

This line is added in end in body of html unlike css which is added in head tag.

<html>

<head>

    <title>JS</title>

</head>

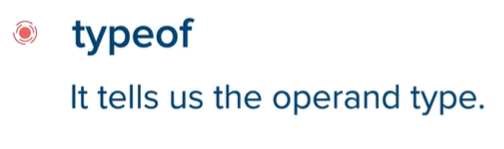
<body>

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

</body>

</html>

‘Let’ keyword has block level scope and ‘var’ keyword has global scope.



**For-of loop:**

let arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

*for* (let element of arr) {

    document.write(element + " ");

}

**For-in loop:**(used for key-value pairs)

Foreach in PHP is same as for-in in JS.

For-in is used exclusively for javascript objects.

let arr = { "George": 2001, "Barak": 2009, "Donald": 2017 };

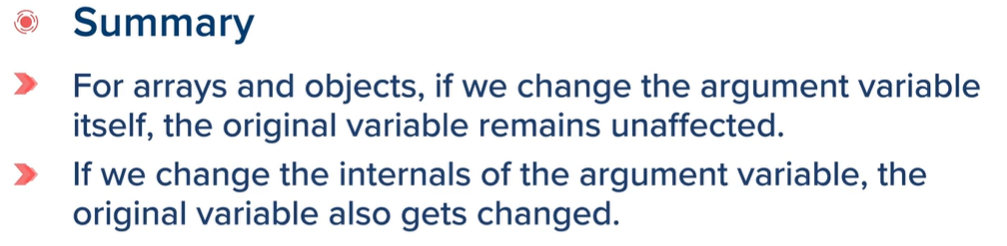
*for* (let name in arr) {

    document.write(name + " was elected in " + arr[name] + "<br>");

}

**Foreach** in JS is not a loop, it is a function. It is used to iterate through arrays and not through objects.

**Call by reference is not allowed in JS.**



Function -Call by Value approach does not change values:

**Except when elements of array are change one-by-one.**

function change(*a*) {

*a*[4] = 90;

*a*[3] = 1000;

}

let a = [20, 30, 40, 50, 60];

change(a);

*for* (let element of a) {

    document.write(element + " ");

}

**OUTPUT:**

**20 30 40 1000 90**

* **ANONYMOUS FUNCTIONS:**

Function is not given a name.

**NOTE multiply function below ends with a semi-colon.**

let multiply = function (*a*, *b*) {

*return* (*a* \* *b*);

};

let ans = multiply(2, 4);

document.write(ans);

**Function as an argument:**

let multiply = function (*a*, *b*) {

*return* (*a* \* *b*);

};

let add = function (*a*, *b*) {

*return* (*a* + *b*);

};

let a = 5, b = 20;

function action(*x*, *y*, *z*) {

    let r = z(*x*, *y*);

*return* r;

}

let e = action(a, b, multiply);

document.write(e);

**Arrow functions:**

let multiply = (*a*, *b*) => {

*return* (*a* \* *b*);

};

document.write(multiply(2, 4));

**OBJECTS:**

Dot (.) operator is used to access inside things of objects and can be used to change and create the existing and new things inside the objects as well.

Square Brackets can also be used but for accessing things the property(things) must be enclosed.

‘this’ refers to the class of which it is a part of.

let dog = {

    breed: 'Golden Retriever',

    height: '4ft',

    age: 2,

    display: function () {

        document.write(this.breed + this.height + this.age);

    }

};

document.write(dog.breed + " " + dog.height + " " + dog.age + "<br>");

dog.weight = '32kg';

document.write(dog.breed + " " + dog.height + " " + dog.age + " " + dog.weight + "<br>");

*//square brackets can also be used.*

document.write(dog['breed'] + " " + dog['height'] + " " + dog['age']);

**MAKING CLASS USING new Object() function:**

let dog = new Object();

dog.breed = 'Golden Retriever';

dog.height = '4ft';

dog.age = '2';

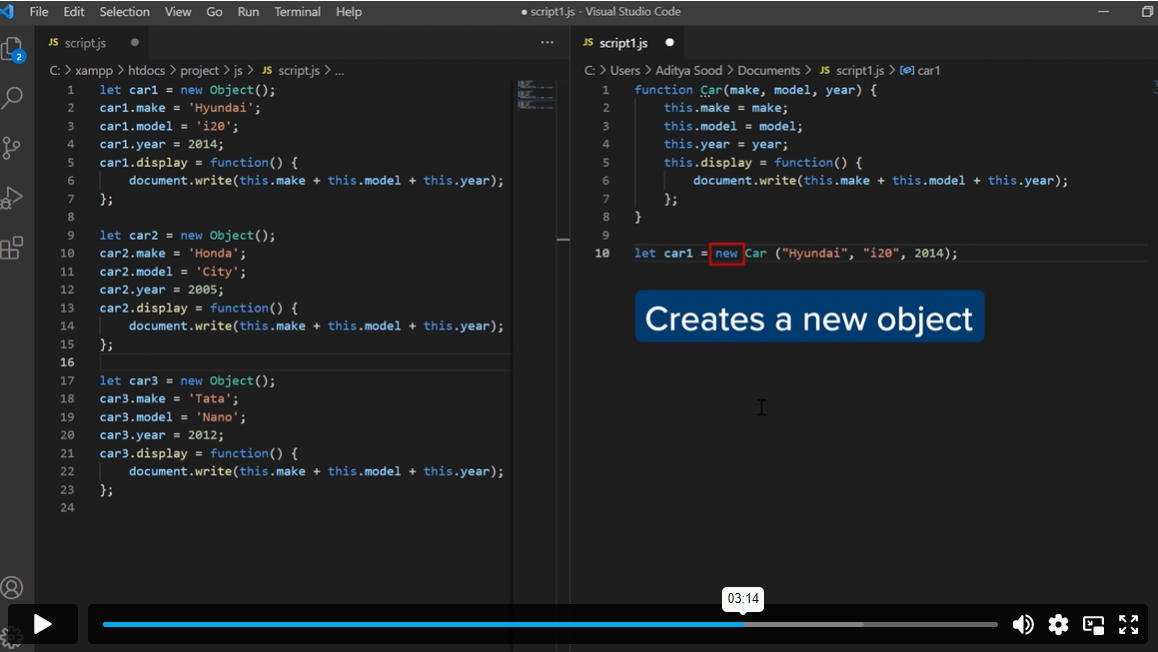
function display() {

    document.write(dog.breed + " " + dog.height + " " + dog.age);

}

display();

**Making class Using function constructor:**



**Used since different objects have same properties.**

**Note the use of ‘new’ keyword.**

function car(*make*, *model*, *year*) {

    this.make = *make*;

    this.model = *model*;

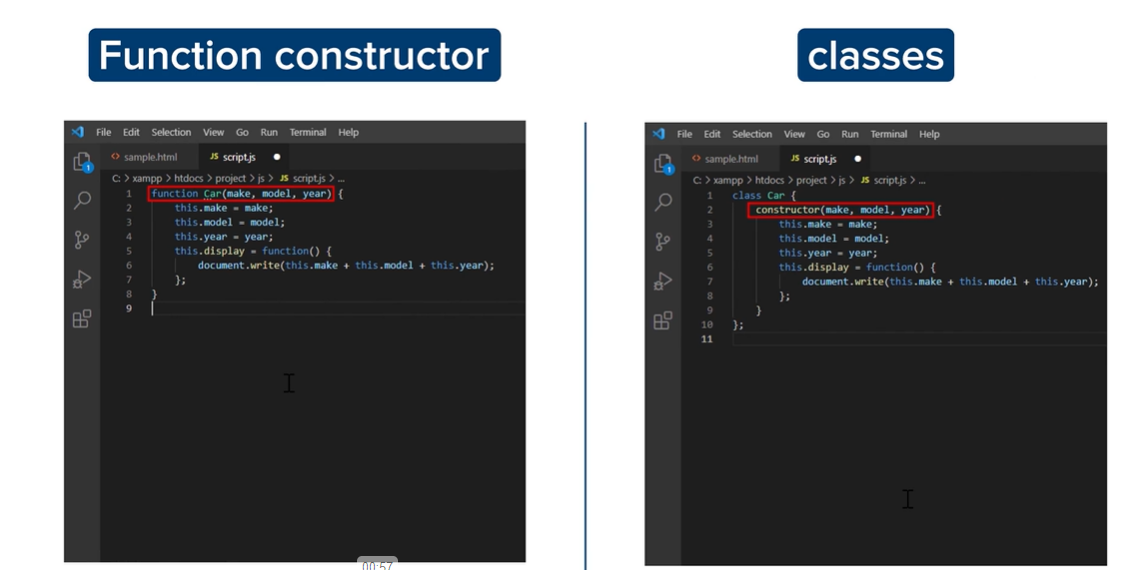
    this.year = *year*;

}

let car1 = new car("Maruti-Suzuki", "Ritz", "2012");

document.write(car1.make + " " + car1.model + " " + car1.year);

**Using Classes:**



class car {

    constructor(*make*, *model*, *year*) {

        this.make = *make*;

        this.model = *model*;

        this.year = *year*;

    }

};

let car1 = new car("Maruti-Suzuki", "Ritz", "2012");

document.write(car1.make + " " + car1.model + " " + car1.year);

**using for-in loop for printing each key:value pair:**

class car {

    constructor(*make*, *model*, *year*) {

        this.make = *make*;

        this.model = *model*;

        this.year = *year*;

    }

};

let car1 = new car("Maruti-Suzuki", "Ritz", 2012);

let car2 = new car("BMW", "BMW X5", 2019);

*for* (let prop in car1) {

    document.write(prop + ":" + car1[prop] + "<br>");

}

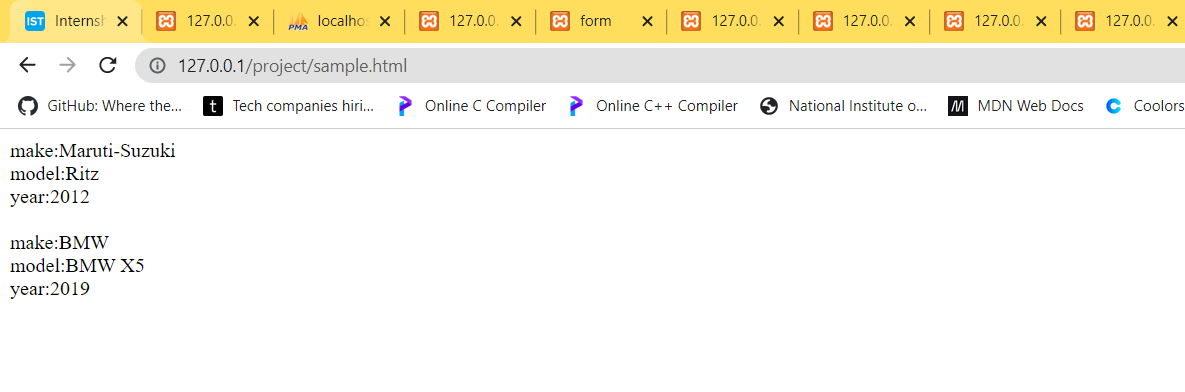
document.write("<br>");

*for* (let ele in car2) {

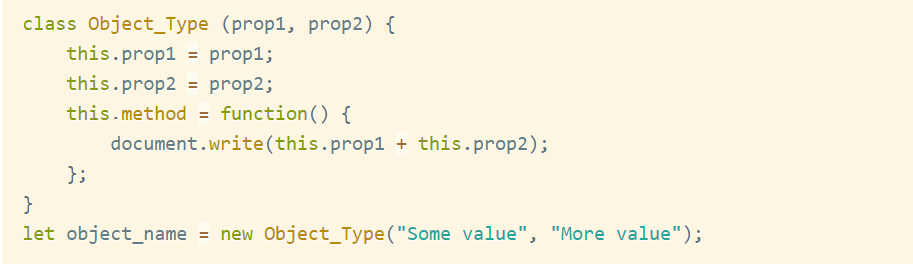
    document.write(ele + ":" + car2[ele] + "<br>");

}

**OUTPUT:**



**Another way of making object using classes:**



**ARRAYS:**

**Numeric indexed arrays:**

let modules = ["HTML", "CSS", "JAVASCRIPT", "PHP"];

*for* (let element of modules) {

    document.write(element + " ");

}

**ARRAYS as OBJECTS:**

We use new Array() function to make arrays.

let modules = new Array("HTML", "CSS", "JavaScript", "PHP");

*for* (let element of modules) {

    document.write(element + " ");

}

Numeric indexed arrays can be considered as Objects and Objects can be considered as numeric-indexed arrays.

**ARRAY METHODS:**

**PUSH() and POP() :**

let modules = new Array("HTML", "CSS", "DBMS");

*for* (let element of modules) {

    document.write(element + " ");

}

document.write("<br>");

modules.push("Bootstrap");

*for* (let element of modules) {

    document.write(element + " ");

}

document.write("<br>");

modules.push("PHP");

*for* (let element of modules) {

    document.write(element + " ");

}

document.write("<br>");

document.write(modules.pop());

document.write("<br>");

*for* (let element of modules) {

    document.write(element + " ");

}

**forEach() :** forEach is not a loop but it is used as a loop.

let nums = [2, 4, 6, 8, 10];

nums.forEach(function (*element*) {

    document.write(*element* + " ");

});

Converting to arrow function format:

let nums = [2, 4, 6, 8, 10];

nums.forEach((*element*) => {

    document.write(*element* + " ");

});

**map() :**

**It is specific to array objects.**

let nums = [1, 2, 3, 4, 5, 6];

let squares = nums.map(*x* => (*x* \* *x*));

squares.forEach(function (*element*) {

    document.write(*element* + " ");

});

**Filter() :**

let numbers = [1, 2, 3, 4, 5];

let even\_numbers = numbers.filter(*x* => *x* % 2 == 0);

even\_numbers.forEach(function (*element*) {

    document.write(*element* + " ");

})

**Rest parameter(…) :**

It is used so as to make a function take different amount of arguments.

function sum(...*arg*) {

    let sum = 0;

*arg*.forEach(function (*element*) {

        sum += *element*;

    });

*return* sum;

}

document.write(sum(1, 2, 3) + "<br>");

document.write(sum(1, 2, 3, 4) + "<br>");

We can take arguments ‘a’ and ‘b’ at the beginning and hence, we must start our sum with sum=(a+b) else everything except ‘a’ and ‘b’ will be added.

function sum(*a*, *b*, ...*arg*) {

    let sum = (*a* + *b*);

*arg*.forEach(function (*element*) {

        sum += *element*;

    });

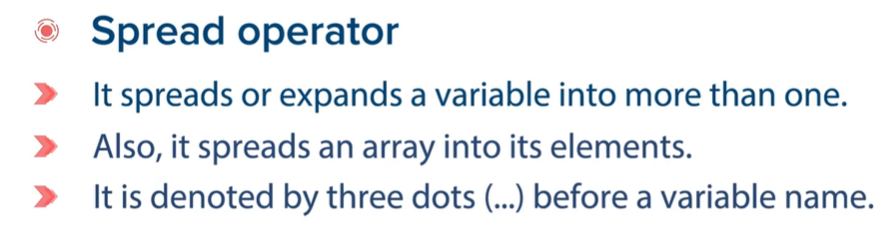
*return* sum;

}

document.write(sum(1, 2, 3) + "<br>");

document.write(sum(1, 2, 3, 4, 5) + "<br>");

**Spread operator(…) :**



let even = [2, 4, 6, 8];

let odd = [1, 3, 5, 7];

let numbers = [...even, ...odd];

numbers.forEach(function (*element*) {

    document.write(*element* + " ");

})

**Strings:**

**strings can be written using single, double quotes as well as backticks(`)**

let name = `apurv anand`;

document.write(name);

**Template literals: For using it , we must use backticks(`).**

let name = "Apurv Anand";

let sentence = `hello ${name}!`;

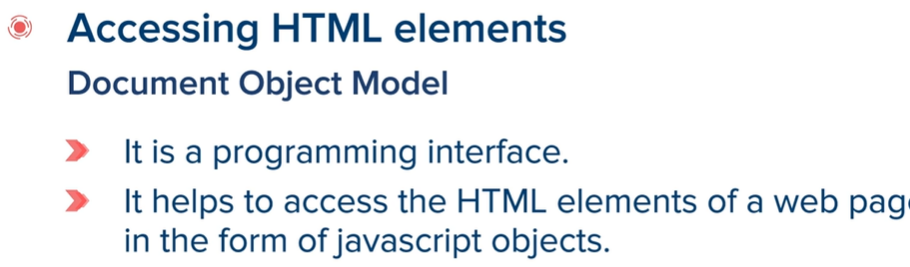
document.write(sentence);

**Strings as objects: use new String() function.**

let name = new String("Apurv Anand");

document.write(name);

**Java Script as Client-side-scripting-language:**

**DOM i.e. Document Object Model is used to access or change HTML elements of webpage even after they have been rendered by the browser.** 

**Html file:**

<html>

<head>

    <title>JS</title>

</head>

<body>

    <h1 id="h1">I am the h1 heading.</h1>

    <h2 id="h2">I am h2 heading.</h2>

    <p class="para">I am a paragraph. Lorem ipsum dolor sit amet consectetur adipisicing elit. Eum sunt quod rem vel

        officiis modi

        dolorem. Et alias libero dolorum, repellendus minima maiores, blanditiis laboriosam perferendis suscipit enim

        nisi eum.</p>

    <p class="para">I am second paragraph. Lorem ipsum dolor sit amet consectetur adipisicing elit. Eum sunt quod rem

        vel

        officiis modi

        dolorem. Et alias libero dolorum, repellendus minima maiores, blanditiis laboriosam perferendis suscipit enim

        nisi eum.</p>

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

</body>

</html>

**Corresponding JS file for showing different methods of DOM:**

let e1 = document.getELementById("h1");

let e2 = document.getElementById("h2");

let p = document.getElementsByClassName("para");

let z = document.getElementsByTagName("h1");

let q = document.querySelectorAll(".para"); *//.para => class para*

**Different properties of DOM:**

**innerHTML**

**It returns the content. Do not store it in any other variable , directly print it.**

**Sample.html**

<html>

<head>

    <title>JS</title>

</head>

<body>

    <h1 id="hd1">I am the h1 heading.</h1>

    <h2 id="hd2">I am h2 heading.</h2>

    <p class="para">I am a paragraph. Lorem ipsum dolor sit amet consectetur adipisicing elit. Eum sunt quod rem vel

        officiis modi

        dolorem. Et alias libero dolorum, repellendus minima maiores, blanditiis laboriosam perferendis suscipit enim

        nisi eum.</p>

    <p class="para">I am second paragraph. Lorem ipsum dolor sit amet consectetur adipisicing elit. Eum sunt quod rem

        vel

        officiis modi

        dolorem. Et alias libero dolorum, repellendus minima maiores, blanditiis laboriosam perferendis suscipit enim

        nisi eum.</p>

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

</body>

</html>

**Using innerhtml in script.js**

document.write(document.getElementById("hd1").innerHTML);

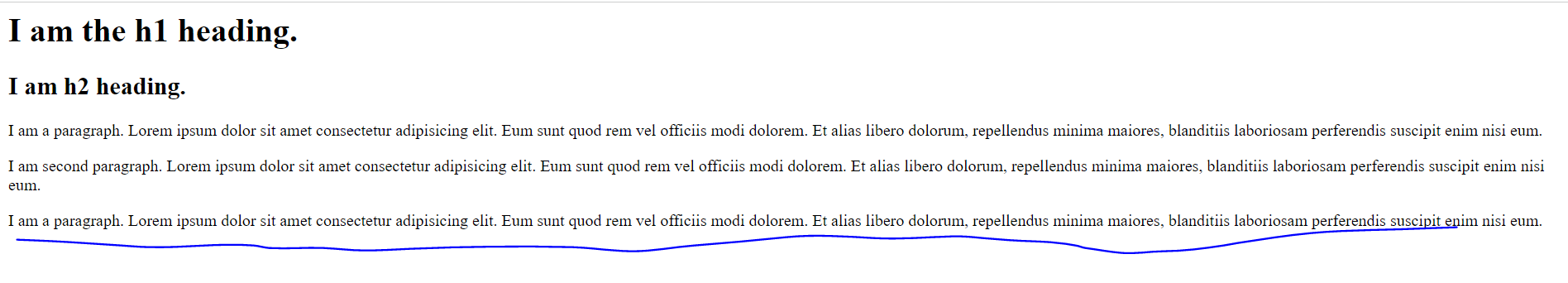
**using alert():**

alert(document.getElementById("hd1").innerHTML);

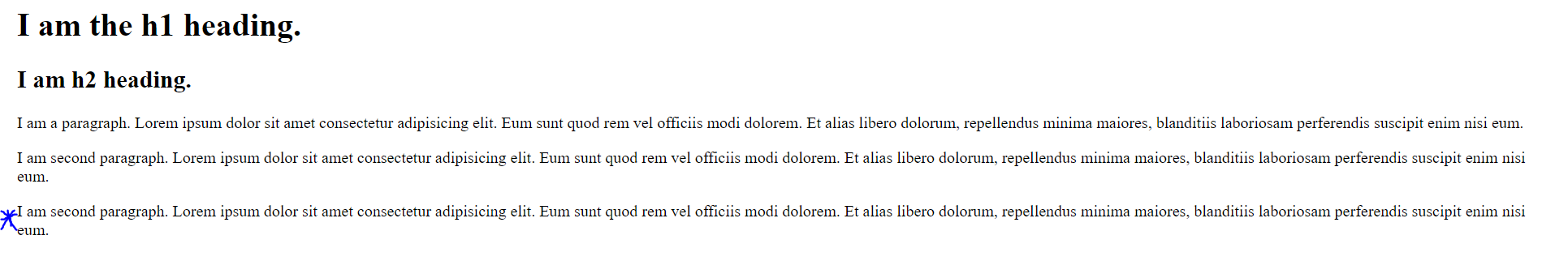
**in getElementByClassName: it returns an array.**

document.write(document.getElementsByClassName("para")[0].innerHTML);

**notice [0] written to access first element of array containing the “para”.**



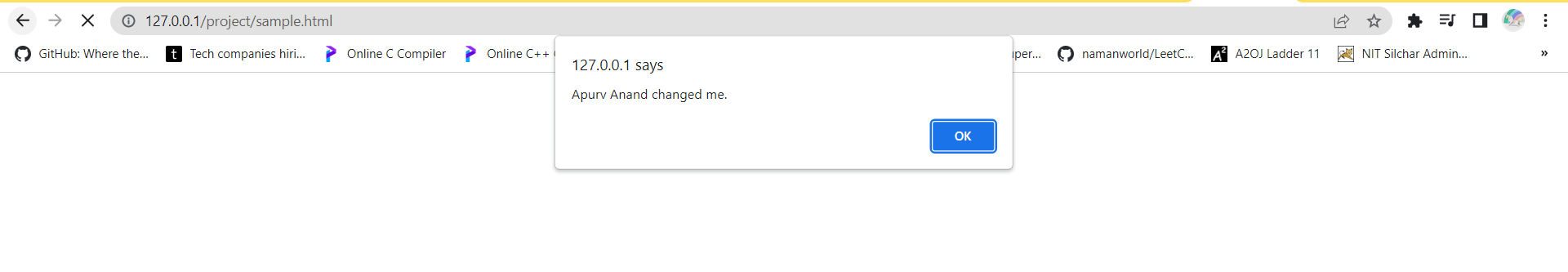
document.write(document.getElementsByClassName("para")[1].innerHTML);

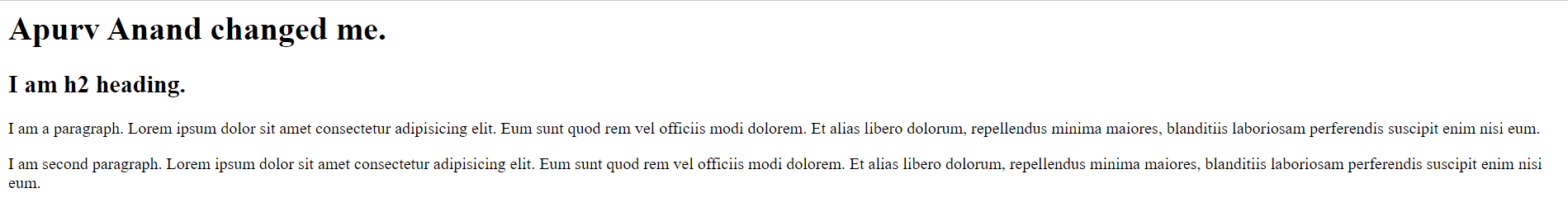


Inner HTML can be used to change contents of HTML page as well.

document.getElementById("hd1").innerHTML = "Apurv Anand changed me.";

alert(document.getElementById("hd1").innerHTML);



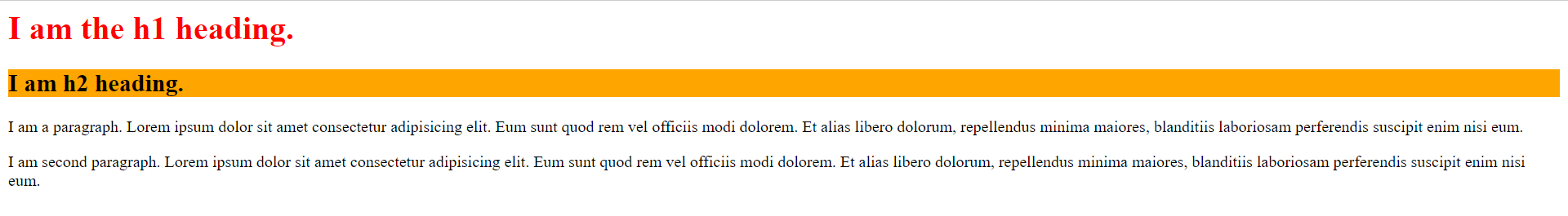


Heading is changed.

**Using style property:** using style property we can change css properties.

document.getElementById("hd1").style.color = "red";

document.getElementById("hd2").style.backgroundColor = "orange";

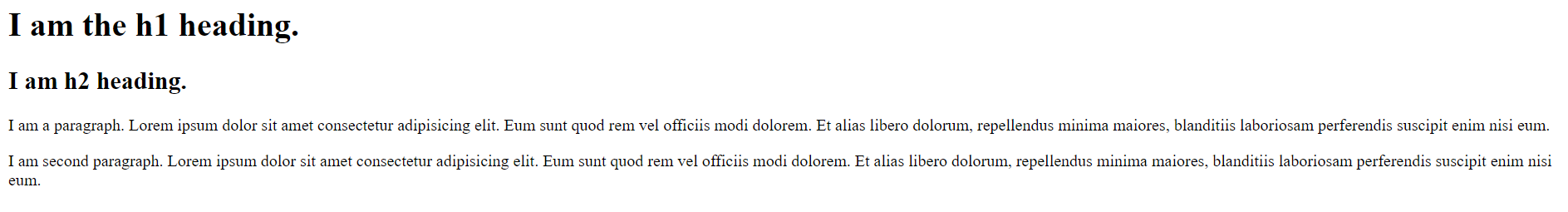


**Using Event handlers:OnClick:**

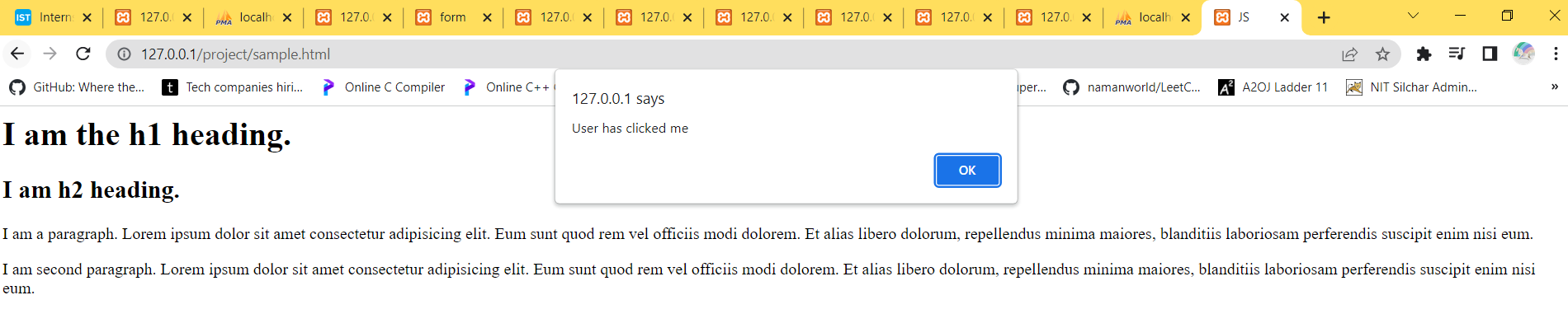
document.getElementById('hd1').onclick = function () {

    alert("User has clicked me");

};



**On clicking ‘I am the h1 heading.’**



document.getElementById("hd1").innerHTML = "I changed.";

document.getElementById("hd1").onclick = function () {

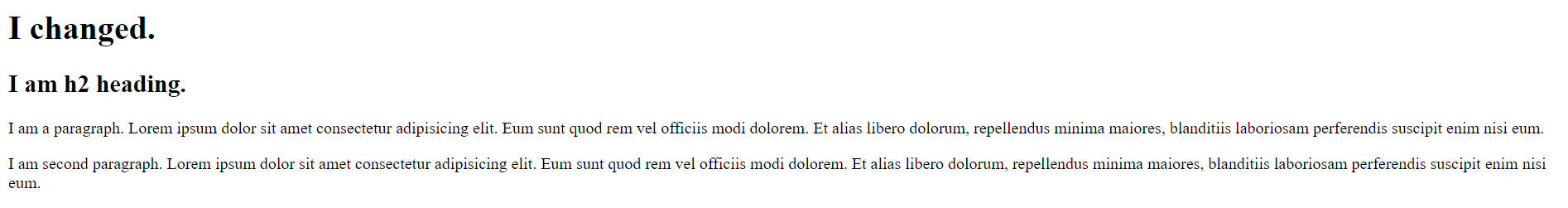
    document.getElementById("hd1").style.color = "red";

    document.getElementById("hd1").style.backgroundColor = "yellow";

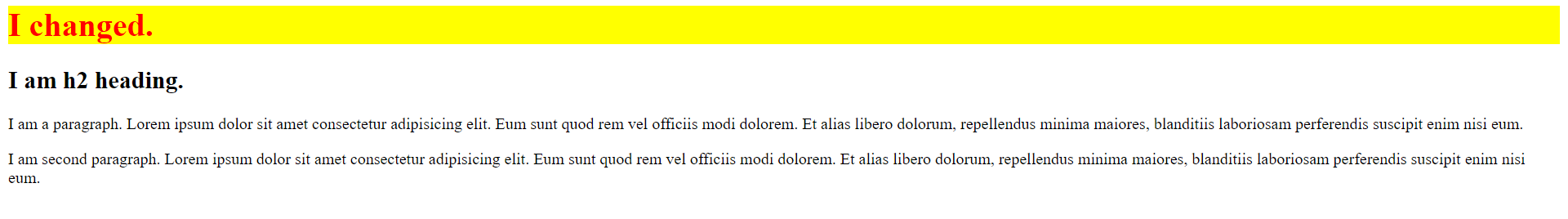
}

**OUTPUT-**

**INITIALLY:**



**On clicking ‘I changed’ :**



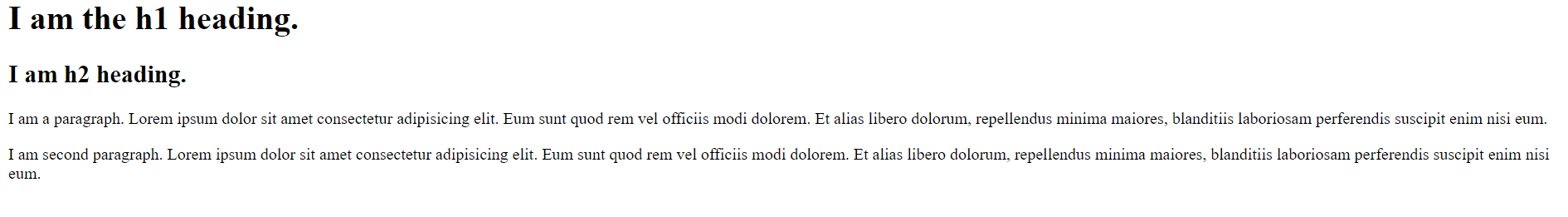
document.getElementById("hd1").onclick = function () {

    document.getElementById("hd1").innerHTML = "You Clicked ME!";

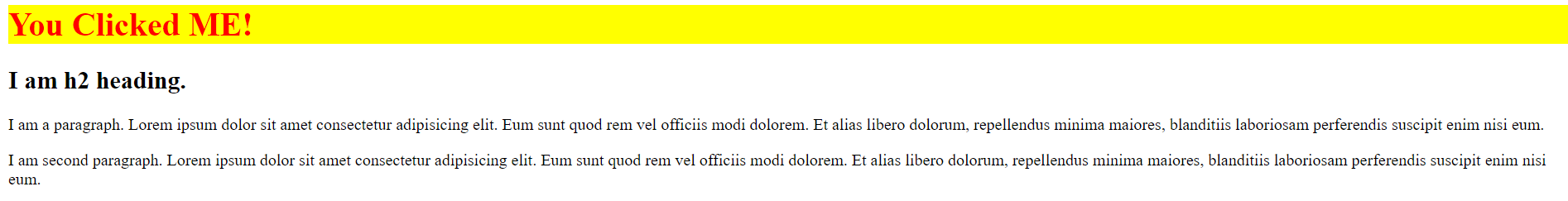
    document.getElementById("hd1").style.color = "red";

    document.getElementById("hd1").style.backgroundColor = "yellow";

}

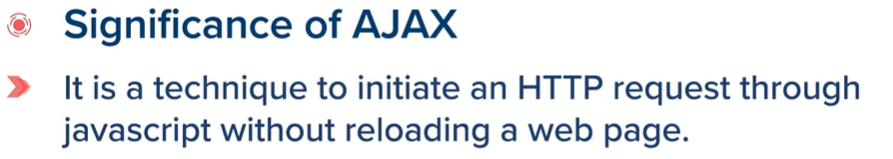


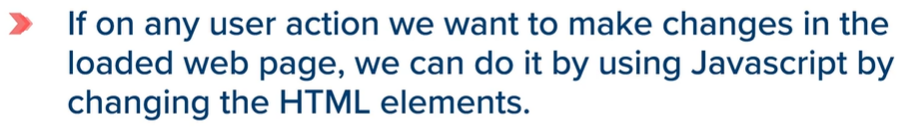
**On clicking:**

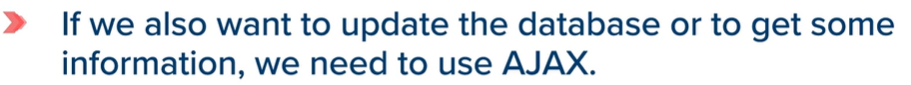


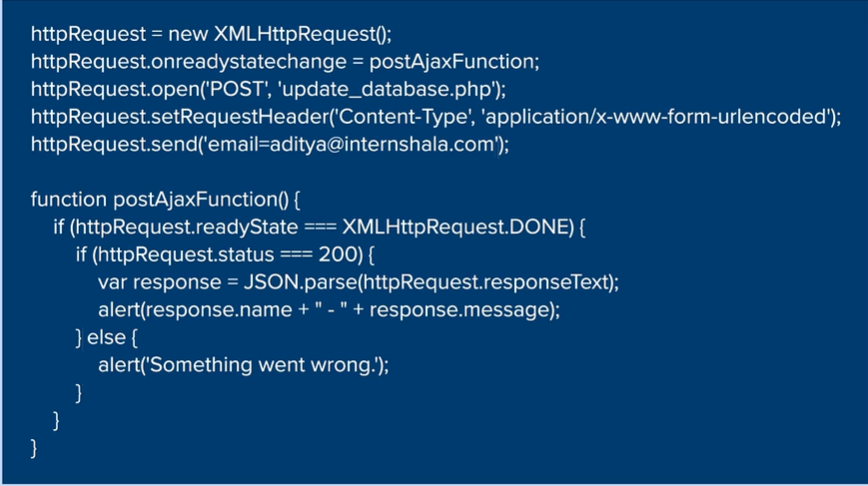
**AJAX( Asynchronous Java Script And XML)**

It is a combination of java script and XML.

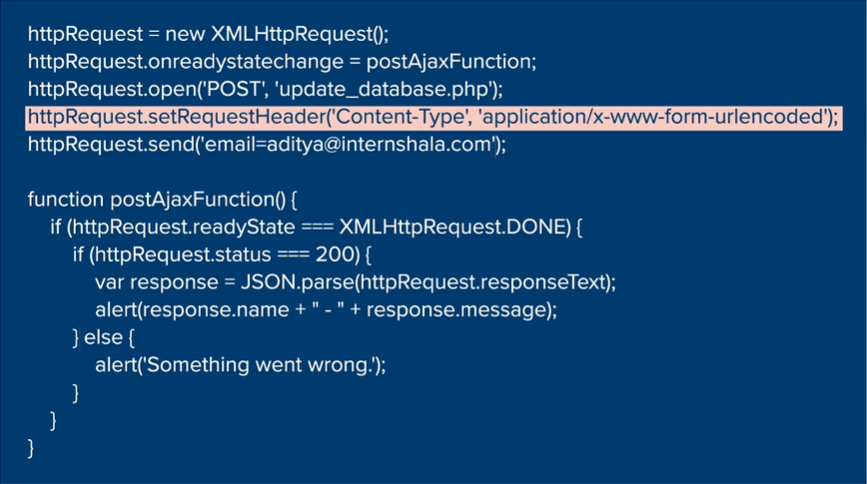








**If we are using ‘GET’ type of request , then , we must remove the next coloured line.**





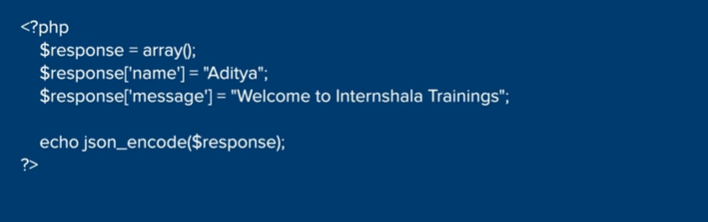
To add more arguments :



The data returned by the above code has to be in form of an associative array , means in form of key: value pairs.

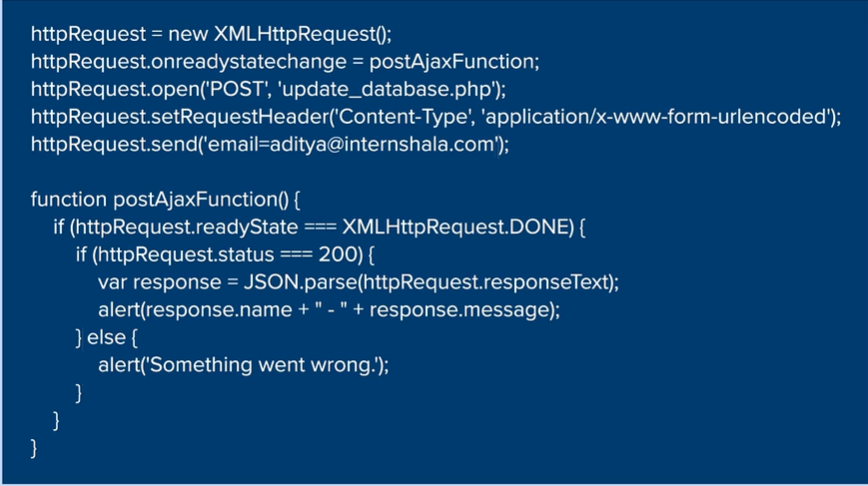
The code below stores in form of Key: value pairs.

In last line of the code below, **json\_encode is used to stringify or convert to string , the data which was entered above.** And hence, using echo , we are able to print that string.



So, here in the above code, we have converted the key:value paired array into string.

But



Later on, when out httpRequest.status===200 is true , then, for var response , JSON.parse() converts the string back to key:value pairs.

This line was being talked about above.



**THE SAME CODE AS ABOVE:**

**Script.js**

"use strict";  *//for using only 'let' and no 'var' in the entire code.*

let httpRequest = new XMLHttpRequest();

let el = document.getElementById("abc");

el.onclick = function () {

    el.innerHTML = "This is some new content";

    el.style.color = "red";

    httpRequest.onreadystatechange = postAjaxFunction;

    httpRequest.open('POST', 'update\_database.php');

    httpRequest.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');

    httpRequest.send('email=anuj@internshala.com&id=4');

}

function postAjaxFunction() {

*if* (httpRequest.readyState === XMLHttpRequest.DONE) {

*if* (httpRequest.status === 200) {

            var response = JSON.parse(httpRequest.responseText);

            alert(response.name + " - " + response.message);

        } *else* {

            alert('Something went wrong.');

        }

    }

}

**Update\_database.php**

<?php

    $response = array();

    $response['name'] = "Aditya";

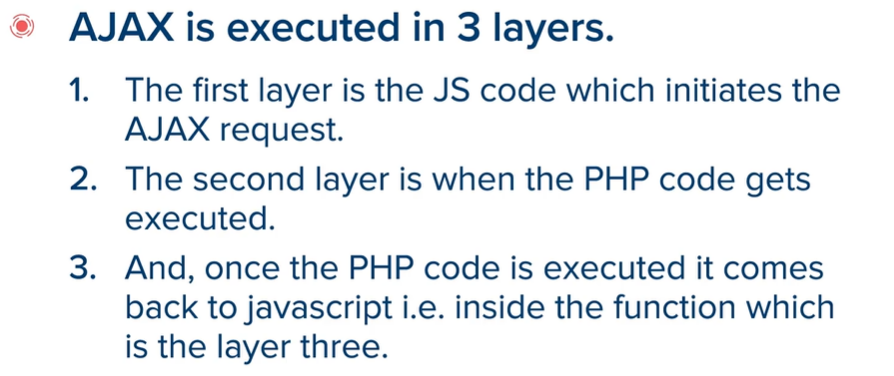
    $response['message'] = "Welcome to Internshala Trainings";

    echo json\_encode($response);

?>

**Sample.html**

|  |  |
| --- | --- |
|  | <html> |
|  | <head> |
|  | <title>My Title</title> |
|  | </head> |
|  | <body> |
|  | <h1>My Header</h1> |
|  | <p id="abc">This is a paragraph</p> |
|  | <script type="text/javascript" src="[js/script.js](file:///C:\\Users\\apurv\\AppData\\Local\\Temp\\Temp1_M6T10V3_helper_text.zip\\M6T10V3_helper%20text\\js\\script.js" \t "_blank)"> </script> |
|  | </body> |
|  | </html> |
|  |  |



**Other String functions:**

**Slice() function:**

let a = "Apurv is Akshay and Akshay is Apurv";

let b = a.slice(-10);

document.write(b);

**output:**



It fetches last 10 letters from the entered string.It fetches last 10 since, -10 was entered.

**Replace() function:**

let a = "Apurv is Akshay and Akshay is Apurv";

let b = a.replace("Apurv", "Akshay");

document.write(b);

**output:**



It replaces first word with another but only for the first occurrence.