1. Write a function that creates a closure and returns a function that can add a specific number to any number passed to it. For example, if the closure is created with 5, the returned function should add 5 to any number passed to it.

**Solution:**

function additionNumbers(num1) {

return function(num2){

console.log(num1+num2)

}

}

let innerFunction=additionNumbers(5)

innerFunction(5)

2. Write a recursive function that searches an array for a specific value. The function should return true if the value is found, and false if it is not. You can assume that the array is not nested.

**Solution:**

function searchArray(array, value) {

// base case: array is empty

if (array.length === 0) {

return false;

}

// recursive case: search current element and rest of array

if (array[0] === value) {

return true;

} else {

return searchArray(array.slice(1), value);

}

}

let finalResult = searchArray([1,2,3,4,5,6,7],2)

console.log(finalResult)

3. Write a function that adds a new paragraph element to the bottom of an HTML document. The function should take a string argument that will be used as the text content of the new paragraph element.

**Solution:**

**App.js file**

function paragraphText(text){

//create an element p

const para = document.createElement("p");

//set id to p

para.setAttribute('id','paragraph1')

//add it to body

document.body.appendChild(para);

//get the p element

let p1 = document.getElementById("paragraph1");

//add the given text

p1.innerText=text;

console.log(p1)

}

paragraphText("Hello I am Aqsa");

**index.html file**

<!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>Assignment2</title>

<script src="./app.js" defer></script>

</head>

<body>

</body>

</html>

4. Write a function that adds a new list item to an unordered list in an HTML document. The function should take a string argument that will be used as the text content of the new list item.

**Solution:**

**app.js file**

function listText(text) {

//create an element listItem

const listItem = document.createElement("li");

//set id to listItem

listItem.setAttribute('id', 'listItem4')

//add it to the ul tag

let list = document.getElementById("List").appendChild(listItem);

//get the listItem element

let l1 = document.getElementById("listItem4");

//add the given text

l1.innerText = text;

console.log(l1)

}

listText("Grapes");

**index.html file**

<!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>Assignment2</title>

<script src="./app.js" defer></script>

</head>

<body>

<ul id="List">

<li id="listItem1">Apple</li>

<li id="listItem2">Banana</li>

<li id="listItem3">Strawberry</li>

</ul>

</body>

</html>

5. Write a function that changes the background color of an HTML element. The function should take two arguments: the first argument is a reference to the HTML element, and the second argument is a string representing the new background color.

**Solution:**

**app.js file**

function changeColor(referenceId,backgroundColor) {

let htmlElement = document.getElementById(referenceId);

htmlElement.style.backgroundColor=backgroundColor

console.log(htmlElement)

}

changeColor("paragraphh","yellow")

**index.html file**

<!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>Assignment2</title>

<script src="./app.js" defer></script>

</head>

<body>

<p id="paragraphh">Hello I am Aqsa</p>

</body>

</html>

6. Write a function that saves an object to localStorage. The function should take two arguments: the first argument is a string representing the key to use for storing the object, and the second argument is the object to store.

**Solution:**

let infoDetails = {

fullName: prompt("What is your full name?"),

speakEnglish: prompt("Can you speak English?")

}

let previousRecords = localStorage.getItem("info")

let info = previousRecords ? JSON.parse(previousRecords) : []

info.push(infoDetails)

function storeObject(keyName, objectName) {

// Here infoDetails is the key and objectName is the object

localStorage.setItem(keyName, objectName)

}

let convertString = JSON.stringify(info)

storeObject("info", convertString)

7. Write a function that retrieves an object from localStorage. The function should take one argument, which is a string representing the key used to store the object. The function should return the object.

**Solution:**

function getObject(keyName) {

const item = localStorage.getItem(keyName);

return item ? JSON.parse(item) : null

}

const info = getObject("info");

console.log(info);

8. Write a function that takes an object and saves each property to localStorage using the property name as the key and the property value as the value. The function should also retrieve the object from localStorage and return it as a new object.

**Solution:**

function saveObject(objectName) {

for (let keyName in objectName) {

localStorage.setItem(keyName, objectName[keyName]);

}

let newObject = {};

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

let key = localStorage.key(i);

let value = localStorage.getItem(key);

newObject[key] = value;

}

return newObject;

}

let myObject = {

fullName:prompt("What is your full name?"),

speakEnglish:prompt("Can you speak English?")

};

let savedObject = saveObject(myObject);

console.log(savedObject);