1) Introduction of javascript

JavaScript was first known as **LiveScript.**

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

```
<html>
  <body>
    <script language="javascript" type="text/javascript">
      <!--
        document.write("Hello World!");
      //-->
    </script>
  </body>
</html>
```

Old JavaScript examples may use a type attribute: <script type="text/javascript">. The type attribute is not required. JavaScript is the default scripting language in HTML.

2) Where JavaScript is used

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

- Client-side validation
- Dynamic drop-down menus
- Displaying data and time
- Displaying popup windows and dialog boxes (like alert dialog box, confirm dialog box and prompt dialog box)
- Displaying clocks etc.

3) Advantages of javascript

- Less server interaction
- Immediate feedback to the visitors
- Increased interactivity
- Richer interfaces

4) Syntax of javascript

<script ...>

JavaScript code

</script>

5) What is innerHTML?

One of many JavaScript HTML methods is **getElementById()**.

This example uses the method to "find" an HTML element (with id="demo") and changes the element content (**innerHTML**) to "Hello JavaScript":

<!DOCTYPE html>

<html>

<body>

<h2>What Can JavaScript Do?</h2>

```
<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!"'>Click Me!</button>

</body>

</html>
```

6) What is script tag?

In HTML, JavaScript code must be inserted between <script> and </script> tags.

```
<!DOCTYPE html>

<html>

<body>

<h2>JavaScript in Body</h2>

<p id="demo"></p>

<script>

document.getElementById("demo").innerHTML = "My First JavaScript";

</script>

</body>

</html>
```

8) Where the <script> tag should be placed?

JavaScript in <head>...</head> section

If you want to have a script run on some event, such as when a user clicks somewhere, then you will place that script in the head as follows −

```
<html>

  <head>

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

```
      <!--
         function sayHello() {
            alert("Hello World")
         }
      //-->
   </script>
</head>
<body>
   <input type="button" onclick="sayHello()" value="Say Hello" />
</body>
</html>
```

9) JavaScript in <body>...</body> section

If you need a script to run as the page loads so that the script generates content in the page, then the script goes in the <body> portion of the document. In this case, you would not have any function defined using JavaScript. Take a look at the following code.

```
<html>

<head>
</head>

<body>

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

```
      <!--
         document.write("Hello World")
      //-->
   </script>

   <p>This is web page body </p>

</body>
</html>
```

10) JavaScript in <body> and <head> Sections

You can put your JavaScript code in <head> and <body> section altogether as follows –

```
<html>
   <head>
      <script type="text/javascript">
         <!--
            function sayHello() {
               alert("Hello World")
            }
         //-->
      </script>
   </head>
```

```
  <body>
    <script type="text/javascript">
      <!--
        document.write("Hello World")
      //-->
    </script>

    <input type="button" onclick="sayHello()" value="Say Hello" />

  </body>
</html>
```

11) JavaScript in External File

As you begin to work more extensively with JavaScript, you will be likely to find that there are cases where you are reusing identical JavaScript code on multiple pages of a site.

You are not restricted to be maintaining identical code in multiple HTML files. The **script** tag provides a mechanism to allow you to store JavaScript in an external file and then include it into your HTML files.

```
<html>

  <head>
    <script type="text/javascript" src="filename.js" ></script>
  </head>

  <body>
```

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

To use JavaScript from an external file source, you need to write all your JavaScript source code in a simple text file with the extension ".js" and then include that file as shown above.

you can keep the following content in **filename.js** file and then you can use **sayHello** function in your HTML file after including the filename.js file.

```
function sayHello() {
  alert("Hello World")
}
```

12) What is JavaScript Datatypes

JavaScript allows you to work with three primitive data types –

- **Numbers,** eg. 123, 120.50 etc.
- **Strings** of text e.g. "This text string" etc.
- **Boolean** e.g. true or false.

JavaScript also defines two trivial data types, **null** and **undefined,** each of which defines only a single value.

13) What is JavaScript Variables

JavaScript has variables. Variables can be thought of as named containers. You can place data into these containers and then refer to the data simply by naming the container.

Before you use a variable in a JavaScript program, you must declare it. Variables are declared with the **var** keyword as follows.

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

```
   var money;
   var name;
  //-->
</script>
```

You can also declare multiple variables with the same **var** keyword as follows –

```
<script type="text/javascript">
  <!--
   var money, name;
  //-->
</script>
```

Storing a value in a variable is called **variable initialization**. You can do variable initialization at the time of variable creation or at a later point in time when you need that variable.

14) How to display output in javascript

JavaScript can "display" data in different ways:

- Writing into an HTML element, using **innerHTML**.
    - To access an HTML element, JavaScript can use the **document.getElementById(id)** method.
    - The **id** attribute defines the HTML element. The **innerHTML** property defines the HTML content:

Ex:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
```

```
<p>My First Paragraph</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```

15) Writing into the HTML output using document.write().

- For testing purposes, it is convenient to use **document.write()**:

Ex:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

Note: Using document.write() after an HTML document is fully loaded, will **delete all existing HTML**:

16) Writing into an alert box, using window.alert().

- You can use an alert box to display data:

Ex:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

17) Writing into the browser console, using console.log().

- For debugging purposes, you can use the **console.log()** method to display data.

Ex:

```
<!DOCTYPE html>
<html>
<body>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

18)What are javascript operators?

Operator	Description
+	Addition
-	Subtraction
*	Multiplication

/	Division
%	Modulus
++	Increment
--	Decrement

19) What are JavaScript Assignment Operators?

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

Example

```
<!DOCTYPE html>
<html>
<body>

<h2>The += Operator</h2>

<p id="demo"></p>

<script>
var x = 10;
```

```
x += 5;

document.getElementById("demo").innerHTML = x;

</script>


</body>

</html>
```

20) What are JavaScript String Operators?

The + operator can also be used to add (concatenate) strings.

Example

```
txt1 = "John";

txt2 = "Doe";

txt3 = txt1 + " " + txt2;
```

The result of txt3 will be:

John Doe

21) What are Adding Strings and Numbers?

```
x = 5 + 5;
y = "5" + 5;
z = "Hello" + 5;
```

The result of *x*, *y*, and *z* will be:

```
10
55
Hello5
```

22) What are JavaScript Comparison Operators?

Operator Description

== equal to

=== equal value and equal type

!= not equal

!== not equal value or not equal type

> greater than

< less than

>= greater than or equal to

<= less than or equal to

? ternary operator

23) What are JavaScript Logical Operators?

Operator Description

&& logical and

|| logical or

! logical not

24) What are JavaScript Type Operators?

Operator Description

typeofReturns the type of a variable

instanceof Returns true if an object is an instance of an object type

25) What are JavaScript Bitwise Operators?

Bit operators work on 32 bits numbers.

Any numeric operand in the operation is converted into a 32 bit number. The result is converted back to a JavaScript number.

Operator Description Example Same as ResultDecimal

&	AND	5 & 1	0101 & 0001	0001	1
|	OR	5 | 1	0101 | 0001	0101	5
~	NOT	~ 5	~0101	1010	10
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	Zero fill left shift	5 << 1	0101 << 1	1010	10
>>	Signed right shift	5 >> 1	0101 >> 1	0010	2
>>>	Zero fill right shift	5 >>> 1	0101 >>> 1	0010	2

26) What are JavaScript Arithmetic Operators?

Arithmetic operators perform arithmetic on numbers (literals or variables).

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Increment
--	Decrement

27) Example of Arithmetic operators?

```
<!DOCTYPE html>
<html>
<body>
```

```
<p>A typical arithmetic operation takes two numbers and produces a new number.</p>


<p id="demo"></p>


<script>

var x = 100 + 50;

document.getElementById("demo").innerHTML = x;

</script>


</body>

</html>
```

28) What is Operators and Operands?

The numbers (in an arithmetic operation) are called operands.

The operation (to be performed between the two operands) is defined by an operator.

Operand	Operator	Operand
100	+	50

29) What is Operator Precedence?

The numbers (in an arithmetic operation) are called operands.

The operation (to be performed between the two operands) is defined by an operator.

Operand	Operator	Operand
100	+	50

Is the result of example above the same as 150 * 3, or is it the same as 100 + 150.

30) What JavaScript Operator Precedence Values?

Value	Operator	Description	Example
19	( )	Expression grouping	(3 + 4)
18	.	Member	person.name
18	[]	Member	person["name"]
17	()	Function call	myFunction()
17	new	Create	new Date()
16	++	Postfix Increment	i++
16	--	Postfix Decrement	i--
15	++	Prefix Increment	++i
15	--	Prefix Decrement	--i
15	!	Logical not	!(x==y)
15	typeof	Type	typeof x
14	*	Multiplication	10 * 5
14	/	Division	10 / 5
14	%	Modulo division	10 % 5
14	**	Exponentiation	10 ** 2

13	+	Addition	10 + 5
13	-	Subtraction	10 - 5
12	<<	Shift left	x << 2
12	>>	Shift right	x >> 2
12	>>>	Shift right (unsigned)	x >>> 2
11	<	Less than	x < y
11	<=	Less than or equal	x <= y
11	>	Greater than	x > y
11	>=	Greater than or equal	x >= y
10	==	Equal	x == y
10	===	Strict equal	x === y
10	!=	Unequal	x != y
10	!==	Strict unequal	x !== y
6	&&	Logical and	x && y
5	||	Logical or	x || y
3	=	Assignment	x = y
3	+=	Assignment	x += y

3	-=	Assignment	x -= y
3	*=	Assignment	x *= y
3	%=	Assignment	x %= y
3	<<=	Assignment	x <<= y
3	>>=	Assignment	x >>= y
3	>>>=	Assignment	x >>>= y
3	&=	Assignment	x &= y
3	^=	Assignment	x ^= y
3	|=	Assignment	x |= y