**Program-1 |**Receive two numbers and display their sum.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Sum of Numbers</title>

<link rel="preconnect" href="https://fonts.googleapis.com" />

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />

<link

href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500&display=swap"

rel="stylesheet"

/>

<style>

body {

font-family: 'Poppins', sans-serif;

margin: 1rem;

padding: 2rem;

}

button {

margin-top: 1rem;

}

#result {

margin-top: 1rem;

font-weight: 500;

}

.red {

color: red;

}

</style>

</head>

<body>

<form>

<h2>Sum of two numbers</h2>

<label> Enter first number: <br /><input type="text" id="num1" /> </label>

<br />

<label>

Enter second number: <br /><input type="text" id="num2" />

</label>

<br />

<button id="sum">Sum</button>

<button type="reset">Reset</button>

<div id="result"></div>

</form>

<script>

const num1 = document.querySelector('#num1');

const num2 = document.querySelector('#num2');

// Get the result span

const result = document.getElementById('result');

// Add event listener for sum button

document.querySelector('form').addEventListener('submit', function (e) {

e.preventDefault();

// Get the values from the input elements

const n1 = num1.value;

const n2 = num2.value;

// Add the numbers

const sum = parseInt(n1) + parseInt(n2);

// Display the result

if (sum) {

result.innerHTML = 'Sum of ' + n1 + ' and ' + n2 + ' is ' + sum;

return;

}

result.innerHTML =

'<span class="red">Please enter valid numbers</span>';

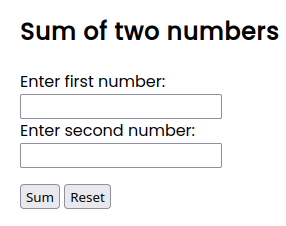
});

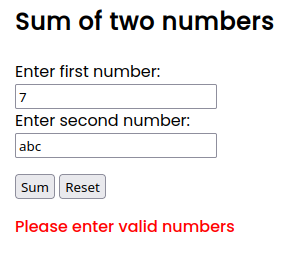
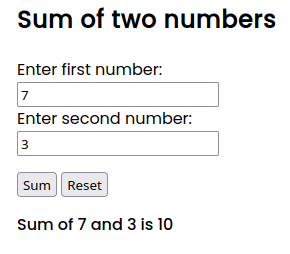
</script>

</body>

</html>

# **Output of Program-1**





**Program-2 |** Receive lower and upper range and display all prime numbers between the range.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Prime Numbers</title>

<link rel="preconnect" href="https://fonts.googleapis.com" />

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />

<link

href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500&display=swap"

rel="stylesheet"

/>

<style>

body {

font-family: 'Poppins', sans-serif;

margin: 1rem;

padding: 2rem;

}

.error {

color: red;

margin-top: 2rem;

font-weight: 500;

}

#result {

max-width: 67%;

margin-top: 1rem;

}

</style>

</head>

<body>

<h2>Prime Numbers</h2>

<label> Enter Range From: <input type="text" id="num1" /> </label>

<label> Enter Range To: <input type="text" id="num2" /> </label>

<button id="btn">Get Prime Numbers</button>

<div id="result"></div>

<script>

const num1 = document.querySelector('#num1');

const num2 = document.querySelector('#num2');

console.log(num1, num2);

// Get the result span

const result = document.getElementById('result');

const getPrime = document

.querySelector('#btn')

.addEventListener('click', () => {

const lower = parseInt(num1.value);

const higher = parseInt(num2.value);

if (lower > higher) {

result.innerHTML = '<span class="error">Invalid Range</span>';

return;

}

if (lower && higher) {

result.textContent = '';

const primes = primeNumber(lower, higher);

console.log(primes);

for (let temp of primes) {

result.textContent += temp + ' ';

}

return;

}

result.textContent = 'Please enter valid numbers';

});

// Prime number between two numbers

function primeNumber(num1, num2) {

let prime = [];

for (let i = num1; i <= num2; i++) {

let isPrime = true;

for (let j = 2; j < i; j++) {

if (i % j == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

prime.push(i);

}

}

return prime;

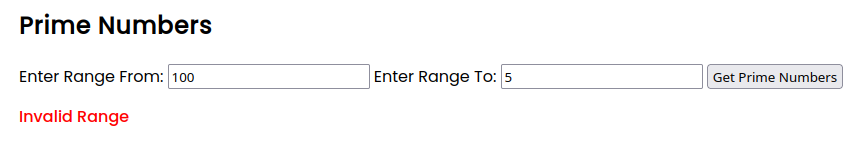
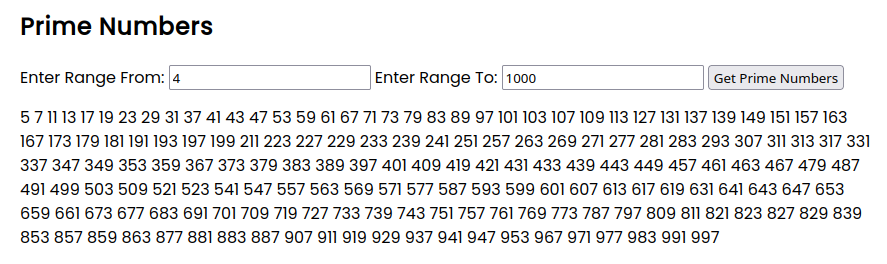
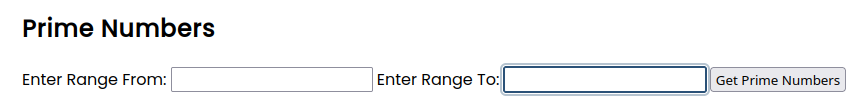
}

</script>

</body>

</html>

# Output of Program-2



**Program-3 |** Take two strings and display "found" if first string is a substring of the second string else "not found".

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Check Substring</title>

</head>

<body>

<h2>Check Sub-string</h2>

<label> Enter first string: <input type="text" id="num1" /> </label>

<label> Enter second string: <input type="text" id="num2" /> </label>

<button id="btn">Check</button>

<div id="result"></div>

<script>

const num1 = document.querySelector('#num1');

const num2 = document.querySelector('#num2');

console.log(num1, num2);

// check if the string is substring of another string

const result = document.getElementById('result');

const btn = document.querySelector('#btn');

btn.addEventListener('click', () => {

const str1 = num1.value;

const str2 = num2.value;

if (str1 && str2) {

result.textContent = '';

const found = str2.includes(str1);

if (found) {

result.textContent = 'found';

} else {

result.textContent = 'Not found';

}

return;

}

result.textContent = 'Please enter valid strings';

});

// Check if string1 is substring of string2

function isSubstringOf(string1, string2) {

for (let i = 0; i < string2.length; i++) {

if (string2[i] == string1[0]) {

let j = i;

for (let k = 0; k < string1.length; k++) {

if (string2[j] != string1[k]) {

break;

}

j++;

}

if (k == string1.length - 1) {

return true;

}

}

}

return false;

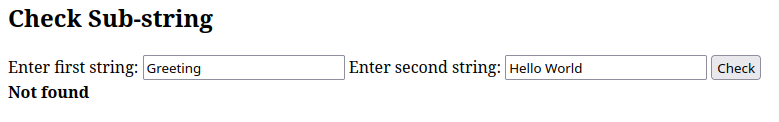
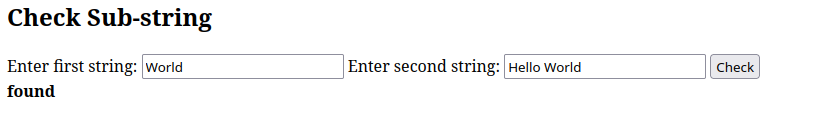
}

</script>

</body>

</html>

# Output of Program-3



**Program-4 |**Design two combos (select) and the options selected in the first combo shall not appear in the second combo. Both the combos are having same set of options.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Combo box</title>

</head>

<body>

<!-- Select tag with 10 random options -->

<select id="select1"></select>

<select id="select2"></select>

<script>

const selectOptions = [

'Bilaspur',

'Raipur',

'Korba',

'Dhamtari',

'Durg',

'Dantewada',

'Rajnandgaon',

'Bhatapara',

'Belha',

'Bhilai'

];

const select1 = document.querySelector('#select1');

select1.addEventListener('change', removeOptionFromSelect);

const select2 = document.querySelector('#select2');

// Initial render of select options

for (let i = 0; i < selectOptions.length; i++) {

const option = document.createElement('option');

option.value = selectOptions[i];

option.textContent = selectOptions[i];

select1.appendChild(option);

select2.appendChild(option.cloneNode(true));

}

function removeOptionFromSelect(e) {

select2.innerHTML = '';

const selectedIndex = select1.options.selectedIndex;

const filteredArray = selectOptions.filter((value, index) => {

return index !== selectedIndex;

});

updateOptions(filteredArray);

}

// Receives an array and renders it in a select tag

function updateOptions(filteredOptions) {

console.log(filteredOptions);

for (let i = 0; i < filteredOptions.length; i++) {

const option = document.createElement('option');

option.value = filteredOptions[i];

option.textContent = filteredOptions[i];

select2.appendChild(option);

}

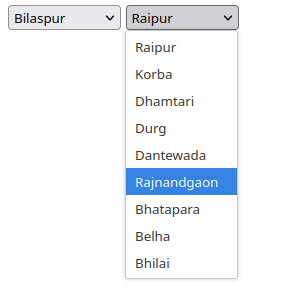
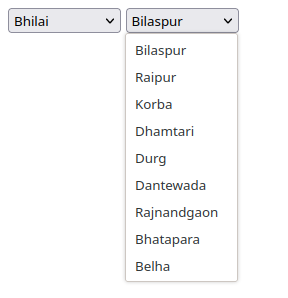
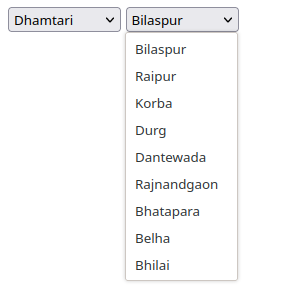
}

</script>

</body>

</html>

# Output of Program-4



**Program-5 |**

Design a textbox only receives exactly six characters.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>6 digit pattern</title>

</head>

<body>

<form>

<h2>Enter 6 digits:</h2>

<!-- Pattern -->

<input

type="text"

pattern=".{6}"

title="Please enter exact six characters."

/>

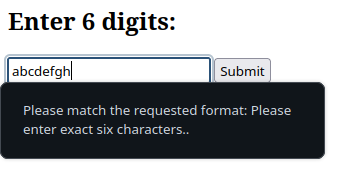
<button type="submit">Submit</button>

</form>

</body>

</html>

# Output of Program-5



**Program-6 |** Design a dynamic table with click add row button which create another row with "add row" button.

Once add row is clicked it would change to "delete row" and onclick that particularrow will be deleted.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Table - Add New Row</title>

<style>

table {

margin-top: 2rem;

padding: 2rem 1rem 2rem 1rem;

font-family: Arial, Helvetica, sans-serif;

border-collapse: collapse;

width: 100%;

}

table td,

table th {

border: 1px solid #ddd;

padding: 8px;

}

table tr:nth-child(even) {

background-color: #f2f2f2;

}

table tr:hover {

background-color: #ddd;

}

table th {

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: #04aa6d;

color: white;

}

</style>

</head>

<body>

<table id="table"></table>

<script>

let tableData = [];

const content = document.querySelector('#content');

const table = document.querySelector('#table');

const addrow = document.querySelector('#addrow');

function renderTable() {

let html = ` <tr>

<th>S.No</th>

<th>Content</th>

<th>Action</th>

</tr>`;

for (let [index, value] of tableData.entries()) {

html += `<tr>

<td>${index + 1}</td>

<td>${value}</td>

<td><button class="deletebtn" onClick="delRow(${index})">Delete</button></td>

</tr>`;

}

html += `<tr>

<td>\*</td>

<td>

<input

type="text"

style="width: 80%"

placeholder="Enter text content here"

/>

</td>

<td><button onClick="addRow()">Add</button></td>

</tr>`;

table.innerHTML = html;

}

function delRow(i) {

tableData = tableData.filter((item, index) => index !== i);

renderTable();

}

function addRow() {

const input = document.querySelector('input');

const value = input.value.trim();

tableData.push(value);

input.value = '';

renderTable();

}

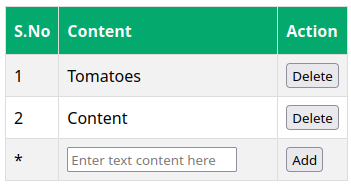
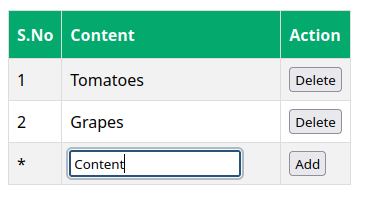
renderTable();

</script>

</body>

</html>

# Output of Program-6



**Program-7 |** On click of a button "create button" will create another button "button1" and onclick of "button1","button2" will be created and so on... Once create button is clicked then it should be inactive.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Button Creator</title>

</head>

<body>

<button id="b0">Click Button</button>

<script>

let count = 0;

const button0 = document.querySelector('#b0');

button0.addEventListener('click', createButton);

function createButton(e) {

console.log('hii');

const currentButton = e.target;

currentButton.disabled = true;

const newBtn = document.createElement('button');

newBtn.textContent = `Button ${count + 1}`;

newBtn.addEventListener('click', createButton);

document.body.append(newBtn);

count++;

}

</script>

</body>

</html>

# Output of Program-7





**Program-8 |** Receive one quadratic equation

[AX2 + BX + C = 0] and find the roots of the equation.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Quadratic Eq. Roots</title>

<style>

input {

width: 2rem;

}

</style>

</head>

<body>

<h2>Quadratic Equation:</h2>

<p>Enter values of A B and C:</p>

<div>

<input type="text" placeholder="A" id="a" />X<sup>2</sup> +

<input type="text" placeholder="B" id="b" />X +

<input type="text" placeholder="C" id="c" /> = 0

<button id="calc">Find Roots</button>

</div>

<div id="result"></div>

<script>

const inputA = document.querySelector('#a');

const inputB = document.querySelector('#b');

const inputC = document.querySelector('#c');

const calcBtn = document.querySelector('#calc');

const result = document.querySelector('#result');

calcBtn.addEventListener('click', () => {

let root1, root2;

let a = inputA.value;

let b = inputB.value;

let c = inputC.value;

if (isNaN(a) || isNaN(b) || isNaN(c)) {

result.innerHTML = 'Please enter valid numbers';

clearInputs();

return;

}

// calculate discriminant

let discriminant = b \* b - 4 \* a \* c;

// condition for real and different roots

if (discriminant > 0) {

root1 = (-b + Math.sqrt(discriminant)) / (2 \* a);

root2 = (-b - Math.sqrt(discriminant)) / (2 \* a);

// result

console.log(

`The roots of quadratic equation are ${root1} and ${root2}`

);

result.textContent = `The roots of quadratic equation are ${root1} and ${root2}`;

}

// condition for real and equal roots

else if (discriminant == 0) {

root1 = root2 = -b / (2 \* a);

// result

console.log(

`The roots of quadratic equation are ${root1} and ${root2}`

);

result.textContent = `The roots of quadratic equation are ${root1} and ${root2}`;

}

// if roots are not real

else {

let realPart = (-b / (2 \* a)).toFixed(2);

let imagPart = (Math.sqrt(-discriminant) / (2 \* a)).toFixed(2);

result.textContent = `The roots of quadratic equation are ${realPart} + ${imagPart}i and ${realPart} - ${imagPart}i`;

}

});

function clearInputs() {

inputA.value = '';

inputB.value = '';

inputC.value = '';

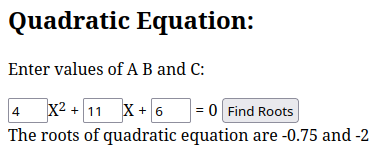
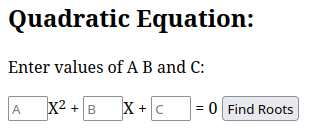
}

</script>

</body>

</html>

# Output of Program-8



**Program-9 |** Design an Html form contains paragraph, text, checkbox, button. All elements have a common onclick event handler functionA() which displays clicked elements’ content, value, style in alert box.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Practical 9</title>

<style>

.h1 {

font-size: 1.5rem;

font-weight: 700;

padding: 0;

margin: 1rem 0 0.5rem 0;

}

.h2 {

padding: 0;

margin: 1rem 0 0.5rem 0;

font-size: 1rem;

font-weight: 400;

}

</style>

</head>

<body>

<p class="h1" onclick="functionA(this)">Form Elements</p>

<form>

<p onclick="functionA(this)" class="h2">

Please enter all the required details.

</p>

<input

type="text"

name="sometext"

id="textid"

onclick="functionA(this)"

/>

<input

type="checkbox"

value="Checkbox1"

name="checkbox1"

onclick="functionA(this)"

/>

<input

type="checkbox"

value="Checkbox2"

name="checkbox2"

onclick="functionA(this)"

/>

<button onclick="functionA(this)">Submit</button>

<button onclick="functionA(this)">Reset</button>

</form>

<script>

function functionA(clickedItem) {

console.log(clickedItem.style);

const style = getComputedStyle(clickedItem);

console.log(style);

alert(

`Clicked tag: ${clickedItem.tagName}\nClicked Content: ${

clickedItem.innerHTML

}\nClicked Value: ${

clickedItem.value

}\nClicked Style: ${JSON.stringify(style)}`

);

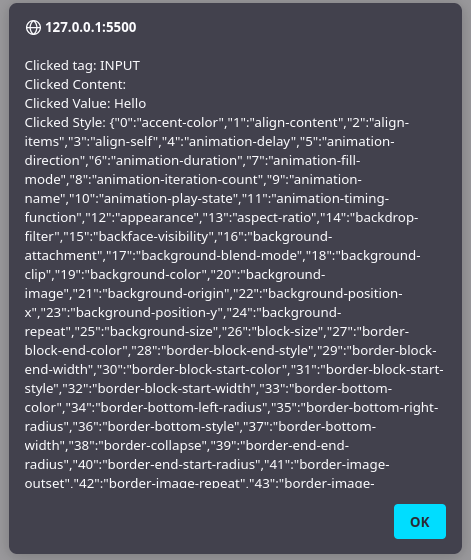
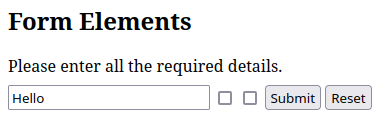
}

</script>

</body>

</html>

# Output of Program-9



**Program-10 |** Design an HTML page given below: If we click toggle button “>” then copy left to right and button becomes “<” if left is empty otherwise right becomes empty and button becomes “>”.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Practical 10</title>

<style>

.container {

margin-top: 3rem;

display: flex;

justify-content: center;

align-items: center;

gap: 1.5rem;

}

</style>

</head>

<body>

<div class="container">

<div class="localaddress">

<textarea placeholder="Local address" id="txtlocal"></textarea>

</div>

<button id="togglebtn">></button>

<div class="permanent">

<textarea placeholder="Permanent address" id="txtperm"></textarea>

</div>

</div>

<script>

const localAddress = document.querySelector('#txtlocal');

const permanentAddress = document.querySelector('#txtperm');

const button = document.querySelector('#togglebtn');

button.addEventListener('click', () => {

console.log(localAddress);

if (localAddress.value === '') {

button.textContent = '>';

permanentAddress.value = '';

} else {

permanentAddress.value = localAddress.value;

localAddress.value = '';

button.textContent = '<';

}

});

</script>

</body>

</html>

# Output of Program-10



**Program-11 |** Design an HTML with one text box and one list box and one add button. Click on add button the text of the text box (if not empty) will be added to the list with serial number. The list shows only three items rest items will be kept page wise. Page index appears at the bottom of the list as buttons.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>List3</title>

<style>

.pagination {

display: flex;

justify-content: center;

align-items: center;

margin: 0 auto;

padding: 0;

list-style: none;

width: 100%;

max-width: 100%;

margin-top: 1rem;

margin-bottom: 1rem;

border-radius: 0.25rem;

gap: 0.5rem;

}

#list {

margin-top: 2rem;

width: 18rem;

}

.container {

display: flex;

justify-content: center;

flex-direction: column;

align-items: center;

margin: 0 auto;

padding: 0;

list-style: none;

width: 100%;

max-width: 100%;

margin-top: 1rem;

margin-bottom: 1rem;

border-radius: 0.25rem;

}

.controls {

display: flex;

}

</style>

</head>

<body>

<div class="container">

<h2>My List</h2>

<div class="controls">

<input type="text" id="txt" />

<button id="addbtn">Add</button>

</div>

<select id="list" multiple size="3"></select>

</div>

<div class="pagination"></div>

<script>

const listItems = [];

let currentpage = 1;

const inputBox = document.querySelector('#txt');

const list = document.querySelector('#list');

const paginationDiv = document.querySelector('.pagination');

const addBtn = document

.querySelector('#addbtn')

.addEventListener('click', addItemToList);

function addItemToList() {

const newItem = inputBox.value.trim();

if (newItem) {

listItems.push({ index: listItems.length + 1, value: newItem });

inputBox.value = '';

updateList();

}

}

function updateList() {

const items = getItemsFromList();

list.innerHTML = items

.map((item, index) => `<option>${item.index}. ${item.value}</option>`)

.join('');

getPagination();

}

function getPagination() {

const numOfPaginationButtons = Math.ceil(listItems.length / 3);

console.log('Num of pagination buttons: ' + numOfPaginationButtons);

// Create pagination buttons

let paginationButtons = '';

for (let i = 0; i < numOfPaginationButtons; i++) {

const pageIndex = i + 1;

paginationButtons += `<button id="${pageIndex}" onClick="paginationHandler(${pageIndex})">${pageIndex}</button>`;

}

paginationDiv.innerHTML = paginationButtons;

}

function getItemsFromList() {

const page\_size = 3;

return listItems.slice(

(currentpage - 1) \* page\_size,

currentpage \* page\_size

);

}

function paginationHandler(pageIndex) {

currentpage = pageIndex;

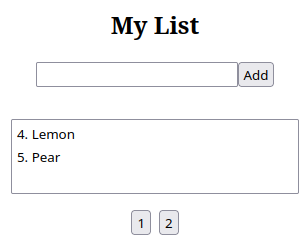
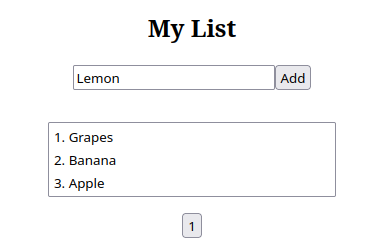
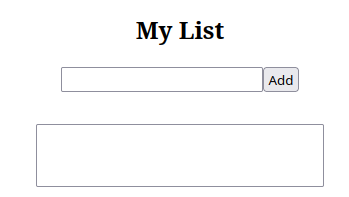
updateList();

}

</script>

</body></html>

# Output of Program-11



**Program-12 |** Design an html page which ask you your name in prompt on load and repeats asking if you give empty name otherwise, it displays your name in a paragraph.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Document</title>

</head>

<body>

<p id="name"></p>

<script>

let name;

while (!name) {

console.log(name, !name);

name = prompt('Enter your name');

}

document.getElementById(

'name'

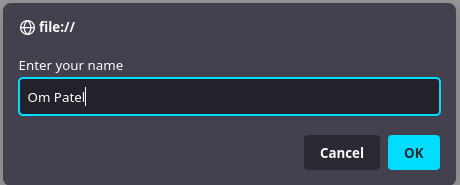
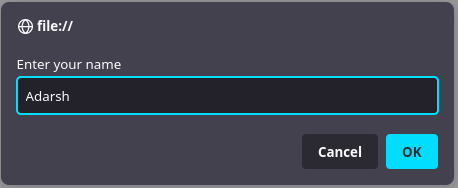
).innerHTML = `Hello <strong>${name}</strong>!`;

</script>

</body>

</html>

# Output of Program-12



**Program-13 |** Do practice on addEventListener(), removeEventListener(), multiple event listeners on the same

element, bubbling and capturing events on nested elements.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<link

href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500&family=Uchen&display=swap"

rel="stylesheet"

/>

<title>Bubbling and Capturing</title>

<style>

div {

transition: all 0.5s;

transition-timing-function: ease-in-out;

}

body {

margin: 0;

min-height: 100vh;

font-family: 'Poppins', sans-serif;

}

body,

div {

display: flex;

justify-content: center;

align-items: center;

}

.grandparent {

width: 200px;

height: 200px;

background-color: red;

}

.parent {

width: 130px;

height: 130px;

background-color: blue;

}

.child {

width: 60px;

height: 60px;

background-color: green;

}

.clicked {

background-color: #ddd;

box-shadow: rgba(0, 0, 0, 0.16) 0px 1px 4px,

rgb(51, 51, 51) 0px 0px 0px 3px;

}

.container {

display: flex;

flex-direction: column;

gap: 2rem;

}

.controls {

display: flex;

gap: 1rem;

}

</style>

</head>

<body>

<div class="container">

<div class="grandparent" id="grandparent">

<div class="parent" id="parent">

<div class="child" id="child"></div>

</div>

</div>

<div class="controls">

<label

>Event Bubbling:<input

type="radio"

name="choice"

id="bubble"

value="bubble"

checked

/></label>

<label

>Event Capturing:<input

type="radio"

name="choice"

id="capture"

value="capture"

/></label>

</div>

</div>

<script>

let bubbling = true;

const grandparent = document.querySelector('.grandparent');

const parent = document.querySelector('.parent');

const child = document.querySelector('.child');

// Same Event Listener for element

const radiobtn1 = document.querySelector('#bubble');

const radiobtn2 = document.querySelector('#capture');

radiobtn1.addEventListener('change', optionChangeListener);

radiobtn2.addEventListener('change', optionChangeListener);

function grandParentEventHandler(e) {

console.log(e.eventPhase);

console.log('grandparent');

// capturing 500; 900

grandparent.classList.add('clicked');

setTimeout(

() => {

grandparent.classList.remove('clicked');

},

bubbling ? 900 : 500

);

}

function parentEventHandler(e) {

parent.classList.add('clicked');

setTimeout(

() => {

parent.classList.remove('clicked');

},

bubbling ? 600 : 800

);

}

function childEventHandler(e) {

child.classList.add('clicked');

console.log('child');

setTimeout(

() => {

child.classList.remove('clicked');

},

bubbling ? 300 : 1000

);

}

function optionChangeListener(e) {

setupEventListeners(e.target.value);

}

function setupEventListeners(type) {

switch (type) {

case 'bubble': {

bubbling = true;

// Remove capturing event

grandparent.removeEventListener(

'click',

grandParentEventHandler,

true

);

parent.removeEventListener('click', parentEventHandler, true);

child.removeEventListener('click', childEventHandler, true);

// Add bubble event

grandparent.addEventListener('click', grandParentEventHandler);

parent.addEventListener('click', parentEventHandler);

child.addEventListener('click', childEventHandler);

break;

}

case 'capture': {

bubbling = false;

// Remove bubbling event

grandparent.removeEventListener('click', grandParentEventHandler);

parent.removeEventListener('click', parentEventHandler);

child.removeEventListener('click', childEventHandler);

// Add bubble event

grandparent.addEventListener(

'click',

grandParentEventHandler,

true

);

parent.addEventListener('click', parentEventHandler, true);

child.addEventListener('click', childEventHandler, true);

break;

}

default: {

grandparent.addEventListener('click', grandParentEventHandler);

parent.addEventListener('click', parentEventHandler);

child.addEventListener('click', childEventHandler);

break;

}

}

}

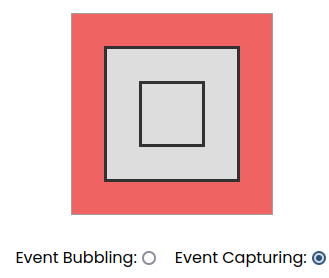
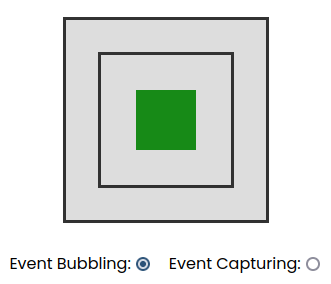
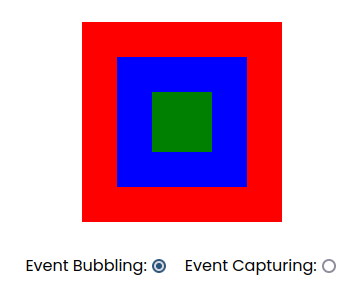
setupEventListeners();

</script>

</body>

</html>

# Output of Program-13



PHP Programs

**Program-1 |** Take two values from two text boxes and display their sum if values are numeric otherwise concat them. Output in a paragraph on the same page.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Concat or SUM</title>

</head>

<body>

<form action="/php-practicals/1.php" method="post">

<label>Enter first value:<input type="text" id="value1" placeholder="Some value" name="value1" required></label>

<label>Enter second value:<input type="text" id="value2" placeholder="Some value" name="value2" required></label>

<button type="submit">Submit</button>

</form>

</body>

</html>

<?php

if(isset($\_POST)) {

$value1 = $\_POST['value1'];

$value2 = $\_POST['value2'];

if(is\_numeric($value1) && is\_numeric($value2)) {

$result = "Sum: " . $value1 + $value2;

} else {

$result = "Concatenation:" . $value1 . $value2;

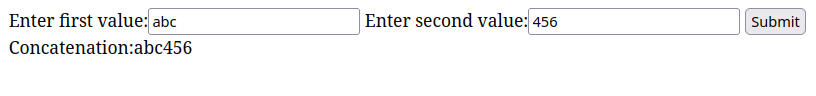
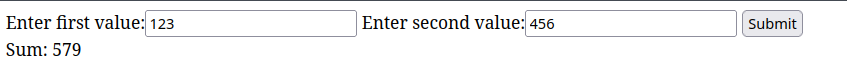
}

echo $result;

}

?>

# Output of Program-1



**Program-2 |** Display the checked values of all checkboxes having same name in a list on the same page.

<?php

if (isset($\_POST['adarsh'])) {

$checkboxValues = $\_POST['adarsh'];

echo "<hr/>";

echo "<h2>You selected:</h2>";

echo "<ul>";

foreach($checkboxValues as $checkboxValue){

echo "<li>$checkboxValue</li>";

}

echo "</ul>";

echo "<hr/>";

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Checkboxes array</title>

<style>

.checkboxes {

display: flex;

flex-direction: column;

align-items: flex-start;

}

</style>

</head>

<body>

<form action="/php-practicals/2.php" method="post">

<div class="checkboxes">

<label><input type="checkbox" name="adarsh[]" value="Fix Bugs">Fix Bugs</label>

<label><input type="checkbox" name="adarsh[]" value="Create a new brach">Create a new brach</label>

<label><input type="checkbox" name="adarsh[]" value="Commit new changes to code">Commit new changes to code</label>

<label><input type="checkbox" name="adarsh[]" value="Push latest changes to github">Push latest changes to github</label>

<label><input type="checkbox" name="adarsh[]" value="Pay Electricity Bill">Pay Electricity Bill</label>

<label><input type="checkbox" name="adarsh[]" value="Pay Broadband Bill">Pay Broadband Bill</label>

<label><input type="checkbox" name="adarsh[]" value="Add More Bugs to Fix later">Add More Bugs to Fix later</label>

<button type="submit">Submit</button>

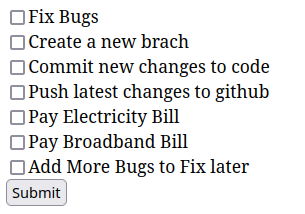
</div>

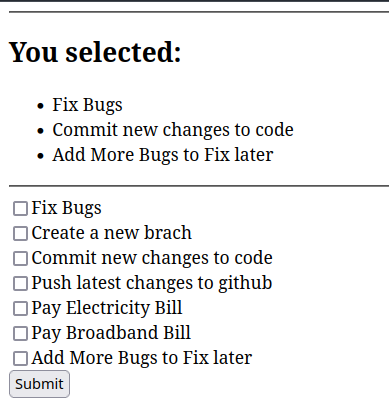
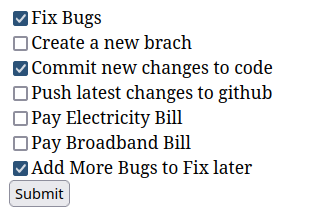
</form>

</body>

</html>

# Output of Program-2





**Program-3 |** Web page contains text box to write name and add button. On click of add button one checkbox with value same as text box and name as “chk1” appears in the same page provided user gives a new name which does not exist in checkboxes. User can click the add button many times. There will be one submit button when the page contains at least one check box and on press submit button all selected names are displayed on the same page.

[do not use JavaScript]

<?php

// create an empty array.

session\_start();

// Check if names is not set in session object

if (!isset($\_SESSION['names'])) {

// if not set, set it to an empty array

$\_SESSION['names'] = [];

}

if (isset($\_POST['chk1'])) {

$checkboxValues = $\_POST['chk1'];

echo "<h2>Selected Names: </h2>";

echo "<ul>";

foreach($checkboxValues as $checkboxValue){

echo "<li>$checkboxValue</li>";

}

echo "</ul>";

}

if(isset($\_POST['newname'])){

// add the new name to the array.

// check if the name is already in the array.

if(in\_array($\_POST['newname'], $\_SESSION['names'])){

echo "<h2 class='red'>Name already exists</h2>";

} else {

// add the name to the array.

array\_push($\_SESSION['names'], $\_POST['newname']);

echo "<h2>Name added</h2>";

}

}

// check if names contains any item.

if(count($\_SESSION['names']) > 0){

// loop through the names array.

echo "<hr/>";

echo "<form action='/php-practicals/3.php' method='post'>";

echo "<div class='listnames'>";

foreach($\_SESSION['names'] as $name){

// display the name.

echo "<label><input type='checkbox' name='chk1[]' value='$name'>$name</label>";

}

echo '<button type="submit" name="submitcheck">Submit</button>';

echo "</div>";

echo "</form>";

echo "<hr/>";

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Add Name</title>

<style>

.red{

color: red;

}

.listnames{

display: flex;

flex-direction: column;

gap: 0.2rem;

margin: 2rem 1rem 2rem 1rem;

}

button{

max-width: min-content;

margin-top: 1rem;

}

</style>

</head>

<body>

<form action="/php-practicals/3.php" method="post">

<input type="text" placeholder="Enter name" name="newname" required>

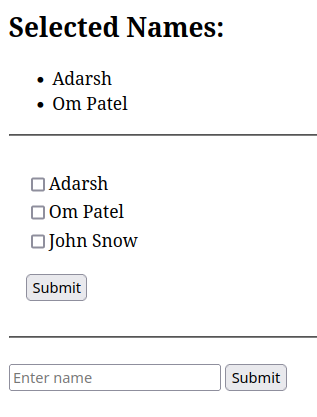
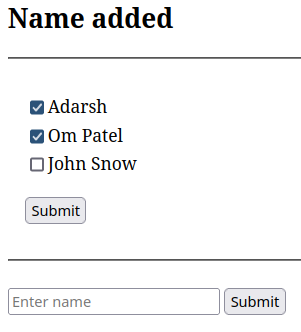
<button type="submit">Submit</button>

</form>

</body>

</html>

# Output of Program-3



**Program-4 |** Design a select box having multiple selection facility and on click of submit display all selected items on a list item.

<?php

if(isset($\_POST["cities"]))

{

echo "Selected Cities:";

echo "<ul>";

// Retrieving each selected option

foreach ($\_POST['cities'] as $city)

echo "<li>$city</li>";

}

echo "</ul>";

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Checkboxes array</title>

<style>

select {

height: 15rem;

width: 14rem;

display: block;

}

</style>

</head>

<body>

<form action="/php-practicals/4.php" method="post">

<label for="cities" >Select cities</label>

<select id="cities" name="cities[]" multiple>

<option>Bilaspur</option>

<option>Raipur</option>

<option>Durg</option>

<option>Raigarh</option>

<option>Kolkata</option>

<option>Delhi</option>

<option>Nagpur</option>

<option>Mumbai</option>

<option>Kerela</option>

<option>Banglore</option>

</select>

<button type="submit">Submit</button>

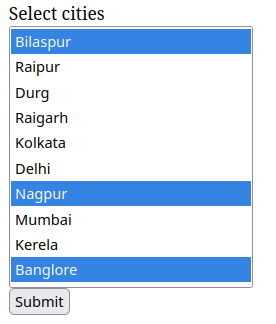
</div>

</form>

</body>

</html>

# Output of Program-4



**Program-5 |**Display all SERVER superglobals in a web page.

<?php

$globals = array(

'$\_SERVER' => $\_SERVER, '$\_ENV' => $\_ENV,

'$\_REQUEST' => $\_REQUEST, '$\_GET' => $\_GET,

'$\_POST' => $\_POST, '$\_COOKIE' => $\_COOKIE,

'$\_FILES' => $\_FILES

);

?>

<html>

<style>

<?php // Adjust CSS formatting for your output ?>

.left {

font-weight: 700;

}

.right {

font-weight: 700;

color: #009;

}

.key {

color: #d00;

font-style: italic;

}

</style>

<body>

<?php

echo '<h1>Superglobals</h1>';

foreach ($globals as $globalkey => $global) {

echo '<h3>' . $globalkey . '</h3>';

foreach ($global as $key => $value) {

echo '<span class="left">' . $globalkey . '[<span class="key">\'' . $key . '\'</span>]</span> = <span class="right">' . $value . '</span><br />';

}

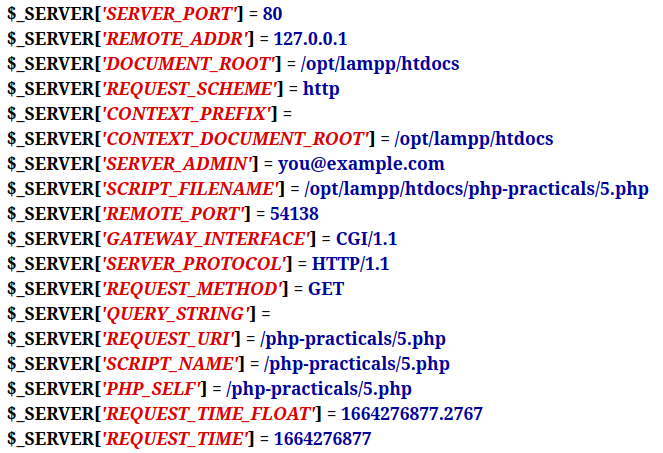
}

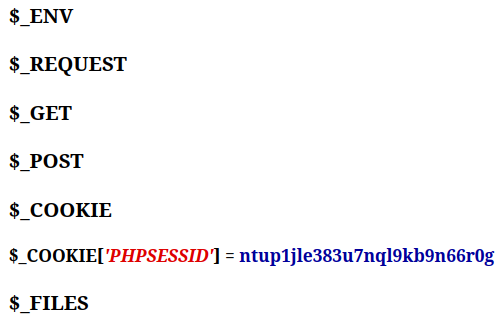
?>

</body>

</html>

# Output of Program-5



****

**Program-6 |** Demonstrate sort, asort and ksort

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>SORT ASORT KSORT</title>

</head>

<body>

<?php

echo "<h2>Sort, ASORT, KSORT Functions</h2>";

echo "<h3>Sort()</h3>";

echo "<p>An Array to be sorted</p>";

$cars = array("Volvo", "BMW", "Toyota", "Honda", "Mercedes");

echo "<pre>";

print\_r($cars);

echo "</pre>";

echo "Sorted Array:<br>";

sort($cars);

echo "<pre>";

print\_r($cars);

echo "</pre>";

echo "<p>Same with array of integers</p>";

$numbers = array(4, 6, 2, 22, 11);

echo "<pre>";

print\_r($numbers);

echo "</pre>";

// sort numbers

sort($numbers);

echo "<p>Sorted Array</p>";

echo "<pre>";

print\_r($numbers);

echo "</pre>";

echo "<hr/>";

echo "<h3>ASORT()</h3>";

echo "<p>Associative Array to be sorted by value</p>";

$people = array("Peter" => 24, "Joe" => 44, "Glenn" => 21, "Cleveland" => 33, "Michael" => 41);

echo "<pre>";

print\_r($people);

echo "</pre>";

// sort people

asort($people);

echo "<p>Sorted Array</p>";

echo "<pre>";

print\_r($people);

echo "</pre>";

echo "<hr/>";

echo "<h3>KSORT()</h3>";

echo "<p>Associative Array to be sorted by key</p>";

$people = array("Adarsh" => 24, "Raj" => 44, "Rajesh" => 21, "Rajeshwari" => 33, "Rajeshwary" => 41,"A Dog" => 11);

echo "<pre>";

print\_r($people);

echo "</pre>";

// sort people

ksort($people);

echo "<p>Sorted Array</p>";

echo "<pre>";

print\_r($people);

echo "</pre>";

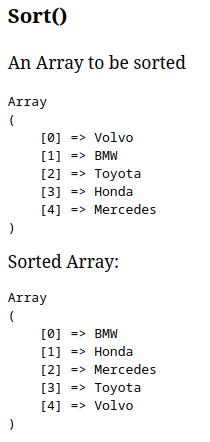
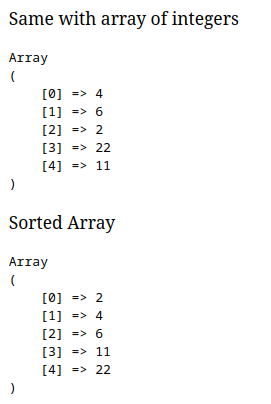
echo "<hr/>";

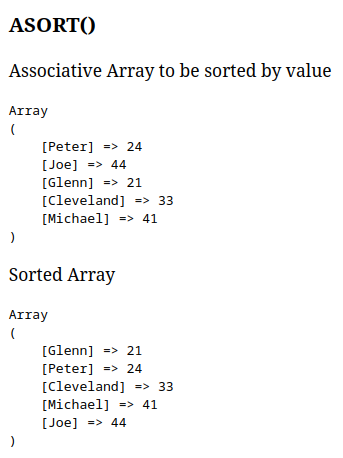
?>

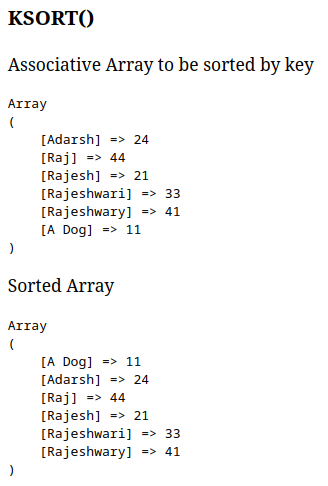
</body>

</html>

# Output of Program-6



****

****

**Program-7 |** Demonstrate how to find any item exist in an array or not.

• If exist, delete that particular item from the array.

• If exist, add a new item in the array just after that particular item.

<?php

session\_start();

// get items from session or initialize empty array.

$items = isset($\_SESSION['items']) ? $\_SESSION['items'] : array('Apple', 'Banana', 'Orange', 'Pear', 'Grape');

if(isset($\_POST['finddel'])){

$search = $\_POST['finddel'];

$key = array\_search($search, $items);

if($key !== false){

// delete item from array.

unset($items[$key]);

// save the array back to session.

$\_SESSION['items'] = $items;

echo "<strong>\* Item was found and deleted.</strong>";

} else {

echo "<strong>Item not found in array!</strong>";

}

}

if(isset($\_POST['newitem']) && isset($\_POST['search'])){

$search = $\_POST['search'];

$newItem = $\_POST['newitem'];

$key = array\_search($search, $items);

if($key !== false){

// put new item after the key.

array\_splice($items, $key + 1, 0, $newItem);

$\_SESSION['items'] = $items;

echo "<strong>\* Item found! New item added next to it.</strong>";

} else {

echo "<strong>Item not found in array!</strong>";

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<?php

echo "<h2>Items in the array:</h2>";

# print items in a list

echo '<ul>';

foreach ($items as $item) {

echo '<li>' . $item . '</li>';

}

echo '</ul>';

?>

<h2>Find and Delete item from array</h2>

<form action="/php-practicals/7.php" method="post">

<input type="text" name="finddel" placeholder="Search" required>

<input type="submit" name="submit" value="Search">

</form>

<h2>Find and Insert new item next to the array.</h2>

<form action="/php-practicals/7.php" method="post">

<input type="text" name="search" placeholder="Search" required>

<input type="text" name="newitem" placeholder="New Item" required>

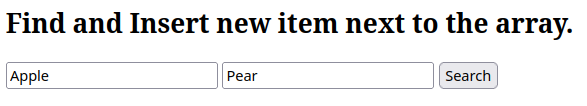
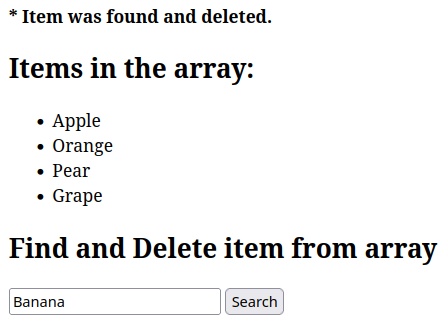
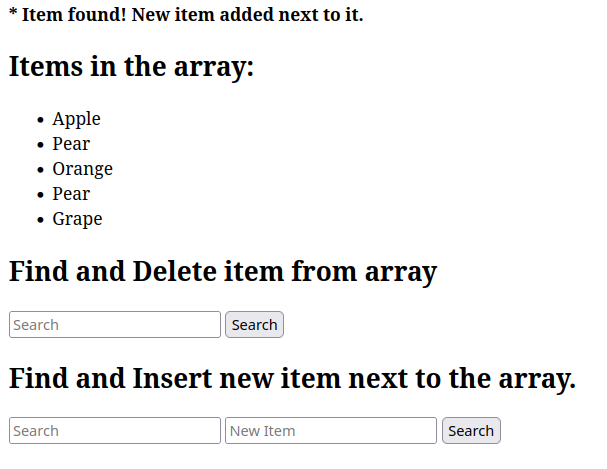
<input type="submit" name="submit" value="Search">

</form>

</body>

</html>

# Output of Program-7



**Program-8 |** Submit text value from a web page and store it in session provided the text value does not exist in session. Also display all the distinct text values stored in session.

<?php

// start session and store username and password in session

session\_start();

if(isset($\_POST['text']) && isset($\_POST['key'])){

$\_SESSION[$\_POST['key']] = $\_POST['text'];

echo "<strong>Session Saved</strong>";

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Store text in session</title>

</head>

<body>

<?php

# print all values stored in session

echo "<h2>Session Values</h2>";

echo "<ul>";

foreach ($\_SESSION as $key => $value) {

print\_r("<li>$key: $value</li>");

}

echo "</ul>";

?>

<form action="/php-practicals/8.php" method="post">

<input type="text" name="key" placeholder="key" required>

<input type="text" name="text" placeholder="Some text" required>

<button type="submit">Submit</button>

</form>

</body>

</html>

# Output of Program-8

