

Assignment No. 3

Q1 Difference betⁿ client side scripting & serverside scripting.

Features	Server side	Client side
Primary Function	The main function of this scripting is to manipulate & grant access to the requested database.	The main purpose of this scripting is to give the requested output to the end-user.
Uses	It is employed at the backend, where the source code is invisible or concealed on the client side.	It is utilized at the end, which users may view through the browser.
Processing	It needs server interaction.	It doesn't need any server interaction.
Security	It is more secure while working on a web app.	It is less secure than server-side scripting due to the code accessibility offered to the client.
Running	It executes on the web server.	It executes on the remote computer system.
Dependability	It doesn't depend on the client.	It depends on the user browser version.

File access	It offers complete access to the file that is stored in the web database server.	It doesn't offer any access to the files on the web servers.
Code Allowance	It enables the backend developer to hide the source code from the user.	The user is given access to the written code after confirming their requirements.
Occurrence	It only responds after the user begins the browsing request.	It happens when the browser processes all of the codes & then acts according to the client's needs.
Affect	It may reduce the server load.	It may effectively customize web pages & offer dynamic websites.
Languages Involved	The server-side scripting programming languages, such as PHP, ColdFusion, Python, ASP.net, Java, C++, Ruby, C# etc.	Its programming languages are HTML, CSS, & JavaScript.

Q2 What is javascript and how to implement javascript with different display possibilities.

JavaScript is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser.

JavaScript Display Possibilities:->

- Writing into an HTML element, using inner HTML
- Writing into the HTML output using `document.write()`
- Writing into an alert box, using `window.alert()`
- Writing into the browser console using `console.log()`

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:→

```
<script>  
document.getElementById("demo").innerHTML = S + S;  
</script>
```

Using `document.write()`:→ For testing purposes, it is convenient to use `document.write()`

Ex:→

```
<script>  
document.write(S + S);  
</script>
```

Using `window.alert()`:→ You can use an alert box to display data:

Ex:→

```
<script>  
window.alert(S + S);  
</script>
```

Using `console.log()`:→ For debugging purposes, you can call the `console.log()` method in the browser to display data.

Ex: \rightarrow `<script>`
 `console.log(5+5);`
`</script>`

JavaScript Print: \rightarrow

- JavaScript does not have any print object or print methods.
- You cannot access output devices from JavaScript.
- The only exception is that you can call the `window.print()` method in the browser to print the content of the current window.

Ex: \rightarrow

`<button onclick="window.print()">Print this page </button>`

Q.3 List ways to declare a js variable & explain
⇒ js let.

In JavaScript, we can declare a variable in different ways by using different keyword. Each keyword holds some specific reason or feature in JS.

Basically, we can declare variables in three different ways by using var, let and const keywords.

1) JavaScript var :-

This keyword is used to declare variables globally. If you used this keyword to declare a variable then the variable can be accessible globally and changeable also.

Ex:-

```
<script>
  var rushi = "Rushi Patil"
  console.log(rushi);
</script>
```

2) JavaScript let:

This keyword is used to declare variable locally. If you used this keyword to declare a variable then the variable can be accessible locally and it is changeable as well. It is good if the code gets huge.

Ex:-

```
<script>
  if (true) {
    let Guru = "Guru Patil"
    console.log(Guru);
  }
</script>
```

- 37) JavaScript Const: This keyword is used to declare variable locally. If you use this keyword to declare a variable then the variable will only be accessible within that block similar to the variable defined by using let and difference between let and const is that the variables declared using const value can't be reassigned.
- EX:-

```
<script>
```

```
const rushi = "Rushi patil";
```

```
console.log(rushi);
```

```
</script>
```

- 4) Automatically:

```
<P id="demo"></P>
```

```
<script>
```

```
x = 5;
```

```
y = 6;
```

```
z = x + y;
```

```
document.getElementById("demo").innerHTML  
= "The value of z is : " + z;
```

```
</script>
```


Q4 List JavaScript operators & explain assignment & comparison operators.

following types of operators in JavaScript:-

- ① Arithmetic Operators
- ② Comparison (Relational) Operators
- ③ Bitwise Operators
- ④ Logical Operators
- ⑤ Assignment operators.
- ⑥ Special operators.

JavaScript Comparison Operators:-

- JavaScript Comparison Operators are mainly used to perform the logical operations that determine the equality or difference betⁿ the values.

Operator Name	Usage	Operation
Equality operator	$a == b$	Compares the equality of two operators.
Inequality operator	$a != b$	Compare inequality of two operators.
Strict Equality operator	$a === b$	compare both value & type of the operand.
Strict Inequality operator	$a !== b$	compare inequality with type.
Greater than operator	$a > b$	Checks if the left operator is greater than the right operator.

Greater than or equal to	$a \geq b$	checks if the left operator is greater than or equal to the right operator.
Less than	$a < b$	checks if the left operator is smaller than the right operator.
Less than or equal to	$a \leq b$	checks if the left operator is smaller than or equal to the right operator.

JavaScript assignment operators:→

Assignment operators assign values to JavaScript variables.

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

Shift Assignment Operator:→

$<<=$	$x << = y$	$x = x << y$
$>>=$	$x >> = y$	$x = x >> y$
$>>>=$	$x >>> = y$	$x = x >>> y$

Bitwise Assignment Operators :-

$\&=$	$x \&= y$	$x = x \& y$
$\wedge=$	$x \wedge= y$	$x = x \wedge y$
$ =$	$x = y$	$x = x y$

Logical Assignment Operators :-

$\&\&=$	$x \&\&= y$	$x = x \&\& (x = y)$
$ =$	$x = y$	$x = x (x = y)$
$??=$	$x ??= y$	$x = x ?? (x = y)$

Q5 Write a JS program for finding vowels & consonant.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script>
```

```
var ch;
```

```
function checkVowel()
```

```
{
```

```
ch = document.getElementById("char").value;
```

```
if (ch)
```

```
{
```

```
temp = document.getElementById("respara");
```

```
temp.style.display = "block";
```

```
if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
```

```
{
```

```
if (ch == "a" || ch == "e" || ch == "i" || ch == "o" || ch == "u")
```

```
document.getElementById("res").innerHTML  
= "a Vowel";
```

```
else if (ch == "A" || ch == "E" || ch == "I" || ch == "O" || ch == "U")
```

```
document.getElementById("res").innerHTML  
= "a Vowel";
```

```
else
```

```
document.getElementById("res").innerHTML  
= "a Consonant";
```

```
}
```

```
else
```

```
document.getElementById("res").innerHTML  
= "neither vowel nor Consonant";
```

```
}
```

```
}
```

```
</script>
```


</head>

<body>

<p> Enter the Character! <input id="char">
<button onclick="checkVowel()"> check
</button> </p>

<p id="resPara" style="display: none;"> It is
 </p>

</body>

</html>

O/p:->

Enter the Character:

Q6 Write JS code for displaying star pyramid.
up to four layers by using for loop.

let n=4;

let string="";

for (let i=1; i<=n; i++)

{

for (let j=1; j<=n-i; j++)

{

string += " ";

}

for (let k=0; k<=2*i-1; k++)

{

string += "*";

}

string += "\n";

}

console.log(string);

O/p:->

```
  *
 * *
* * *
* * * *
```

Q7 Write JS program to displaying even and odd numbers betⁿ 1 to 50.

```
var evenNumbers = [];
var oddNumbers = [];
for (var i = 1; i <= 50; i++)
{
    if (i % 2 == 0)
    {
        evenNumbers.push(i);
    }
    else
    {
        oddNumbers.push(i);
    }
}
console.log("Even number:" + evenNumbers);
console.log("Odd number:" + oddNumbers);
```

O/p:->

~~1-3~~

Even number: 2 4 6 8 10 12 14 16 18
20 22 24 26 28 30 32 34 36
38 40 42 44 46 48 50

odd numbers: 1 3 5 7 9 11 13 15 17
 19 21 23 25 27 29 31 33 35
 37 39 41 43 45 47 49

Q8 Write JS program for displaying leap year or not

```
function checkLeapYear(year)
{
const leap = new Date(year,
  if ((0 == year % 4) && (0 != year % 100) ||
    (0 == year % 400))
  {
    console.log(year + ' leap year');
  }
  else
  {
    console.log(year + ' not a leap year');
  }
}
const year = prompt('Enter a year:');
checkLeapYear(year);
```

O/p!->

Enter a year! 2000
 leap year.

Q9 Write a note popup boxes:

In Javascript popup boxes are used to display the message or notification to the user. There are three types of pop-up boxes in javascript namely Alert Box, Confirm box & Prompt Box.

① Alert Box:→ It is used when a warning messages is needed to be produced when the alert box is displayed to the user. The user needs to press ok & processed.

Syntax:→

```
alert ("your Alert here")
```

Ex:→ <!DOCTYPE html>

```
<html>
```

```
<head>
```

```
<title> Pop-up box - Alert Box </title>
```

```
<style>
```

```
h1 {
```

```
color: green;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h3> alert box </h3>
```

```
<button onclick = "greekAlert()">
```

```
click here </button>
```

```
<script>
```

```
function greekAlert() {
```

```
  alert("An online computer science" +  
        "portal for Greeks");
```

```
} </script> </body> </html>
```

⑪ Prompt Box:→ It is a type of pop up box, which is used to get user input for further use. After entering the required details, user has to click ok to proceed to the next stage else by processing the cancel button, user returns the null value.

Syntax:→

prompt("your prompt here");

Ex:→

```
<!DOCTYPE html>
<html>
<head>
  <title> pop-up prompt-box </title>
</head>
<body>
  <h3> Prompt Box </h3>
  <input type="button" onclick="geek prompt();"
    value="click here" />
  <script>
    function geekprompt() {
      let x = prompt("Enter your mail add.");
      document.write("Your ID: " + x);
    }
  </script>
</body>
</html>
```

⑫ Confirm Box:→ It is a type of pop-up box that is used to get authorization from the user. The user has to press the ok or cancel button to proceed.

Syntax:->

```
confirm("your query here")
```

Ex:->

```
<body>
  <h3> confirm box </h3>
  <button on click = "geek confirm()">
    click here for confirm box </button>
  <p id = "geek"> </p>
  <script>
    function geek confirm () {
      var x;
      if (confirm ("press a button!"))
        x = "OK pressed!";
      else {
        x = "cancel!";
      }
      document.getElementById ("geek").innerHTML
        = x;
    }
  </script>
</body>
```


10 List events in Javascript & explain any two events with examples.

Events in JavaScript:-

- ① Click - onclick
- ② mouseover - onmouseover
- ③ mouseout - onmouseout
- ④ mousedown - onmousedown
- ⑤ mouseup - onmouseup
- ⑥ mousemove - onmousemove
- ⑦ keydown & keyup - onkeydown & onkeyup
- ⑧ focus - onfocus
- ⑨ submit - onsubmit
- ⑩ blur - onblur
- ⑪ change - onchange
- ⑫ load - onload
- ⑬ unload - onunload
- ⑭ resize - onresize
- ⑮ doubleclick - ondblclick

Onload Event's

- The onload attribute fires when an object has been loaded.
- onload is most often used within the <body> element to execute a script once a web page has completely loaded all content (images, script files, css files etc)
- It can be used on other elements as well as ("supported HTML tags")
- The onload attribute can be used to check the visitor's browser type & browser version, & load the proper version of the web page based on the information.

Ex:->

```

```

```
<script>
```

```
function loadImage(){
    alert("Image is loaded");
}
```

```
</script>
```

OnClick Event:->

- The onclick event generally occurs when the user clicks on an element.
- It allows the programmer to execute a JavaScript's function when an element gets clicked.
- This event can be used for validating a form, warning messages & many more.
- This event can be dynamically added to any element.
- It supports all HTML elements except <html>, <head>, <title>, <style>, <script>, <base>, <iframe>, <bdo>,
, <meta> & <param>. It means we cannot apply the onclick event on the given tags.

Ex:->

```
<script>
```

```
function func() {
    alert("Welcome");
}
```

```
</script>
```

```
<body>
```

```
<button onclick="func()"> click me </button>
```

```
</body>
```


Javascript Array Methods:->

- ① ~~map~~ map():-> This method creates a new array with the results of calling a provided function on every element in this array.
- ② filter():-> This method creates a new array with only elements that passes the condition inside the provided function.
- ③ ~~sort~~ sort():-> This method is used to arrange/sort array's elements either in ascending or descending order.
- ④ forEach():-> This method helps to loop over array by executing a providing callback function for each element in an array.
- ⑤ concat():-> This method is used to merge two or more arrays & returns a new array, without changing the existing arrays.
- ⑥ every():-> This method checks every element in the array that passes the condition, returning true or false as appropriate.
- ⑦ some():-> This method checks if at least one element in the array that passed the condition, returning true or false as appropriate.
- ⑧ includes():-> This method checks if an array includes the element that passes the condition, returning true or false as appropriate.

- ⑨ `join()` → This method returns a new string by concatenating all of the array's elements separated by the specified separator.
- ⑩ `reduce()` → This method applies a function against an accumulator & each element in the array to reduce it to a single value.
- ⑪ `find()` → This method returns the value of the first element in an array that passes the test in a testing function.
- ⑫ `findIndex()` → This method returns the index of the first element in an array that passes the test in a testing function.
- ⑬ `indexOf()` → This method returns the index of the first occurrence of the specified element in the array, or -1 if it is not found.
- ⑭ `fill()` → This method fills the elements in an array with a static value & returns the modified array.
- ⑮ `slice()` → This method returns a new array with specified start to end elements.
- ⑯ `reverse()` → This method reverses an array in place. Element at last index will be first & element at 0 index will be last.

①⑦ `push()` :- This method adds one or more elements to the end of array & returns the new length of the array.

①⑧ `pop()` :- This method removes the last element from the end of array & returns that element.

①⑨ `shift()` :- This method removes the first element from an array & returns that element.

②② `unshift()` :- This method adds one or more elements to the beginning of an array & returns the new length of the array.

Q12 Write a code for by using JS function for given string or number is palindrome or not.

For Given string is palindrome or not.

```
<body> <script>
```

```
Function checkPalindrome(string) {
```

```
    const len = string.length;
```

```
    for (let i = 0; i < len/2; i++) {
```

```
        if (string[i] !== string[len-1-i]) {
```

```
            return "It is not palindrome";
```

```
        }
```

```
    }
```

```
    return "It is palindrome";
```

```
}
```

```
const string = prompt("Enter the string!");
```

```
const value = checkPalindrome(string);
```

```
console.log(value);
```

```
</script>
```

```
</body>
```

For Given ~~string~~ number is palindrome or not

```
<script>
```

```
function palindrome() {
```

```
    var a, b, no, temp = 0;
```

```
    no = Number(document.getElementById  
        ("palindrome").value);
```

```
    b = no;
```

```
    while (no > 0)
```

```
    {
```

```
        a = no % 10;
```

```
no = parseInt(no / 10);  
temp temp = temp * 10 + a;  
}  
if (temp == b)  
{  
    alert("It is a palindrom Number");  
}  
else  
{  
    alert("It is not a palindrom number");  
}  
}  
</script>
```