1. OVERVIEW

What is JavaScript?

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 was first known as **LiveScript**, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The ECMA-262 Specification defined a standard version of the core JavaScript language.

- 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.

Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Advantages of JavaScript

The merits of using JavaScript are:

- **Less server interaction:** You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors:** They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity:** You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces:** You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features:

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multithreading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

JavaScript Development Tools

One of major strengths of JavaScript is that it does not require expensive development tools. You can start with a simple text editor such as Notepad. Since it is an interpreted language inside the context of a web browser, you don't even need to buy a compiler.

To make our life simpler, various vendors have come up with very nice JavaScript editing tools. Some of them are listed here:

- **Microsoft FrontPage:** Microsoft has developed a popular HTML editor called FrontPage. FrontPage also provides web developers with a number of JavaScript tools to assist in the creation of interactive websites.
- **Macromedia Dreamweaver MX:** Macromedia Dreamweaver MX is a very popular HTML and JavaScript editor in the professional web development crowd. It provides several handy prebuilt JavaScript

components, integrates well with databases, and conforms to new standards such as XHTML and XML.

- **Macromedia HomeSite 5:** HomeSite 5 is a well-liked HTML and JavaScript editor from Macromedia that can be used to manage personal websites effectively.

Where is JavaScript Today?

The ECMAScript Edition 5 standard will be the first update to be released in over four years. JavaScript 2.0 conforms to Edition 5 of the ECMAScript standard, and the difference between the two is extremely minor.

The specification for JavaScript 2.0 can be found on the following site: http://www.ecmascript.org/

Today, Netscape's JavaScript and Microsoft's JScript conform to the ECMAScript standard, although both the languages still support the features that are not a part of the standard.

Enable | Disable Javascript from browser

✓ Google chrome

 → Settings → Privacy & Security → Site setting → Javascript.

Tripple dot ✓
Top right ✓

Internet Explorer ✓
Firefox ✓
Opera ✓

No script tag

If a browser does not support JavaScript or JavaScript is not enabled, then message from <noscript> will be displayed.

```
<html>
<body>
<script language="javascript" type="text/javascript">
<!--
 document.write ("Hello World!")
//-->
<noscript>
 Sorry...JavaScript is needed to go ahead.
</noscript>
</body>
</html>
```

JAVASCRIPT

- A scripting language
- create dynamic content | Add interactivity
- Multi-paradigm language
- Syntax is based on Java

<script> tag ✓

Display in Javascript

OUTPUT

- document.write()
- inner HTML
- window.alert()
- console.log()
- window.print()

eg.
```
<script>
    document.write("Hello world");
</script>
```

Use external file ✓

myjs.js
```
document.write("Hello word");
```

myprogs.html
```
<html>
<body>
    <script src="myjs.js">
    </script>
</body>
</html>
```

innerHTML property of a ~~paragraph~~ element

<P> "Hello, I am here </P> → static content.

<P id = "hey"> Hello, I am here </P>

<script src = "myjs.js"> </script>

myjs.js

~~let paragraph = document.getElementById~~

```
let paragraph = document.getElementById ("hey");
paragraph.innerHTML = " I am in INDIA";
```

Content is made dynamic ∴ dynamic content

eg. Newspaper headline

~~document.write() can be used to flush existing HTML.~~

~~<button type = "button" onclick = "document.write("Hello")">~~
~~Try it </button>~~

document.write() can be used to write a full HTML page. ✓

<script> document.write ("<h2> Hello world </h2> <P> New </P>") </script>.

Write HTML through document.write on click ✓

<button type = "button" onclick = " document.write(. .)" </button>

Alternately onclick = "myfunc()"

```
<script>
myfunc()
{
  document.write( );
}
</script>
```

Write HTML in new window ✓

Alert box

```
window.alert("Hello");
```

on Console

```
console.log("Hello");
```

A window onto real prompter

INPUT ✓ using prompt → ✓

DECISION + SWITCH, LOOP → ✓

Taking input from HTML elements ✓