**CALLBACK FUNCTION IN JS:**

html>

<body style="text-align: right">

<h1>JavaScript Functions</h1>

<h2>Callback Functions</h2>

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

<script>

// Create an Array

const myNumbers = [4, 1, -20, -7, 5, 9, -6];

// Call removeNeg with a Callback

const posNumbers = removeNeg(myNumbers, (x) => x >= 0);

// Display Result

document.getElementById("demo").innerHTML = posNumbers;

// Remove negative numbers

function removeNeg(numbers, callback) {

const myArray = [];

for (const x of numbers) {

if (callback(x)) {

myArray.push(x);

}

}

return myArray;

}

</script>

</body>

</html>

In the example above, (x) => x >= 0 is a callback function.

It is passed to removeNeg() as an argument.

O/P:

Javascript Functions

Callback functions

4,1,5,9

**CALCULATOR:**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Calculator</title>

<link

href="https://fonts.googleapis.com/css?family=Roboto&display=swap"

rel="stylesheet"

/>

<link rel="stylesheet" href="./style.css" />

</head>

<body>

<div class="wrapper">

<section class="screen">

<div class="previous" data-operand-previous></div>

<div class="current" data-operand-current></div>

</section>

<section class="calc-btn\_row">

<div class="calc\_btn\_row">

<button class="calc\_btn double" data-all-clear>C</button>

<button class="calc\_btn" data-delete>←</button>

<button class="calc\_btn" data-operation>÷</button>

</div>

<div class="calc\_btn\_row">

<button class="calc\_btn" data-number>7</button>

<button class="calc\_btn" data-number>8</button>

<button class="calc\_btn" data-number>9</button>

<button class="calc\_btn" data-operation>×</button>

</div>

<div class="calc\_btn\_row">

<button class="calc\_btn" data-number>4</button>

<button class="calc\_btn" data-number>5</button>

<button class="calc\_btn" data-number>6</button>

<button class="calc\_btn" data-operation>-</button>

</div>

<div class="calc\_btn\_row">

<button class="calc\_btn" data-number>1</button>

<button class="calc\_btn" data-number>2</button>

<button class="calc\_btn" data-number>3</button>

<button class="calc\_btn" data-operation>+</button>

</div>

<div class="calc\_btn\_row">

<button class="calc\_btn" data-number>.</button>

<button class="calc\_btn double" data-number>0</button>

<button class="calc\_btn" data-equals>=</button>

</div>

</section>

</div>

<script src="./index.js"></script>

</body>

</html>

**Index.js**

**class** Calculator {

**constructor**(currentScreenTextElement, previousScreenTextElement) {

this.currentScreenTextElement = currentScreenTextElement;

this.previousScreenTextElement = previousScreenTextElement;

this.clear();

}

clear() {

this.currentOperand = "";

this.previousOperand = "";

this.operation = undefined;

}

delete() {

this.currentOperand = this.currentOperand.toString().slice(0, -1);

}

appendNumber(number) {

if (number === "." && this.currentOperand.includes(".")) return;

this.currentOperand = this.currentOperand.toString() + number.toString();

}

flushOperator(operation) {

if (this.currentOperand === "") return;

if (this.previousOperand !== "") {

this.compute();

}

this.operation = operation;

this.previousOperand = this.currentOperand;

this.currentOperand = "";

}

compute() {

**let** computation;

**const** previous = parseFloat(this.previousOperand);

**const** current = parseFloat(this.currentOperand);

if (isNaN(previous) || isNaN(current)) return;

switch (this.operation) {

case "+":

computation = previous + current;

break;

case "-":

computation = previous - current;

break;

case "×":

computation = previous \* current;

break;

case "÷":

computation = previous / current;

break;

default:

return;

}

this.currentOperand = computation;

this.previousOperand = "";

this.operation = undefined;

}

updateDisplay() {

this.currentScreenTextElement.innerText = this.currentOperand;

if (this.operation != null) {

this.previousScreenTextElement.innerText = `${this.previousOperand} ${this.operation}`;

}

}

}

**const** numberButtons = document.querySelectorAll("[data-number]");

**const** operationButtons = document.querySelectorAll("[data-operation]");

**const** equalsButton = document.querySelector("[data-equals]");

**const** deleteButton = document.querySelector("[data-delete]");

**const** allClearButton = document.querySelector("[data-all-clear]");

**const** currentScreenTextElement = document.querySelector(

"[data-operand-current]"

);

**const** previousScreenTextElement = document.querySelector(

"[data-operand-previous]"

);

**const** calculator = new Calculator(

currentScreenTextElement,

previousScreenTextElement

);

numberButtons.forEach((button) **=>** {

button.addEventListener("click", () **=>** {

calculator.appendNumber(button.innerText);

calculator.updateDisplay();

});

});

operationButtons.forEach((button) **=>** {

button.addEventListener("click", () **=>** {

calculator.flushOperator(button.innerText);

calculator.updateDisplay();

});

});

equalsButton.addEventListener("click", () **=>** {

calculator.compute();

calculator.updateDisplay();

});

allClearButton.addEventListener("click", () **=>** {

calculator.clear();

calculator.updateDisplay();

});

deleteButton.addEventListener("click", () **=>** {

calculator.delete();

calculator.updateDisplay();

});

**style.css**

html {

box-sizing: border-box;

}

\*,

\*:before,

\*:after {

box-sizing: inherit;

}

body {

background-color: #ebb5d3;

}

header {

display: flex;

flex-flow: row nowrap;

justify-content: space-between;

width: 80%;

margin: 0 auto;

*/\* border: 3px solid blue; \*/*

}

div.right {

padding: 10px;

}

.left h2 {

font-weight: bold;

font-size: 2.5rem;

font-family: "Ma Shan Zheng", cursive;

color: #000000;

margin: 25px 16px;

}

.left h2:hover {

color: white;

}

nav ul {

display: flex;

justify-content: flex-end;

margin: 0;

padding: 0;

list-style-type: none;

*/\* text-align: center; \*/*

}

nav li {

display: inline-block;

padding: 1.5rem;

}

nav a {

text-decoration: none;

color: #73352e;

display: inline;

padding: 0.5rem 1rem;

}

nav a:hover {

background-color: black;

color: #ffffff;

}

.wrapper {

max-width: 400px;

margin: 0 auto;

*/\* border: 5px solid red; \*/*

background-color: #000000;

}

.screen {

display: flex;

flex-flow: column nowrap;

background-color: #000000;

min-height: 80px;

padding: 10px;

font-size: 2.5rem;

font-weight: bold;

font-family: "Roboto", sans-serif;

text-align: right;

}

.screen .previous {

font-size: 1.5rem;

color: rgba(255, 255, 255, 0.75);

word-wrap: break-word;

word-break: break-all;

}

.screen .current {

font-size: 2.5rem;

color: #ffffff;

word-wrap: break-word;

word-break: break-all;

}

.calc\_btn\_row {

display: flex;

align-content: stretch;

justify-content: space-between;

margin-bottom: 0.5%;

}

.calc\_btn {

background-color: #d8d8d8;

color: black;

font-family: "Roboto", sans-serif;

font-weight: bold;

font-size: 2.5rem;

*/\* padding: 20px; \*/*

border: none;

height: 100px;

flex-basis: 24.5%;

cursor: pointer;

*/\* border-radius: 0; \*/*

}

.calc\_btn:hover {

background-color: #ebebeb;

}

.calc\_btn:active {

background-color: #bbbcbe;

}

.calc\_btn:last-child {

background-color: #df974c;

color: #ffffff;

}

.calc\_btn:last-child:hover {

background-color: #dfb07e;

}

.double {

flex-basis: 49.7%;

}

.tripple {

flex-basis: 74.8%;

}

@media only screen and (max-width: 300px) {

.wrapper {

max-width: 100%;

max-height: 100%;

}

}

**Form using js**

**index.html**

<html>

<head>

<title> Login Form</title>

</head>

<body>

<h3> LOGIN </h3>

<form = "Login\_form" onsubmit="submit\_form()">

<h4> FULL NAME</h4>

<input type="text" placeholder="Enter your Name">

<h4> USERNAME</h4>

<input type="email" placeholder="Enter your email id"/>

<h4> PASSWORD</h4>

<input type="password" placeholder="Enter your password"/></br></br>

<input type="submit" value="Login"/>

<input type="button" value="SignUp" onClick="create()"/>

</form>

<script type="text/javascript">

**function** submit\_form(){

alert("Login successfully");

}

**function** create(){

window.location="signup.html";

}

</script>

</body>

</html>

**Website visitor counter in JavaScript**

**Index.html**

<html>

<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>Doc</title>

<body>

<div id="counter"></div>

<script src="index.js"></script>

</body>

</head>

</html>

**Index.js**

**let counter=document.getElementById("counter")**

**let visitCount=localStorage.getItem("visitCount")**

**if(visitCount){**

**visitCount=Number(visitCount)+1**

**}**

**else{**

**visitCount=1**

**}**

**localStorage.setItem("visitCount", visitCount)**

**counter.innerHTML=visitCount**

**Functional Component:**

**App.js**

import React from 'react';

**function** Demo()

{

return <h1>Functional Component</h1>

}

Export default Demo;

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom';

import Example from './App';

ReactDOM.render(

<React.StrictMode>

<Example />

</React.StrictMode>,

document.getElementById('root')

);

**Class Component:**

**App.js**

import React from 'react';

**class** App **extends** React.Component{

render(){

return <h1>Class Component</h1>;

}

}

export default App;

**Index.js**

import React from 'react';

import ReactDOM from 'react';

import App from './App'

ReactDOM.render(

<React.StrictMode>

<App />

</React.StrictMode>,

document.getElementById('root')

)

**Use of Arrow function**

**Index.html**

<!DOCTYPE html>

<html>

<body>

<h1>Event Listener / Arrow Fucntion</h1>

<p>Prepares pdf for printing the page </p>

<p></p>

<button id="btn">Click Me!</button>

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

<script>

let print = "";

print = () => {

  document.getElementById("demo").innerHTML = window.print();

}

//A button object calls the function:

document.getElementById("btn").addEventListener("click", print);

</script>

</body>

</html>

**Event emitter**

**Eventemitter.js**

// Importing events

const EventEmitter = require('events');

// Initializing event emitter instances

var eventEmitter = new EventEmitter();

var geek1= (msg) => {

    console.log("Message from geek1: " + msg);

};

var geek2 = (msg) => {

    console.log("Message from geek2: " + msg);

};

// Registering geek1 and geek2

eventEmitter.on('myEvent', geek1);

eventEmitter.on('myEvent', geek1);

eventEmitter.on('myEvent', geek2);

// Removing listener geek1 that was

// registered on the line 13

eventEmitter.removeListener('myEvent', geek1);

// Triggering myEvent

eventEmitter.emit('myEvent', "Event occurred");

// Removing all the listeners to myEvent

eventEmitter.removeAllListeners('myEvent');

// Triggering myEvent

eventEmitter.emit('myEvent', "Event occurred");

**Modules in node.js**

**Index.js**

var myLogModule = require('./module')

myLogModule.name('Datta Meghe College of Engineering')

myLogModule.domain('B.E. in IT');

**Module.js**

var detail ={

    name: function (name){

        console.log('Name of college :' +name);

    },

    domain:function(domain){

        console.log('Degree:' +domain)

    }

};

module.exports = detail;

**Callback function**

**callback.js**

function readfile(){

    var fs =require("fs");

    var filedata =fs.readFileSync('input.txt');

    console.log(filedata.toString());

    console.log("End of Program execution");

}

readfile();

**input.txt**

This is a Nodejs Callback Function

**Calculator 2.0**

**index.html**

<!DOCTYPE html>

<html lang = "en">

<head>

<title> JavaScript Calculator </title>

<style>

h1 {

text-align: center;

padding: 23px;

background-color: skyblue;

color: white;

}

#clear{

width: 270px;

border: 3px solid gray;

border-radius: 3px;

padding: 20px;

background-color: red;

}

.formstyle

{

width: 300px;

height: 530px;

margin: auto;

border: 3px solid skyblue;

border-radius: 5px;

padding: 20px;

}

input

{

width: 20px;

background-color: green;

color: white;

border: 3px solid gray;

border-radius: 5px;

padding: 26px;

margin: 5px;

font-size: 15px;

}

#calc{

width: 250px;

border: 5px solid black;

border-radius: 3px;

padding: 20px;

margin: auto;

}

</style>

</head>

<body>

<h1> Calculator Program in JavaScript </h1>

<div class= "formstyle">

<form name = "form1">

*<!-- This input box shows the button pressed by the user in calculator. -->*

<input id = "calc" type ="text" name = "answer"> <br> <br>

*<!-- Display the calculator button on the screen. -->*

*<!-- onclick() function display the number prsses by the user. -->*

<input type = "button" value = "1" onclick = "form1.answer.value += '1' ">

<input type = "button" value = "2" onclick = "form1.answer.value += '2' ">

<input type = "button" value = "3" onclick = "form1.answer.value += '3' ">

<input type = "button" value = "+" onclick = "form1.answer.value += '+' ">

<br> <br>

<input type = "button" value = "4" onclick = "form1.answer.value += '4' ">

<input type = "button" value = "5" onclick = "form1.answer.value += '5' ">

<input type = "button" value = "6" onclick = "form1.answer.value += '6' ">

<input type = "button" value = "-" onclick = "form1.answer.value += '-' ">

<br> <br>

<input type = "button" value = "7" onclick = "form1.answer.value += '7' ">

<input type = "button" value = "8" onclick = "form1.answer.value += '8' ">

<input type = "button" value = "9" onclick = "form1.answer.value += '9' ">

<input type = "button" value = "\*" onclick = "form1.answer.value += '\*' ">

<br> <br>

<input type = "button" value = "/" onclick = "form1.answer.value += '/' ">

<input type = "button" value = "0" onclick = "form1.answer.value += '0' ">

<input type = "button" value = "." onclick = "form1.answer.value += '.' ">

*<!-- When we click on the '=' button, the onclick() shows the sum results on the calculator screen. -->*

<input type = "button" value = "=" onclick = "form1.answer.value = eval(form1.answer.value) ">

<br>

*<!-- Display the Cancel button and erase all data entered by the user. -->*

<input type = "button" value = "Clear All" onclick = "form1.answer.value = ' ' " id= "clear" >

<br>

</form>

</div>

</body>

</html**>**