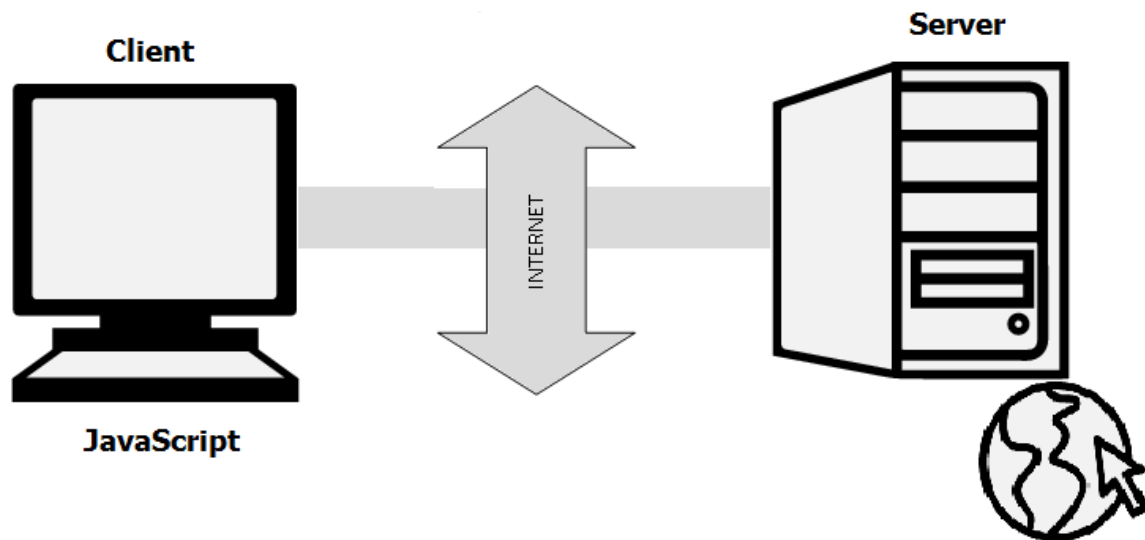


UNIT-III

Introduction to JavaScript

JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and [Mobile](#) application development.



We can use it to check or modify the contents of forms, change images, open new windows and write dynamic page content. You can even use it with CSS to make DHTML (Dynamic HyperTextMarkup Language).

History

JavaScript was developed by Brendan Eich in 1995, which appeared in Netscape, a popular browser of that time. The language was initially called “LiveScript” and was later renamed JavaScript. There are many programmers who think that JavaScript and [Java](#) are the same. In fact, **JavaScript and Java are very much unrelated. Java is a very complex programming language whereas JavaScript is only a scripting language.** The syntax of JavaScript is mostly influenced by the programming language C.

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JAVASCRIPT VS JAVA

A major difference **Java** and JavaScript is that Java is compiled and interpreted language while **JavaScript** code is directly executed by the browser

Parameters	Java	JavaScript
Variable Definition	Java is a strongly typed language, so the variable should be declared first before using in the program.	JavaScript is a weakly typed language, so its variable can be declared where they are used.
Type of language	It is an object-oriented programming language.	It is an object-based scripting language
Type of object	Objects of Java are class-based, so you can't create any program in java without developing a class.	Objects are prototype-based.
Extension	It has a file extension “.Java”.	It has file extension “.js”
Compilation process	It is interpreted as well as complied. Java translates source code into bytecodes. It is executed by JVM(Java Virtual Machine).	All browser has the JavaScript interpreter, which allows you to execute JavaScript code.
Process	Compiled on the server before execution on the client.	Interpreted (not compiled) by the client.
Code type	Object-oriented. Applets consist of object classes with inheritance.	It is object-based. Code uses built-in, extensible objects but not uses any classes or inheritance.
Syntax	Data types must be declared.	Data types not declared.
Type of language	Static	Dynamic
Key Features	<ul style="list-style-type: none">• Great libraries• Widely used• Excellent tooling	<ul style="list-style-type: none">• Can be used on frontend/backend• It's everywhere• Lots of great frameworks

Code	<pre> class A { public static void main(String args[]){ System.out.println("Hello World"); } } </pre>	<pre> <html> <head> <title>My First JavaScript code!!!</title> <script> alert("Hello World!"); </script> </head> <body> </body> </html> </pre>
------	---	---

JavaScript Syntax

JavaScript can be implemented using JavaScript statements that are placed within the **<script>... </script>**.

We can place the **<script>** tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that we should keep it within the **<head>** tags.

The **<script>** tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

Syntax:

```
<script ...>
```

```
Statement-1;
```

```
Statement-2;
```

```
Statement-n;
```

```
</script>
```

(or)

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

```
Statement-1;
```

```
Statement-2;
```

```
Statement-n;
```

```
</script>
```

The script tag takes two important attributes –

- **Language** – This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
- **Type** – This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

Example:

```
<!doctype html>

<html>

<head>

<scripttype="text/javascript" language="javascript">

    document.write("Welcome to Javascript");

</script>

</head>

</html>
```

General Syntactic Characteristics

- All JavaScript scripts will be embedded in HTML documents
 - Either directly, as in

```
<script type = "text/javaScript">
-- JavaScript Statements -
</script>
```
 - Or indirectly, as a **“.js”** file specified in the src attribute of <script>, as in

```
<script type = "text/javaScript"
    src = "myScript.js">
</script>
```
- Language Basics:
 - Identifier form: begin with a letter or underscore, followed by any number of letters, underscores, and digits

□

25 reserved words, plus future reserved words

- Comments: both // and /* ... */
- Scripts are usually hidden from browsers that do not include JavaScript interpreters by putting them in special comments

```
<!--  
-- JavaScript script -  
//-->
```

 - Also hides it from HTML validators
- Semicolons can be a problem
 - They are "somewhat" optional

JavaScript - Placement in HTML File

There is a flexibility given to include JavaScript code anywhere in an HTML document. However the most preferred ways to include JavaScript in an HTML file are as follows –

- Script in <head>...</head> section.
- Script in <body>...</body> section.
- Script in <body>...</body> and <head>...</head> sections.
- Script in an external file and then include in <head>...</head> section.

In the following section, we will see how we can place JavaScript in an HTML file in different ways.

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

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

```
<html>

<head>

<script type="text/javascript">

<!--functionsayHello() { alert("Hello World")
    }

    //-->

</script>

</head>

<body>

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

</body>

</html>
```

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

If we 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.

Example: 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>
```

JavaScript in <body> and <head> Sections

We 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>
```


JavaScript in External File

As we 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.

The **script** tag provides a mechanism to allow you to store JavaScript in an external file and then include it into your HTML files.

Here is an example to show how we can include an external JavaScript file in our HTML code using “**script** tag” and its “**src**” attribute.

External.html

```
<html>

<head>

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

</head>

<body>

.....

</body>

</html>
```

Script.js

```
functionsayHello()

{

alert("Hello World")

}
```