**JavaScript Assignment-1**

1. **Differences Between Static Typing Language and Dynamic Typing Language:**

|  |  |
| --- | --- |
| Static Typing Language | Dynamic Typing Language |
| A programming language is said to use static typing when type checking is performed during compile-time as opposed to run-time. | A programming language is said to be dynamically typed, when the majority of its type checking is performed at run-time as opposed to at compile-time. |
| In static typing, types are associated with variables not values. | In dynamic typing, types are associated with values not variables. |
| In Static typing,variables need not be defined before they’re used. i.e., Explicitly declaration before employing them. | In Dynamic typing,variables must necessarily be defined before they are used.i.e., do not require the explicit declaration before they’re used. |
| Statically typed languages include Ada, C, C++, C#, JADE, Java, Fortran, Haskell, ML, Pascal. | Dynamically typed languages include JavaScript, Lisp, PHP, Prolog, Python, Ruby, Smalltalk |
| Static Typing Languages are less flexible. | Compared to static typing, dynamic typing can be more flexible |
| If a script written in a statically-typed language contains errors, it will fail to compile until the errors have been fixed | scripts written in dynamically-typed languagescan compile even if they contain errors that will prevent the script from running properly . |
| Ex: static int num, sum;  num = 5; | Ex: num = 10 |

1. **Difference Between let, var and const in JavaScript.**

|  |  |  |
| --- | --- | --- |
| **let** | **var** | **const** |
| The **let** statement declares local variable in a block scope. | var statement is used to declare a variable | **const**` means that the identifier can't be reassigned. |
| The scope of a variable defined with the keyword “let” is limited to the block defined by curly braces. | The scope of a variable defined with the keyword “var” is limited to the “function” within which it is defined. | The scope of a variable defined with the keyword “const” is limited to the block defined by curly braces. |
| Let is Block-scoped | Var is Function-scoped | Const is block-scoped |
| Let variables can be updated but not re-declared | Var variables can be updated and re-declared within its scope; | Const variables can neither be updated nor re-declared. |
| let variables are not initialized | variables are initialized with undefined, | Const variables are not initialized |
| let can be declared without being initialized, | Var can be declared without being initialized, | const must be initialized during declaration. |

1. **Create a calculator using javascript and html and css .**

**Calc.html:**

<!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>Calculator</title>

<link rel="stylesheet" type="text/css" href="Style1.css">

</head>

<body>

<section>

</div class="container">

<div id="display"> </div>

<div class="buttons">

<div class="button">C</div>

<div class="button">/</div>

<div class="button">\*</div>

<div class="button">&larr;</div>

<div class="button">7</div>

<div class="button">8</div>

<div class="button">9</div>

<div class="button">-</div>

<div class="button">4</div>

<div class="button">5</div>

<div class="button">6</div>

<div class="button">+</div>

<div class="button">1</div>

<div class="button">2</div>

<div class="button">3</div>

<div class="button">.</div>

<div class="button">(</div>

<div class="button">0</div>

<div class="button">)</div>

<div id="equal"class="button">=</div>

</div>

</div>

</section>

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

</body>

</html>

**Style1.css:**

#container{

max-width:0px;

margin:20px auto 30px auto;

box-shadow: 0px 0px 0px 0px rgba(0,0,0,0.2);

}

#display{

text-align:right;

height: 200px;

line-height:70px ;

padding :8px 8px;

font-size:18px;

/\* background-image:url(BG1.jpg); \*/

}

.buttons{

display: grid;

border-bottom: 1px solid #999;

border-left: 1px solid #999;

grid-template-columns: 1fr 1fr 1fr 1fr;

}

.buttons > div{

border-top: 1px solid rgb(11, 119, 241);

border-right:1px solid rgb(204, 19, 173);

}

.button{

border: 0.5px solid rgb(9, 207, 125);

line-height:100px;

text-align: center;

font-size:25px;

cursor: pointer;

}

#equal{

background-color: #11b8e2;

color: white;

}

.button:hover{

background-color: #b41bf0;

color: white;

transition: 0.5s ease-in-out;

}

**CalcJS.js:**

let display = document.getElementById('display');

let buttons=Array.from(document.getElementsByClassName('button'));

buttons.map(button =>{

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

switch(e.target.innerText){

case 'C':

display.innerText='';

break;

case '←':

if(display.innerText){

display.innerText=display.innerText.slice(0,-1);

}

break;

case '=':

try{

display.innerText=eval(display.innerText);

}

catch{

display.innerText='Error';

}

break;

default:

display.innerText += e.target.innerText;

}

});

});

1. **Registration Form:**

**Registration.html:**

<!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>Registration Page</title>

<link rel="stylesheet" type="text/css" href="Style.css">

</head>

<body>

<form class="box" action="Registration.html" method="POST">

<h1 style="font-size: 35px; color:rgb(241, 14, 33)">Registration Form </h1>

First-Name &nbsp; &nbsp; &nbsp; &nbsp; <input type="text" name=" "

placeholder="Enter First-Name" id="First-Name"/><br>

Last-Name &nbsp; &nbsp; &nbsp; &nbsp; <input type="text" name=" "

placeholder="Enter Last-Name" id="Last-Name"/><br>

Age &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;&nbsp;

<input type="text" name=" " placeholder="Enter Age" id="Age"/><br>

Email-Id &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; <input type="text" name="

" placeholder="Enter Email-Id" id="Email-Id"/><br>

&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;

&nbsp; &nbsp; &nbsp; &nbsp; &nbsp;

<button type="button" onclick="validate()">Submit</button>

<button type="reset" class="btnbtn-primary">Cancel</button></form>

<script>

function validate(){

var Firstname=document.getElementById("First-Name").value;

var Lastname=document.getElementById("Last-Name").value;

var age=document.getElementById("Age").value;

var Emailid=document.getElementById("Email-Id").value;

//let mfe=document.getElementById("Registration Form");

alert("Congrats");

console.log("HI");

}

</script>

</body>

</html>

**Style2.css:**

body{

margin: 0;

padding: 0;

font-family: sans-serif;

background-image: url('BG1.jpg');

background-size: cover;

}

.box{

width: 300px;

padding: 30px;

position: absolute;

top: 50%;

left: 50%;

transform: rgba(0,0,0,0.4);

text-align: center;

}

.box b1

{

color: white;

text-transform: uppercase ;

font-weight: 700;

}

.box input[ type="text"], .box input[type="text"],.box input[ type="text"], .box input[type="text"]

{

border: 0;

background: none;

display: block;

margin: 20px auto;

text-align: center;

border:3px solid orchid;

padding: 14px 10px;

width=220px;

outline: none;

color: plum;

border-radius: 24;

transition: o.25px

}

.box input[ type="text"]:focus, .box input[type="text"]:focus.box input[ type="text"],

.box input[type="text"]

{

width: 270px;

border-color: red;

}

.box input[type="Submit"]

{

border: 0;

background: none;

display: block;

margin: 20px auto;

text-align: center;

border:3px solid orchid;

padding: 14px 10px;

outline: none;

color: plum;

border-radius: 24;

transition: o.25px;

cursor: pointer;

}

.box input[type="Submit"]:hover{

background: rosybrown;

}