**DAY 8- JavaScript**

https://lh6.googleusercontent.com/pzfkpYzj_xNKuCIRaMdGOcJRGHimowNa5u7Ned4rI9rrL8AGediPxVgoTzYM65B3F7T2wKnzzs4NN42aMrsGf_ZZvSDkx9V2RHZ5SJ6YjjldEA7E9x9SwUSXZmLWBKV9rfHuNRbkG3bXpFexSg

**Refresh yesterday’s Class-10 Minutes**

1. Which of the following is used to sets an icon inside the input?
2. background-image: search.jpg;
3. background-image: url=”search.jpg”;
4. background-image: url('search.jpg');

**Answer: -** c

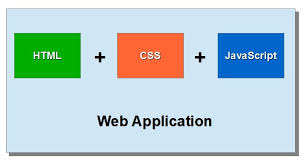
1. What is the use of padding property Input fields?
2. add space inside the text field
3. add space between the text field
4. add space outside the text field

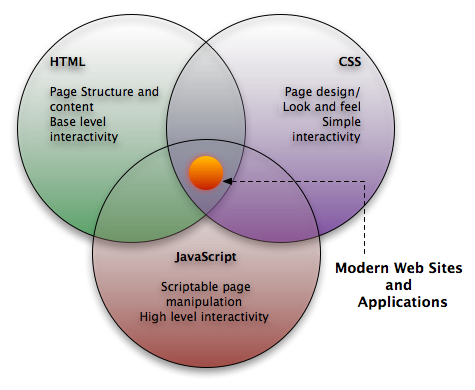
**Answer: -** a

1. If you only want to style a text input type, which of the following is used?
2. input[type=text]{ }
3. input[type=”text”]{ }
4. input{ }

**Answer: -** a

**JavaScript Overview-15 Minutes**





**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**.

**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.

The **<script>** tag is used to define a client-side script (JavaScript). The <script> element either contains scripting statements, or it points to an external script file through the **src** attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content. To select an HTML element, JavaScript very often use the **document.getElementById(id)** method.

<!DOCTYPE html>

<html>

<body>

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

<script>

document.getElementById("demo").innerHTML = "Hello JavaScript!";

</script>

</body>

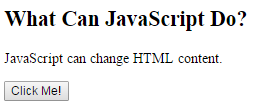
</html>

The above example writes "Hello JavaScript!" into an HTML element with id="demo".

**(Give students 10 minutes to work on the above example)**

**What can JavaScript do? -20 Minutes**

* **JavaScript can Change HTML content**



<!DOCTYPE html>

<html>

<body>

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

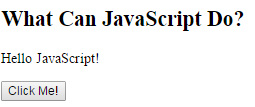
<p id="demo">JavaScript can change HTML content.</p>

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

</body>

</html>

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



* **JavaScript can Change HTML Style**

Changing the style of an HTML element, is a variant of changing an HTML attribute.

document.getElementById("demo").style.fontSize = "25px";  
or  
document.getElementById('demo').style.fontSize = '25px';

* **JavaScript can Hide HTML Element**

Hiding HTML elements can be done by changing the display style.

document.getElementById("demo").style.display = "none";  
or  
document.getElementById('demo').style.display = 'none';

* **JavaScript can Show HTML Element**

Showing hidden HTML elements can also be done by changing the display style.

document.getElementById("demo").style.display = "block";  
or  
document.getElementById('demo').style.display = 'block';

**Activity- 20 Minutes**



<!DOCTYPE html>

<html>

<body>

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

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

<p id="demo1">Hello JavaScript</p>

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

<button type="button" onclick="document.getElementById('demo1').style.fontSize='35px'">Style Me!</button>

<button type="button" onclick="document.getElementById('demo1').style.display='none'">Hide Me!</button>

<button type="button" onclick="document.getElementById('demo1').style.display='block'">Show Me!</button>

</body>

</html>

**JavaScript Outputs-15 Minutes**

JavaScript can "display" data in different ways:

* Writing into an HTML element, using **innerHTML**.
* Writing into the HTML output using **document.write()**.
* Writing into an alert box, using **window.alert()**.
* **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.

<!DOCTYPE html>

<html>

<body>

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

<script>

document.getElementById("demo").innerHTML = 5 + 6;

</script>

</body>

</html>

Changing the innerHTML property of an HTML element is a common way to display data in HTML.

## Using document.write()

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

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

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



<!DOCTYPE html>

<html>

<body>

<h2>My First Web Page</h2>

<p>My first paragraph.</p>

<button type="button" onclick="document.write(15 + 6)">Try it</button>

</body>

</html>

The document.write() method should only be used for testing.

## Using window.alert()

You can use an alert box to display data

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

**Activity- 30 Minutes**

<!DOCTYPE html>

<html>

<body>

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

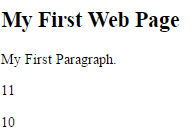
<script>

document.getElementById("demo").innerHTML = 5 + 6;

</script>

</body>

</html>



<!DOCTYPE html>

<html>

<body>

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

<script>

document.getElementById("demo").innerHTML = 5 + 6;

document.write(5 + 5);

</script>

</body>

</html>



<!DOCTYPE html>

<html>

<body>

<h2>My First Web Page</h2>

<p>My first paragraph.</p>

<button type="button" onclick="document.write(15 + 6)">Try it</button>

</body>

</html>



<!DOCTYPE html>

<html>

<body>

<h2>My First Web Page</h2>

<p>My first paragraph.</p>

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

<script>

document.write(5 + 6);

document.getElementById("demo").innerHTML = 5 + 5;

window.alert(1 + 6);

window.alert("Welcome");

</script>

<br>

<button type="button" onclick="document.write(15 + 6)">Try it</button>

</body>

</html>

**JavaScript-Placement in HTML file-20 Minutes**

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.
* **Script 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.

<head>

<script type="text/javascript">

function sayHello() {

alert("Hello World")

}

</script>

</head>

<body>

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

</body>

* **Script 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.

<body>

<script type="text/javascript">

document.write("Hello World")

</script>

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

</body>

* **Script in <body>...</body> and <head>...</head> sections**

You can put your JavaScript code in <head> and <body> section altogether.

<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>

* **Script in an external file and then include in <head>...</head> section**

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.

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

<head>

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

</head>

<body>

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

</body>

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.

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

**file.js**

function sayHello() {

alert("Hello World.................")

}

**Activity-30 Minutes**

<!DOCTYPE html>

<html>

<head>

<script type="text/javascript">

function sayHello() {

alert("Hello World")

}

</script>

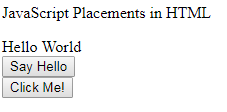
</head>

<body>

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

</body>

</html>



**Javascript.html**

<html>

<head>

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

<script type="text/javascript">

function sayHello()

{

alert("Hello World");

}

</script>

</head>

<body>

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

<script type="text/javascript">

document.write("Hello World")

document.getElementById("demo").innerHTML = "JavaScript Placements in HTML";

</script>

<br>

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

<br>

<input type="button" onclick="helloJavaScript()" value="Click Me!" />

</body>

</html>

**file.js**

function helloJavaScript() {

alert("Hello JavaScript.................")

document.write("Hai World...");

}