("Welcome")>
</body>
</html>
```

Internal JavaScript Example:

```
<!DOCTYPE html>
<html>
<body>
<script>
document.write("Welcome to JavaScript");
</script>
</body>
</html>
```

External JavaScript Example:

```
<!DOCTYPE html>
<html>
<body>
<script src="demo.js"></script>
```

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

demo.js

```
document.write("Hello");
```

By

Mr. Venkatesh Mansani

Naresh i Technologies

JAVA SCRIPT

BASIC PROGRAMS

Literals:

<!-- Program to demonstrate decimal, octal, hexadecimal & binary literals -->

```
<html>
<body>
<script>
let a=30;
let b=036;
let c=0x1e;
let d=0b11110;
document.write(a);
document.write(b);
document.write(c);
document.write(d);
</script>
</body>
</html>
```

Octal literal must be prefixed with 0 (or) 0o (or) 0O.

Hexadecimal literal must be prefixed with 0x (or) 0X.

Binary literal must be prefixed with 0b (or) 0B.

<!-- Program to demonstrate bigint literal -->

```
<html>
<body>
<script>
let a=999999999999999;
let b=9999999999999999;
let c=9999999999999999n;
document.write(a+"<br>");
document.write(b+"<br>");
document.write(c);
</script>
</body>
</html>
```

If the number is having more than 15 digits then use bigint literal.

Bigint literal must be suffixed with n.

<!-- Program to demonstrate string literals -->

```
<html>
<body>
<script>
let s1='welcome';
```

```
let s2="welcome";
let s3=`welcome`;
let s4=`Welcome to
                    Naresh IT`;
document.write(s1+"<br>");
document.write(s2+"<br>");
document.write(s3+"<br>");
document.write(s4);
</script>
</body>
</html>
```

Note: Multi line string literal must be in backticks only.

<!-- Program to demonstrate boolean & object literals -->

```
<html>
<body>
<script>
let a=true;
let b=false;
let c=null;
document.write(a);
document.write(b);
document.write(c);
```

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

Data Types & typeof Operator:

<!-- Program to demonstrate data types & typeof operator -->

```
<html>
<body>
<script>
let a=10;
let b=10n;
let c="hello";
let d=true;
let e=null;
let f;
document.write(typeof(a));
document.write(typeof(b));
document.write(typeof(c));
document.write(typeof(d));
document.write(typeof(e));
document.write(typeof(f));
</script>
</body>
```

```
</html>
```

typeof operator is used to get the type of data.

Here type decided dynamically whenever variable is initialized.

typeof operator can be used as follows:

typeof(variable) (or) typeof variable

<!-- Program to convert Number to String & Boolean -->

```
<html>
<body>
<script>
let a=-5;
let b=String(a);
let c=Boolean(a);
document.write(a);
document.write(typeof(a));
document.write(b);
document.write(typeof(b));
document.write(c);
document.write(typeof(c));
</script>
</body>
</html>
```

<!-- Program to convert Boolean to Number & String -->

```
<html>
<body>
<script>
let a=false;
let b=Number(a);
let c=String(a);
document.write(a);
document.write(typeof(a));
document.write(b);
document.write(typeof(b));
document.write(c);
document.write(typeof(c));
</script>
</body>
</html>
```

<!-- Program to convert String to Number & Boolean -->

```
<html>
<body>
<script>
let a="59a";
let b=Number(a);
```

```
let c=Boolean(a);
document.write(a);
document.write(typeof(a));
document.write(b);
document.write(typeof(b));
document.write(c);
document.write(typeof(c));
</script>
</body>
</html>
```

By

Mr. Venkatesh Mansani

Naresh i Technologies

JAVA SCRIPT LANGUAGE BASICS PART 2

Operations On Data:

Number + Number => Number

Number + String => String

Number + Boolean => Number

String + Number => String

String + String => String

String + Boolean => String

Boolean + Number => Number

Boolean + String => String

Boolean + Boolean => Number

<!-- Program to demonstrate operations on data -->

```
<html>
<body>
<script>
let a=5,b="hello",c=true;
document.write(a+a, a+b, a+c);
document.write(b+a, b+b, b+c);
document.write(c+a, c+b, c+c);
```

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

Operators:

An operator is a special symbol that operates on data.

In Java Script, operators are divided into the following categories.

1) Arithmetic Operators

2) Relational Operators

3) Logical Operators

4) Increment & Decrement Operators

5) Bitwise Operators

6) Assignment Operators

7) Conditional Operators

8) Unary Plus & Unary Minus Operators

9) Other Operators

Arithmetic Operators:

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo Division

** Exponentiation

```
<!-- Program to demonstrate arithmetic operators -->
<html>
<body>
<script>
let a=5, b=3, c=2;
document.write(a*b+c);
document.write(a+b*c);
</script>
</body>
</html>
```

```
<!-- Program to demonstrate arithmetic operators -->
<html>
<body>
<script>
let a=5, b=3, c=2, d=7;
document.write(a+b*c+a**c%d);
</script>
</body>
</html>
```

```
<!-- Program to demonstrate arithmetic operators -->
<html>
```

```
<body>
<script>
let a=5, b=3, c=2, d=7;
document.write((a+b)*(c-d)%b);
</script>
</body>
</html>
```

Relational Operators:

Operator	Meaning
<	Less than
>	Greater than
<=	Less than or equals to
>=	Greater than or equals to
==	Equals to
!=	Not equals to
===	Strict equals to (or) Equal type & equal value
!==	Strict Not equals to (or) Not equal type & not equal value

The above all operators return boolean type values(true or false).

<!-- Program to demonstrate relational operators -->

```
<html>
<body>
<script>
let a=5, b="5";
document.write(a==b);
document.write(a!=b);
document.write(a===b);
document.write(a!==b);
</script>
</body>
</html>
```

Logical Operators:

Operator	Meaning
&&	Logical AND
||	Logical OR
!	Logical NOT

<!-- Program to demonstrate logical operators -->

```
<html>
<body>
<script>
let a=5, b=3, c=2;
let d=!((a>b)&&(a>c));
```

```
let e=(a>b)||(a>c);
document.write(d);
document.write(e);
</script>
</body>
</html>
```

Types of Operators:

There are 3 types of operators:

1) Unary Operator

2) Binary Operator

3) Ternary Operator

Unary Operator:

An operator that operates on only one operand is called as unary operator.

Here operand can be a variable, literal or expression.

Binary Operator:

An operator that operates on two operands is called as binary operator.

Ternary Operator:

An operator that operates on three operands is called as ternary operator.

Increment & Decrement Operators:

Operator	Meaning

++	Increment By 1
--	Decrement By 1

++a => Pre increment

a++ => Post increment

--a => Pre decrement

a-- => Post decrement

<!-- Program to demonstrate increment operators -->

```
<html>
<body>
<script>
let a=10, b=3;
let c=a++ + ++b;
document.write(a,b,c);
</script>
</body>
</html>
```

Bitwise Operators:

Operator	Meaning
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
<<	Left Shift

>>	Right Shift
~	Complement (tilde)

The above all bitwise operators operate on binary data.

<!-- Program to demonstrate bitwise operators -->

```
<html>
<body>
<script>
let a=9, b=13;
document.write((a&b)+"<br>");
document.write((a|b)+"<br>");
document.write(a^b);
</script>
</body>
</html>
```

<!-- Program to demonstrate bitwise operators -->

```
<html>
<body>
<script>
let a=9;
document.write((a<<2)+"<br>");
document.write(a>>2);
</script>
```

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

<!-- Program to demonstrate bitwise operators -->

```
<html>
<body>
<script>
let a=53;
document.write(~a);
</script>
</body>
</html>
```

Assignment Operators:

Operator	Meaning
=	Normal Assignment
a+=b;	a=a+b;
a-=b;	a=a-b;
a*=b;	a=a*b;
a/=b;	a=a/b;
a%=b;	a=a%b;
a**=b;	a=a**b;
a&=b;	a=a&b;
a|=b;	a=a|b;

a^=b; a=a^b;

a<<=b; a=a<<b;

a>>=b; a=a>>b;

Conditional Operators:

It is a ternary operator & it operates on three operands.

Syntax:

operand1 ? operand2 : operand3

Example:

```
<html>
<body>
<script>
let a=5, b=3;
let c=(a>b)?a:b;
document.write(c);
</script>
</body>
</html>
```

Unary Plus & Unary Minus Operators:

Operator	Meaning
+	Positive
-	Negative

Example:

```
<html>
<body>
<script>
let a=-5;
let b=-a;
document.write(b);
</script>
</body>
</html>
```

By

Mr. Venkatesh Mansani

Naresh i Technologies

JAVA SCRIPT LANGUAGE BASICS PART 3

Control Flow Statements:

1) Conditional Statements:

i) if Statement

ii) if else Statement

iii) if else if ... else Statement

iv) Nested if Statement

v) switch Statement

2) Iterative Statements(Loops):

i) while loop

ii) do while loop

iii) for loop

iv) for in loop

v) for of loop

vi) Nested loops

3) Loop Controlling Statements:

i) break Statement

ii) break LABEL Statement

iii) continue Statement

iv) continue LABEL Statement

<!-- Program to demonstrate if else if ... else statement -->

```
<html>
<body>
<script>
let a=0;
if(a>0)
	document.write("Positive Number");
else if(a<0)
	document.write("Negative Number");
else
	document.write("Zero");
</script>
</body>
</html>
```

<!-- Program to demonstrate nested if statement -->

```
<html>
<body>
<script>
let a=-10;
if(a>0)
{
	if(a%2==0)
```

```
        document.write("Even Number");
    else
        document.write("Odd Number");
}
else
{
    if(a<0)
        document.write("Negative Number");
    else
        document.writer("Zero");
}
</script>
</body>
</html>
```

<!-- Program to demonstrate switch statement -->

```
<html>
<body>
<script>
let a=3;
switch(a)
{
    case 1: document.write("One");
```

```
                break;
        case 2: document.write("Two");
                        break;
        default: document.write("Invalid");
}
</script>
</body>
</html>
```

<!-- Program to demonstrate switch statement -->

```
<html>
<body>
<script>
let a="mon";
switch(a)
{
        case "mon":
        case "tue":
        case "wed":
        case "thu":
        case "fri": document.write("Working Day");
                                break;
        case "sat":
```

```
        case "sun": document.write("Holiday");
                        break;
        default: document.write("Invalid Day");
}
</script>
</body>
</html>
```

<!-- Program to demonstrate while loop -->

```
<html>
<body>
<script>
let a=1;
while(a<=10)
{
        document.write(a);
        a++;
}
</script>
</body>
</html>
```

<!-- Program to demonstrate do while loop -->

```
<html>
<body>
<script>
let a=1;
do
{
	document.write(a);
	a++;
}while(a<=10);
</script>
</body>
</html>
```

<!-- Program to demonstrate for loop -->

```
<html>
<body>
<script>
for(let a=1;a<=10;a++)
{
	document.write(a);
}
</script>
```

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

<!-- Program to demonstrate nested loops -->

```
<html>
<body>
<script>
for(let j=1;j<=5;j++)
{
        for(let i=1;i<=j;i++)
        {
                document.write(i);
        }
        document.write("<br>");
}
</script>
</body>
</html>
```

break Statement:

It terminates the nearest enclosing loop or switch statement.

Example1:

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

```
<script>
for(let i=1;i<=10;i++)
{
	if(i==5)
		break;
	document.write(i+"<br>");
}
</script>
</body>
</html>
```

Example2:

```
<html>
<body>
<script>
for(let i=1;i<=3;i++)
{
	for(let j=1;j<=10;j++)
	{
		if(j==5)
			break;
		document.write(j+"<br>");
	}
```

```
}
</script>
</body>
</html>
```

break LABEL Statement:

It terminates the specified LABEL loop. Here break is a keyword & LABEL is an identifier.

Example:

```
<html><body>
<script>
FIRST: for(let i=1;i<=3;i++)
{
        SECOND: for(let j=1;j<=10;j++)
        {
                if(j==5)
                        break FIRST;
                document.write(j+"<br>");
        }
}
</script>
</body>
</html>
```

continue Statement:

It passes the control to the next iteration of a loop.

Example:

```
<html>
<body>
<script>
for(let i=1;i<=3;i++)
{
        for(let j=1;j<=10;j++)
        {
                if(j>5)
                  continue;
                document.write(j+"<br>");
        }
}
</script>
</body>
</html>
```

continue LABEL Statement:

It passes the control to the specified LABEL loop. Here continue is a keyword & LABEL is an identifier.

Example:

```
<html>
<body>
<script>
FIRST:for(let i=1;i<=3;i++)
{
        SECOND:for(let j=1;j<=10;j++)
        {
                if(j==5)
                  continue FIRST;
                document.write(j+"<br>");
        }
}
</script>
</body>
</html>
```


By

Mr. Venkatesh Mansani

Naresh i Technologies

DOCUMENT OBJECT MODEL

Java Script HTML DOM:

The HTML DOM(Document Object Model) is created by browser whenever a web page is loaded.

The HTML DOM is created as a tree of objects.

Example:

demo.html

```
<html>
<head>
<title>Home Page</title>
</head>
<body bgcolor="yellow" text="red">
<h1>Welcome</h1>
</body>
</html>
```

The DOM tree created by browser for the above html document as follows:

Document(demo.html) => Root Element(<html>)=>Element(<head>) & Element(<body>)

Element(<head>) => Element(<title>) => Text("Home Page")

Element(<body>) => Element(<h1>), Attribute(bgcolor="yellow") & Attribute(text="red")

Element(<h1>) => Text("Welcome")

Java Script can access all html elements, attributes & content of an html document.

Java Script can do the following on html document:

1) It can change the content of html elements.

2) It can change the values of html attributes.

3) It can change the css styles.

4) It can add new html elements.

5) It can add new html attributes.

6) It can remove existing html elements.

7) It can remove existing html attributes.

8) It can create new html events.

9) It can remove existing html events.

10) It can react to all existing html events.

HTML DOM methods are used to perform actions on html elements.

HTML DOM properties are used to change the values of html elements.

Example:

```
document.getElementById("demo").innerHTML="Welcome";
```

In the above example, getElementById() is a method & innerHTML is a property.

The following methods are used to find the html elements:

1) document.getElementById(id) => It is used to find an element by element id.

2) document.getElementsByTagName(name) => It is used to find elements by tag name.

3) document.getElementsByClassName(name) => It is used to find elements by class name.

4) document.querySelector(selector) => It is used to get the first element based on selector.

5) document.querySelectorAll(selector) => It is used to find elements by selector.

The following properties are used to change the html elements:

1) element.innerHTML=new HTML content => It is used to change the content.

2) element.attribute=new value => It is used to change the attribute value.

3) element.style.property=new style => It is used to change the style

The following methods are used to add & remove elements:

1) document.createElement(element) => It is used to create an element.

2) document.appendChild(element) => It is used to add an element.

3) document.removeChild(element) => It is used to remove an element.

4) document.replaceChild(new, old) => It is used to replace an element.

The following methods are used to add & remove event listeners:

1) element.addEventListener(event, function) => It is used to add event listener

2) element.removeEventListener(event, function) => It is used to remove event listener

1) <!-- Program to demonstrate getElementById() method -->

```
<html>
 <body>
        <h1 id="one">HTML</h1>
        <h1 id="two">CSS</h1>
        <h1 id="three">JS</h1>
        <script>
              let element=document.getElementById("one");
              element.innerHTML="Hyper Text Markup Language";
        </script>
 </body>
</html>
```

2) <!-- Program to demonstrate getElementsByTagName() method -->

```
<html>
```

```
<body>
    <h1>Hyper Text Markup Language</h1>
    <h1>Cascading Style Sheets</h1>
    <h1>Java Script</h1>
    <script>
        let elements=document.getElementsByTagName("h1");
        elements[0].align="left";
        elements[1].align="center";
        elements[2].align="right";
    </script>
</body>
</html>
```

3) <!-- Program to demonstrate getElementsByClassName() method -->

```
<html>
 <body>
    <h1 class="test">Hyper Text Markup Language</h1>
    <h1 class="test">Cascading Style Sheets</h1>
    <h1>Java Script</h1>
    <script>
        let elements=document.getElementsByClassName("test");
        elements[0].style.color="red";
```

```
            elements[1].style.color="green";
        </script>
 </body>
</html>
```

4) <!-- Program to demonstrate querySelector() method -->

```
<html>
 <body>
        <h1 id="test">Hyper Text Markup Language</h1>
        <h1>Cascading Style Sheets</h1>
        <h1>Java Script</h1>
        <script>
            let element=document.querySelector("#test");
            element.style.backgroundColor="yellow";
        </script>
 </body>
</html>
```

5) <!-- Program to demonstrate querySelectorAll() method -->

```
<html>
 <body>
        <h1 class="test">Hyper Text Markup Language</h1>
        <h1 class="test">Cascading Style Sheets</h1>
        <h1>Java Script</h1>
```

```
        <script>
            let elements=document.querySelectorAll(".test");
            elements[0].style.backgroundColor="red";
            elements[1].style.backgroundColor="green";
        </script>
 </body>
</html>
```

By

Mr. Venkatesh Mansani

Naresh i Technologies

FUNCTIONS

Function:

A group of statements into a single logical unit is called as function. Functions are used to perform the task.

Function is not called automatically. Function must be called explicitly after defining it.

By using function keyword we can define a function.

Advantages of Functions:

1) Modularity
2) Reusability

There are four categories of functions:

1) Functions with arguments and with return value.
2) Functions with arguments and without return value.
3) Functions without arguments and with return value
4) Functions without arguments and without return value.

1) <!-- Function with arguments and with return value -->

```
<html><body><script>

function add(a,b)

{

        return a+b;

}

let x=add(5,3);
```

```
document.write(x);
</script></body></html>
2) <!-- Function with arguments and without return value -->
<html><body><script>
function add(a,b)
{
	document.write(a+b);
}
add(9,3);
</script></body></html>
3) <!-- Function without arguments and with return value -->
<html><body><script>
function get()
{
	return 10;
}
let x=get();
document.write(x);
</script></body></html>
4) <!-- Function without arguments and without return value -->
<html><body><script>
function display()
```

```
{
        document.write("Welcome");
}
display();
</script></body></html>
```

Anonymous Function:

A function that has no name is called as an anonymous function.

Example:

```
<html><body><script>
let x=function(a,b)
{
        return a+b;
}
let y=x(10,20);
document.write(y);
</script></body></html>
```

Self Invoking Function:

A function that can be called itself while defining it is called as self invoking function. It is also one type of anonymous function.

Example:

```
(function(a,b)
{
```

```
    document.write(a+b);
})(10,20);
```

Arrow Functions:

Arrow function that allows to write an anonymous function new way without using function keyword, return keyword and curly braces.

Example1:

```
<html><body><script>
let x=(a,b)=>a+b;
let y=x(10,20);
document.write(y);
</script></body></html>
```

Example2:

```
<html><body><script>
let x=(a,b)=>document.write(a+b);
x(10,20);
</script></body></html>
```


By

Mr. Venkatesh Mansani

Naresh i Technologies

EVENT HANDLING

Event Handling:

Web browser generates events, those events are handled by HTML & JavaScript reacts to events.

For Example, when user clicks a button then the web browser generates event, that event handled by html event attribute & JavaScript code executed.

Window Events	Event Handling Attributes in HTML
1) load	onload
2) unload	onunload
3) resize	onresize

Key Events	Event Handling Attributes in HTML
1) keyup	onkeyup
2) keydown	onkeydown
3) keypress	onkeypress

Mouse Events	Event Handling Attributes in HTML
1) mouseup	onmouseup
2) mousedown	onmousedown
3) mouseover	onmouseover
4) mouseout	onmouseout
5) click	onclick

6) dblclick	ondblclick
7) contextmenu	oncontextmenu

Form Events	Event Handling Attributes in HTML
1) blur	onblur
2) focus	onfocus
3) change	onchange
4) submit	onsubmit
5) reset	onreset

<!-- Program to demonstrate & mouseup, mousedown & mouseover events -->

```
<html>
<body>
<h1 id="demo">Welcome to Naresh IT</h1>
<input type="button" value="Change the Content"
onmouseup="fun1()" onmousedown="fun2()" onmouseover="fun3()">
<script>
function fun1()
{
        let element=document.getElementById("demo");
        element.innerHTML="Welcome to HTML";
}
function fun2()
{
        let element=document.getElementById("demo");
        element.innerHTML="Welcome to CSS";
}
```

```
function fun3()
{
        let element=document.getElementById("demo");
        element.innerHTML="Welcome to Java Script";
}
</script>
</body>
</html>
```

<!-- Program to demonstrate click & dblclick events -->

```
<html>
<body>
<h1 id="demo">Welcome to Naresh IT</h1>
<input type="button" value="Change the Alignment" onclick="fun1()"
ondblclick="fun2()">
<script>
function fun1()
{
        let element=document.getElementById("demo");
        element.align="center";
}
function fun2()
{
        let element=document.getElementById("demo");
        element.align="right";
}
</script>
</body>
</html>
```

<!-- Program to demonstrate mouseout & mouseover events -->

```
<html>
<body>
<h1 id="demo">Welcome to Naresh IT</h1>
<input type="button" value="Change the Font Color"
onmouseover="fun1()" onmouseout="fun2()">
<script>
function fun1()
{
        let element=document.getElementById("demo");
        element.style.color="red";
}
function fun2()
{
        let element=document.getElementById("demo");
        element.style.color="green";
}
</script>
</body>
</html>
```

<!-- Program to demonstrate click & contextmenu events -->

```
<html>
<body>
<h1>Hyper Text Markup Language</h1>
<h1>Cascading Style Sheets</h1>
<h1>Java Script</h1>
<input type="button" value="Click Here" onclick="changeFontColor()"
oncontextmenu="changeBackgroundColor()">
<script>
```

```
function changeFontColor()
{
	let elements=document.getElementsByTagName("h1");
	elements[0].style.color="red";
	elements[1].style.color="green";
	elements[2].style.color="blue";
}
function changeBackgroundColor()
{
	let elements=document.getElementsByTagName("body");
	elements[0].style.backgroundColor="yellow";
}
</script>
</body>
</html>
```

<!-- Program to demonstrate mouseover & mouseout events -->

```
<html>
<body>
<img id="image" src="C:/flowers.jfif" width="100" height="100"
onmouseover="big()" onmouseout="small()">
<script>
function big()
{
	let element=document.getElementById("image");
	element.style.width="200px";
	element.style.height="200px";
	element.style.borderStyle="solid";
	element.style.borderRadius="100px";
}
```

```
function small()
{
        let element=document.getElementById("image");
        element.style.width="100px";
        element.style.height="100px";
        element.style.borderStyle="double";
        element.style.borderRadius="0px";
}
</script>
</body>
</html>
```

By

Mr. Venkatesh Mansani

Naresh i Technologies