

**1. Construct a program in java script to generate random numbers and show the usage of for-in statement with the help of suitable example. Also discuss the features of java script.**

// Generate Random Number :-

```
function randomNum() {  
    return Math.floor((Math.random() * 100 + 1));  
}  
console.log("Random Number : ", randomNum());
```

// For-In loop :-

```
const student = {  
    fName: "Aryan",  
    lName: "Gupta",  
    age: 23  
}  
  
for (i in student) {  
    console.log(i + " : " + student[i]);  
}
```

Features of JavaScript :-

- i. Client Side Scripting : JavaScript is use at client side to write scripts for Webpages too make the interactive.
- ii. Versatile : JavaScript can perform a wide range of tasks from simple things like defining a onClick function for a button to complex Server Side applications.
- iii. Asynchronous : JavaScript is asynchronous in nature, which means that it does not block the User's workflow while perform another tasks like retrieving data from Database, making API calls, etc.
- iv. Dynamically typed : JavaScript is a dynamically typed language, that means type checks are performed at runtime, we can assign any type of data to any variable.
- v. Case Sensitive : JavaScript is a case sensitive language. Case differences can cause potential errors.
- vi. Object Oriented Programming : JavaScript supports the concept of OOPs.
- vii. Event Handling : JavaScript supports event handling by responding to various events like Button Pressed, Mouse Click, Keyboard Interrupts,etc...

- viii. DOM Manipulation : JavaScript is capable of performing DOM Manipulation operations. For Ex – selecting element based on tag name, id, class name, Deleting any children in the Document, creating new element, appending an element as child in Document, etc...
- ix. Rich Set Operators : JavaScript has a Rich set of operators, which help to perform various Arithmetic & Logical operations easily using JavaScript.
- x. Rich Ecosystem : JavaScript has rich ecosystem of Libraries & Framework like React.js, Vue.js, Next.js, Express.js, etc... which make JavaScript a perfect for development.

2. Create a page welcome message :
  1. Get the user name in the input field
  2. When the button is pressed add an event that get the user name and creates a welcome message on the page
  3. Add a check to ensure the length of the input is larger than 3 characters long.

Sample output:

What is you name?

Laurence Svekis

Submit

Welcome to the site Laurence Svekis

```
<html>
```

```
<head>
```

```
<title>Welcome Message</title>
```

```
<script defer>
```

```
function welcomeMessage() {  
  const name = document.getElementById("name").value;  
  if (name.length < 3) {  
    alert("Name can not be less than 3 characters long");  
  }  
  else {  
    const a = document.createElement("h1");  
    a.innerHTML = `Welcome to the site ${name}`;  
    document.body.appendChild(a);  
  }  
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<h1>What is your Name</h1>
```

```
<input type="text" name="name" id="name">
```

```
<button onclick="welcomeMessage()">Submit</button>
```

```
</body>
```

```
</html>
```

**3. List out the different ways an HTML element can be accessed in a JavaScript code. Write a JavaScript program to calculate multiplication and division of two numbers (input from user).**

HTML elements can be accessed in a JavaScript code using various functions, some of them are described below :-

- i. `getElementById(arg)` : `getElementById()` is used to select an element using the value of "id" attribute of the desired element. The "id" is provided to it as an argument. It always returns a single element, if found.

Syntax:

```
const element = document.getElementById("nameInput");
```

- ii. `getElementsByClassName(arg)` : `getElementsByClassName()` is used to select all the elements having a specific class. The name of the class is provided to it as an argument. It always returns an array / list of all elements, if one element has that class, the list will be blank.

Syntax:

```
const element = document.getElementsByClassName("darkBg");
```

- iii. `getElementsByTagName(arg)` : `getElementsByTagName()` is used to select all the elements of a specific Tag. The tag name is provided to it as an argument. It always returns an array / list of all elements, if no element has that class, the list will be blank.

Syntax:

```
const element = document.getElementsByTagName("h1");
```

- iv. `querySelector(arg)` : `querySelector()` is used to select the elements based using CSS query selectors. The query is passed to it as argument. It returns the first matching element that is found in the document.

Syntax:

```
const elements = document.querySelector(".darkBg");  
const element = document.querySelector("#nameInput");
```

- v. `querySelectorAll(arg)` : `querySelectorAll()` is used to select all the elements of a based on a CSS query. The query is passed to it as argument. It always returns an array / list of all elements, if no element is found, it return a blank list.

Syntax:

```
const elements = document.querySelectorAll(".darkBg");  
const element = document.querySelectorAll("ul");
```

Program to calculate multiplication and division of two numbers.

```
<html>
<head>
  <title>Calculator</title>
  <script>
    const a = parseInt(prompt("Enter num 1 : "), 10);
    const b = parseInt(prompt("Enter num 2 : "), 10);
    alert(`Product is : ${a * b}\nDivision is : ${a / b}`);
  </script>
</head>

<body>
</body>
</html>
```

**4. (a) Write a JavaScript function to remove specified number of characters from a string.**

```
<html>
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Remove characters</title>

  <script>
    const str = prompt("Enter the string");
    const pos = parseInt(prompt("Enter the position to remove characters from : "), 10);
    const n = parseInt(prompt("Enter the no. of characters to be removed : "), 10);
    const newStr = str.slice(0, pos - 1) + str.slice(pos + n - 1);

    alert(`New String is : ${newStr}`);
  </script>
</head>

<body>
</body>

</html>
```

**(b) Write a JavaScript function to check whether a string is blank or not.**

```
<html>

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Blank String</title>

  <script>
    const str = prompt("Enter a string");
    if (str.length == 0)
      alert("The string is blank");
    else
      alert("The string is not blank");
  </script>
</head>

<body>
</body>

</html>
```

**5. Build a form that will collect information from a visitor and send it to be processed by JavaScript in the page summary.html. The form should collect the following information: petName (the name of a pet), petType (which can be either Dog, or Cat, or Bird) and pedigreed (which is either true or false, and should start out checked). Your form should include a Submit button, and should demonstrate proper use of the <label> tag.**

```
<html>

<head>
  <title>Summary</title>

  <script>
    function collectInfo() {
      const petName = document.getElementById("petName").value;
      const petType = document.getElementById("petType").value;
      const radio = document.getElementsByName("pedigreed")[0];
      const pedigreed = document.forms[0].pedigreed.value;

      alert(`Pet Name : ${petName}\nPet Type : ${petType}\nPedigreed : ${pedigreed}`)
    }
  </script>
</head>

<body>

  <form>
    <span>
      Pet Name :
    </span>
    <input type="text" name="petName" id="petName">

    <br>

    <label for="petType">Pet Type : </label>
    <select name="petType" id="petType">
      <option value="Dog">Dog</option>
      <option value="Bird">Bird</option>
      <option value="Cat">Cat</option>
    </select>

    <br>

    <span>Pedigreed : </span>
```



```
<input type="radio" id="pediYes" , name="pedigreed" value="Yes" checked="checked">
<label for="pediYes">Yes</label>

<input type="radio" id="pediNo" , name="pedigreed" value="No">
<label for="pediNo">No</label>
<br>
<button onclick="collectInfo()">Submit</button>
</form>
</body>

</html>
```