Assignment Module-3

---

1. Date Object:
a. Create a function that takes a date in the format "YYYY-MM-DD" as a parameter and returns the day of the week for that date. The function should use the Date object.
b. Implement the function to display the day of the week for the current date.

Answer:

```
let currentDate = new Date();

let dayOfTheWeek =  cturrentDate.getDay();

console.log(dayOfTheWeek);
```

---

2. Math Object:
a. Write a JavaScript function that takes an array of numbers as input and returns the square root of the sum of squares of all the numbers. Use the Math object to perform the necessary calculations.
b. Implement the function to display the square root of the sum of squares for an array of numbers.

Answer:

```
let numArray = [2, 5, 10, 20];

let x = 0;

numArray.forEach (y => {

    x += (Math.pow(y,2))

})

let squareRootOfSum = Math.sqrt(x);

console.log(squareRootOfSum);
```

---

3. Numbers:
a. Write a JavaScript function that takes a positive integer as input and returns true if it is a prime number, otherwise returns false.
b. Implement the function to check if a given positive integer is a prime number.

Answer:

```
Function primeNumber(num){
        for (let i = 2; i <= Math.sqrt(num); i++){
        if (num % i === 0){
        return false;
        }
}
Return true;
}
console.log (primeNumber(3));
```

---

4. Window Object:
a. Create a function that opens a new window with a specified URL and dimensions (width and height).
b. Implement the function to open a new window with the URL "https://www.example.com" and dimensions 800x600.

Answer:

```
function creatWindow(){
        window.open("https://www.example.com","_blank","width = 800, height = 600")
        }
```

---

5. Navigator Object:

a. Write a JavaScript function that detects the user's browser name and version using the navigator object.
b. Implement the function to display the user's browser name and version.

Answer:

```
let browserName = navigator.appName;

let browserVersion = navigator.appVersion;

console.log (`Browser Name: ${ browserName }`);

console.log (`Browser Version: ${ browserName }`);
```

---

6. Geolocation:

a. Implement a JavaScript function that retrieves the user's current location (latitude and longitude) using the geolocation API.

Answer:

```
<html>
<head>
</head>
<body>
<button onclick="getLocation()">Location</button>
      <p id="demo"></p>
<script>
  var x = document.getElementById("demo");

function getLocation() {
 if (navigator.geolocation) {
   navigator.geolocation.getCurrentPosition(showPosition);
 } else {
   x.innerHTML = "Geolocation is not supported by this browser.";
 }
}

function showPosition(position) {
 x.innerHTML = "Latitude: " + position.coords.latitude +
 "<br>Longitude: " + position.coords.longitude;
```

```
}

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

---

7. JS Common Events:
Implement a webpage with the following functionality:
a. When the user clicks anywhere on the page, display an alert with the coordinates (x, y) of the click.
b. When the user presses any key on the keyboard, display an alert with the key code of the pressed key.
c. When the user moves the mouse over an image, change the image source to another image of your choice.

Answer:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <body>

      <img id="image" src="orginal photo.jpg" alt="Original Image">


      <script>
      document.addEventListener('DOMContentLoaded', function() {
          document.addEventListener('click', function(event) {
            var x = event.clientX;
            var y = event.clientY;
            alert('Clicked at coordinates (x, y): ' + x + ', ' + y);
          });
        });

      document.addEventListener('DOMContentLoaded', function() {
          document.addEventListener('keydown', function(event) {
            var keyCode = event.keyCode || event.which;
```

```
        alert('Key code: ' + keyCode);
      });
    });


  document.addEventListener('DOMContentLoaded', function() {
      var image = document.getElementById('image');

      image.addEventListener('mouseover', function() {
        image.src = "Tripty.jpg";
      });

      image.addEventListener('mouseout', function() {
        image.src = 'orginal photo.jpg';
      });
    });


  </script>
  </body>


</body>
</html>
```

---

8. HTML DOM Document:
a. Create a webpage with a button and a paragraph element.
b. Implement a JavaScript function that changes the text content of the paragraph element to "Button Clicked!" when the button is clicked.

Answer:

```
<!DOCTYPE html>
<html>
<head>
 <title>Button Click</title>
 <style>
   body {
     height: 100vh;
     display: flex;
     align-items: center;
     justify-content: center;
     font-family: Arial, sans-serif;
```

```
  }
  .container {
   text-align: center;
  }
  button {
   padding: 10px 20px;
   font-size: 16px;
  }
  p {
   margin-top: 20px;
   font-size: 18px;
   font-weight: bold;
  }
 </style>
</head>
<body>
 <div class="container">
  <button id="myButton">change the content</button>
  <p id="myParagraph">complete this assignment with the help of chatGPT</p>
 </div>

 <script>
  document.addEventListener('DOMContentLoaded', function() {
   var button = document.getElementById('myButton');
   var paragraph = document.getElementById('myParagraph');

   button.addEventListener('click', function() {
    paragraph.textContent = 'Button Clicked!';
   });
  });
 </script>
</body>
</html>
```

9. JS DOM Working with Form Input:
Implement a webpage with a form that has the following fields:
a. Name (text input)
b. Email (email input)
c. Password (password input)
d. Confirm Password (password input)
e. Submit (button)
Implement JavaScript validation for the form fields:
f. Name should not be empty and should contain only letters.
g. Email should be a valid email address.
h. Password should have a minimum length of 8 characters and contain at least one uppercase letter, one lowercase letter, and one digit.
i. Confirm Password should match the Password field.
j. Display appropriate error messages for invalid fields.

Answer:

```
<!DOCTYPE html>
<html>
<head>
  <title>Form Validation</title>

</head>
<body>
  <div>
    <form id="myForm">
      <label for="name">Name:</label><br>
      <input type="text" id="name" name="name"><br><br><br>

      <label for="email">Email:</label><br>
      <input type="email" id="email" name="email"><br><br><br>

      <label for="password">Password:</label><br>
      <input type="password" id="password" name="password"><br><br><br>

      <label for="confirmPassword">Confirm Password:</label><br>
      <input type="password" id="confirmPassword" name="confirmPassword"><br><br><br>

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

  <script>
```

```
  document.addEventListener('DOMContentLoaded', function() {
    var form = document.getElementById('myForm');

    form.addEventListener('submit', function(event) {
      event.preventDefault();
      var nameInput = document.getElementById('name');
      var emailInput = document.getElementById('email');
      var passwordInput = document.getElementById('password');
      var confirmPasswordInput = document.getElementById('confirmPassword');

      var name = nameInput.value.trim();
      var email = emailInput.value.trim();
      var password = passwordInput.value;
      var confirmPassword = confirmPasswordInput.value;

      var errors = [];

      // Name validation
      if (name === '') {
        errors.push('Name is required.');
      } else if (!/^[a-zA-Z]+$/.test(name)) {
        errors.push('Name should contain only letters.');
      }

      // Email validation
      if (email === '') {
        errors.push('Email is required.');
      } else if (!/\S+@\S+\.\S+/.test(email)) {
        errors.push('Email is invalid.');
      }

      // Password validation
      if (password === '') {
        errors.push('Password is required.');
      } else if (password.length < 8 || !/[a-z]/.test(password) || !/[A-Z]/.test(password) || !/[0-
9]/.test(password)) {
        errors.push('Password should have a minimum length of 8 characters and contain at least
one uppercase letter, one lowercase letter, and one digit.');
      }

      // Confirm Password validation
      if (confirmPassword === '') {
        errors.push('Confirm Password is required.');
      } else if (password !== confirmPassword) {
```

```
      errors.push('Confirm Password does not match Password.');
    }

    // Display error messages
    var errorContainer = document.getElementById('errors');
    errorContainer.innerHTML = '';
    if (errors.length > 0) {
      errors.forEach(function(error) {
        var errorElement = document.createElement('div');
        errorElement.className = 'error';
        errorElement.textContent = error;
        errorContainer.appendChild(errorElement);
      });
    } else {
      alert('Form submitted successfully!');
      form.reset();
    }
  });
 });
 </script>
</body>
</html>
```

---

10. DOM Manipulate CSS Class:
a. Create a webpage with a button and a paragraph element.
b. Implement a JavaScript function that adds a CSS class named "highlight" to the paragraph element when the button is clicked. The "highlight" class should change the background color of the paragraph to yellow.

Answer:

```
<!DOCTYPE html>
<html>
<head>
 <title>Button Click Highlight</title>
 <style>
  body {
   height: 100vh;
   display: flex;
   align-items: center;
   justify-content: center;
   font-family: Arial, sans-serif;
```

```
    }
    .container {
      text-align: center;
    }
    p {
      padding: 10px;
    }
    .highlight {
      background-color: yellow;
    }
    button {
      padding: 10px 20px;
      font-size: 16px;
      margin-top: 10px;
    }
  </style>
</head>
<body>
  <div>
    <p id="myParagraph">"ChatGPT can makeing life easier" </p>
    <button id="myButton">Highlight</button>
  </div>

  <script>
    document.addEventListener('DOMContentLoaded', function() {
      var button = document.getElementById('myButton');
      var paragraph = document.getElementById('myParagraph');

      button.addEventListener('click', function() {
        paragraph.classList.add('highlight');
      });
    });
  </script>
</body>
</html>
```

11. Create Element & Append Element:
a. Create a webpage with an empty unordered list (<ul>).
b. Implement a JavaScript function that dynamically creates five list items (<li>) with sequential numbers (1 to 5) and appends them to the unordered list.

Answer:

```
<!DOCTYPE html>
<html>
<head>
  <title>Dynamically Create List Items</title>
  <style>
    body {
      height: 100vh;
      display: flex;
      align-items: center;
      justify-content: center;
      font-family: Arial, sans-serif;
    }
    .container {
      text-align: center;
    }
    ul {
      list-style-type: none;
      padding: 0;
    }
    li {
      margin-bottom: 10px;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <div class="container">
    <ul id="myList"></ul>
  </div>

  <script>
    document.addEventListener('DOMContentLoaded', function() {
      var ul = document.getElementById('myList');

      for (var i = 1; i <= 5; i++) {
        var li = document.createElement('li');
```

```
      li.textContent = i;
      ul.appendChild(li);
    }
  });
 </script>
</body>
</html>
```

---

12. DOM Change Attribute Value:
a. Create a webpage with an image element (<img>) that has an initial source URL.
b. Implement a JavaScript function that changes the source URL of the image element when a button is clicked. Use a different image URL of your choice.

Answer:

```
<!DOCTYPE html>
<html>
<head>
 <title>Change Image Source</title>
 <style>
  body {
   height: 100vh;
   display: flex;
   align-items: center;
   justify-content: center;
   font-family: Arial, sans-serif;
  }
  .container {
   text-align: center;
  }
  img {
   width: 300px;
   height: 300px;
   object-fit: cover;
  }
  button {
   padding: 10px 20px;
   font-size: 16px;
   margin-top: 10px;
  }
 </style>
</head>
```

```
<body>
  <div class="container">
    <img id="myImage" src="orginal photo.jpg" alt="Initial Image">
    <button id="myButton">Change Image</button>
  </div>

  <script>
    document.addEventListener('DOMContentLoaded', function() {
      var image = document.getElementById('myImage');
      var button = document.getElementById('myButton');

      button.addEventListener('click', function() {
        image.src = 'Tripty.jpg';
        image.alt = 'Changed Image';
      });
    });
  </script>
</body>
</html>
```

---

13. DOM Query Selector:
a. Create a webpage with multiple paragraphs containing different classes and IDs.
b. Implement a JavaScript function that selects all paragraphs with a specific class and changes their text color to red.

Answer:

```
<!DOCTYPE html>
<html>
<head>
  <title>Change Text Color</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
    .paragraph {
      margin-bottom: 10px;
    }
    #special-paragraph {
      font-weight: bold;
    }
  </style>
```

```
</head>
<body>
  <p class="paragraph">This is paragraph 1.</p>
  <p class="paragraph">This is paragraph 2.</p>
  <p class="paragraph">This is paragraph 3.</p>
  <p class="paragraph">This is paragraph 4.</p>
  <p id="special-paragraph">This is a special paragraph.</p>

  <script>
    document.addEventListener('DOMContentLoaded', function() {
      var paragraphs = document.getElementsByClassName('paragraph');

      for (var i = 0; i < paragraphs.length; i++) {
        paragraphs[i].style.color = 'red';
      }
    });
  </script>
</body>
</html>
```

---

14. AJAX Get Request:
a. Implement a JavaScript function that sends an AJAX GET request to the following URL: "https://api.example.com/data".

Answer:

```
function sendGetRequest() {
    fetch('https://api.example.com/data')
      .then(response => {
        if (response.ok) {
          return response.json();
        } else {
          throw new Error('Request failed');
        }
      })
      .then(data => {
        console.log('Response:', data);
        // Process the response data here
      })
      .catch(error => {
        console.error('Error:', error);
        // Handle any errors that occurred during the request
```

```
    });
}
```

---

15. AJAX Post Request:
a. Implement a JavaScript function that sends an AJAX POST request to the following URL: "https://api.example.com/submit".
b. Send a JSON payload with the following data: { "name": "John Doe", "email": "johndoe@example.com" }.

Answer:

```
function sendPostRequest() {
  var url = 'https://api.example.com/submit';
  var data = {
    name: 'John Doe',
    email: 'johndoe@example.com'
  };

  fetch(url, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
  })
    .then(response => {
      if (response.ok) {
        return response.json();
      } else {
        throw new Error('Request failed');
      }
    })
    .then(data => {
      console.log('Response:', data);
          })
    .catch(error => {
      console.error('Error:', error);
          });
}
```
