

Advance javascript for frontend

1. Write a program to show alert.

Ans:

To show an alert to the user we can use javascript function `alert()` which is inbuilt function. We need to put alert in the script tag to use it to show alert to the user.

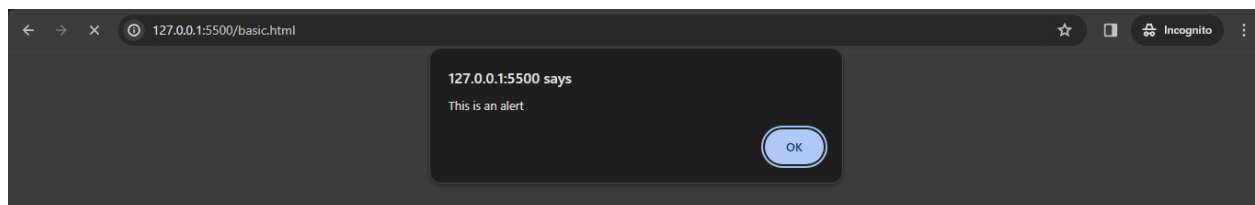
Code:

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

</body>

<script>
  alert("This is an alert");
</script>
</html>
```

Output:



2. What will be the output for these expressions

i) $5 > 4$

> The output will be `true` since 5 is greater than 4 and the datatype is also same.

```
> 5 > 4  
↳ true
```

ii) "apple" > "pineapple"

> Here, if we use greater than or lower than symbol to compare two strings then the browser will compare it as dictionary wise sequence, Here a is smaller than p so that's why it returns `false`

```
> "apple" > "pineapple"  
↳ false
```

iii) "2" > "12"

> Here again the data type are strings so the browser will compare it dictionary wise. 2 is greater than 1 that's why it returns true.

```
> "2" > "12"  
↳ true
```

iv) undefined == null

> the output will be true since undefined is also empty and null is also empty, also it's not comparing strictly.

```
> undefined == null  
↳ true
```

v) undefined === null

> Here it's strictly checking everything with === so it will return false, since undefined And null are both different thing

```
> undefined === null  
↳ false
```

vi) `null == "\n0\n"`

> Null means empty thing, but here we are comparing it with 0, so it will return false.

```
> null == "\n0\n"
< false
```

vii) `null === "\n0\n"`

> Again, it is comparing strictly so it will still return false because null means empty

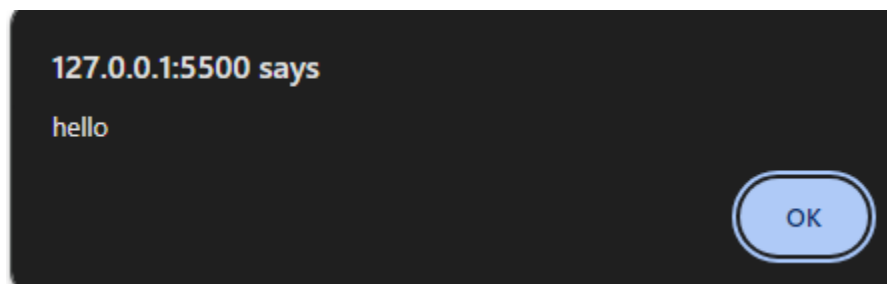
```
> null === "\n0\n"
< false
```

3. Will alert be shown?

```
if("0"){
    alert('hello');
}
```

Ans:

Yes, the alert will be shown, if the value inside parenthesis of if block is true then it will execute that code block.

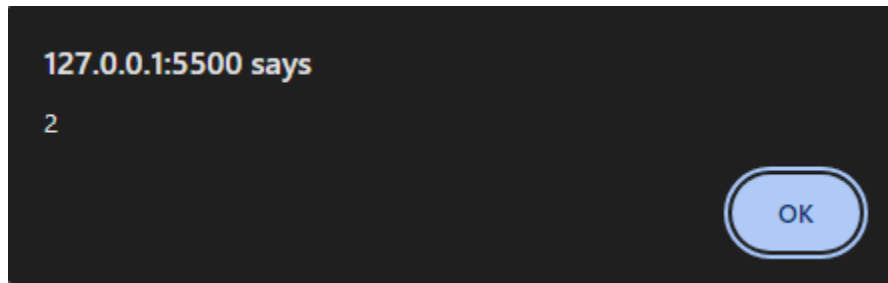


4. What is the code below going to output? `alert(null || 2 || undefined)`

Ans:

Here the above codes is doing or operating that whatever value is true that will be shown in the alert.

Null is empty and undefined is none so the code will alert 2.



5. The following function return true if the age is greater than 18 otherwise it confirms the user.

Code:

```
<!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>
  <script>
    let age = prompt("Enter your age: ")
    function checkAge(age){
      if(age > 18){
        console.log("You are eligible for driving.")
        return true;
      }
      else{
        confirm("Di you parents allowed you ?")
        return false;
      }
    }
    checkAge(age)
```

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

127.0.0.1:5500 says

Enter your age:

OK

Cancel

127.0.0.1:5500 says

Di you parents allowed you ?

OK

Cancel

127.0.0.1:5500 says

Enter your age:

OK

Cancel

You are eligible for driving.

>

6 . Replace functions with arrow functions in the code below

Data types and objects


1. Write the code, one line for each action

a) Create an empty object user

Code:

```
const user = {}
```

```
console.log(user)
```




main.js	  	Output
<pre>1 const user = {} 2 3 console.log(user)</pre>		<pre>node /tmp/6z12sd1ZYp.js {}</pre>

b) Add the property name with the value John.

Code:

```
const user = {  
  "Name": "John"  
}
```

```
console.log(user)
```

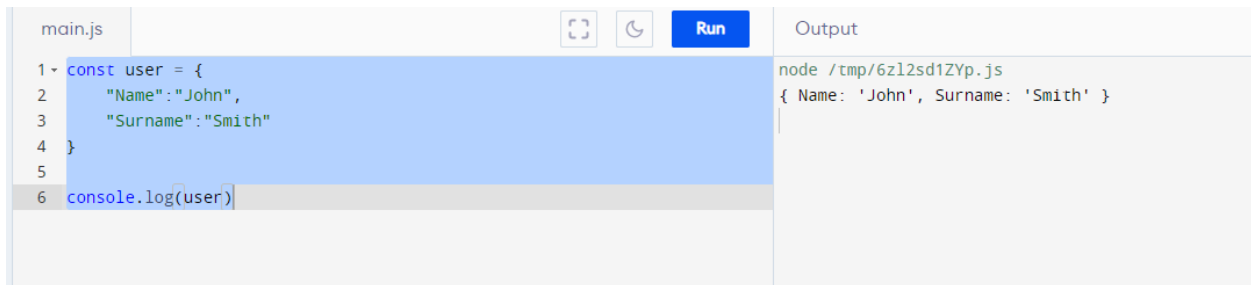
main.js	  	Output
<pre>1 const user = { 2 "Name": "John" 3 } 4 5 console.log(user)</pre>		<pre>node /tmp/6z12sd1ZYp.js { Name: 'John' }</pre>

c) Add the property name with the value Smith.

Code:

```
const user = {  
  "Name": "John",  
  "Surname": "Smith"  
}
```

```
console.log(user)
```



The screenshot shows a code editor with a file named 'main.js'. The code is as follows:

```
1 const user = {  
2   "Name": "John",  
3   "Surname": "Smith"  
4 }  
5  
6 console.log(user)
```

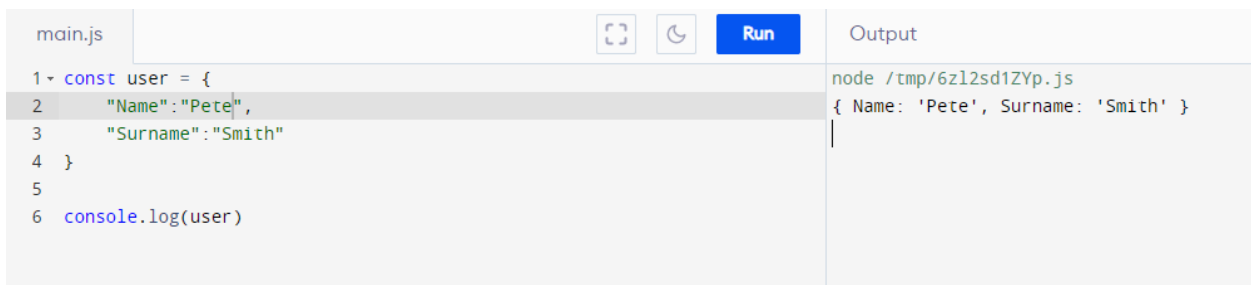
The code is highlighted in blue. To the right of the code editor is an 'Output' panel. It shows the command 'node /tmp/6z12sd1ZYp.js' and the output '{ Name: 'John', Surname: 'Smith' }'.

d) Change the value of name to Pete.

Code:

```
const user = {  
  "Name": "Pete",  
  "Surname": "Smith"  
}
```

```
console.log(user)
```



The screenshot shows a code editor with a file named 'main.js'. The code is as follows:

```
1 const user = {  
2   "Name": "Pete",  
3   "Surname": "Smith"  
4 }  
5  
6 console.log(user)
```

The code is highlighted in blue. To the right of the code editor is an 'Output' panel. It shows the command 'node /tmp/6z12sd1ZYp.js' and the output '{ Name: 'Pete', Surname: 'Smith' }'.

e) Remove the property name from the object.

Code:

```
const user = {  
  "Surname": "Smith"  
}  
  
console.log(user)
```

main.js	Output
<pre>1 const user = { 2 "Surname": "Smith" 3 } 4 5 console.log(user)</pre>	<pre>node /tmp/6z12sd1ZYp.js { Surname: 'Smith' }</pre>

2. Is array copied?

Ans:

Yes, the array is copied

Code:

```
let fruits = ["Apples", "Pear", "Oranges"]  
  
let shoppingCart = fruits;  
shoppingCart.push("Banana")  
  
console.log(fruits);  
console.log(shoppingCart)
```

Output:

main.js	Output
<pre>1 let fruits = ["Apples", "Pear", "Oranges"] 2 3 let shoppingCart = fruits; 4 shoppingCart.push("Banana") 5 6 console.log(fruits); 7 console.log(shoppingCart)</pre>	<pre>node /tmp/6z12sd1ZYp.js ['Apples', 'Pear', 'Oranges', 'Banana'] ['Apples', 'Pear', 'Oranges', 'Banana']</pre>

3. Map to names

let john = { name: "John", age: 25 }; let pete = { name: "Pete", age: 30 }; let mary = { name: "Mary", age: 28 }; let users = [john, pete, mary]; let names = /* ... your code */ alert(names);

Code:

```
let john = { name: "John", age: 25 };
let pete = { name: "Pete", age: 30 };
let mary = { name: "Mary", age: 28 };

let users = [ john, pete, mary ];

let names = users.map(item => item.name);

alert( names );
```

Output:

chrome://new-tab-page says

John,Pete,Mary



4. Map to objects let john = { name: "John", surname: "Smith", id: 1 }; let pete = { name: "Pete", surname: "Hunt", id: 2 }; let mary = { name: "Mary", surname: "Key", id: 3 }; let users = [john, pete, mary]; let usersMapped = /* ... your code ... */

Code:

```
let john = { name: "John", surname: "Smith", id: 1 };
let pete = { name: "Pete", surname: "Hunt", id: 2 };
let mary = { name: "Mary", surname: "Key", id: 3 };

let users = [ john, pete, mary ];

let usersMapped = users.map(user => ({
  fullName: `${user.name} ${user.surname}`,
  id: user.id
}));

/*
usersMapped = [
  { fullName: "John Smith", id: 1 },
  { fullName: "Pete Hunt", id: 2 },
  { fullName: "Mary Key", id: 3 }
]
*/

alert( usersMapped[0].id ); // 1
alert( usersMapped[0].fullName ); // John Smith
```

Output:

chrome://new-tab-page says

1

OK

chrome://new-tab-page says

John Smith

OK

5. Sum the properties There is a salaries object with arbitrary number of salaries. Write the function `sumSalaries(salaries)` that returns the sum of all salaries using `Object.values` and the `for..of` loop. If salaries is empty, then the result must be 0. `let salaries = { "John": 100, "Pete": 300, "Mary": 250 }; alert(sumSalaries(salaries)); // 650`

Code:

```
function sumSalaries(salaries) {  
    let sum = 0;  
    for (let salary of Object.values(salaries)) {  
        sum += salary;  
    }  
  
    return sum; // 650  
}  
  
let salaries = {  
    "John": 100,  
    "Pete": 300,  
    "Mary": 250  
};  
  
alert( sumSalaries(salaries) );
```

Output:

chrome://new-tab-page says

650

OK

6. Destructuring assignment We have an object: Write the Destructuring assignment that reads:
a) Name property into the variable name. b) Year's property into the variable age. c) isAdmin
property into the variable isAdmin (false, if no such property) d) let user = { name: "John", years:
30};

Code:

```
let user = {  
  name: "John",  
  years: 30  
};  
  
let {name, years: age, isAdmin = false} = user;  
  
alert( name );  
alert( age );  
alert( isAdmin );
```

Output:

chrome://new-tab-page says

John

OK

chrome://new-tab-page says

30

OK

chrome://new-tab-page says

false

OK

7. Turn the object into JSON and back Turn the user into JSON and then read it back into another variable. user = { name: "John Smith", age: 35};

Code:

```
let user = {  
  name: "John Smith",  
  age: 35  
};  
  
let user2 = JSON.parse(JSON.stringify(user));
```

Output:

```
▶ {name: 'John Smith', age: 35}  
undefined
```

MODULE: 3 (Document, Event and Controls)

1. Create a program to hide/show the password

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Password Toggle</title>
  <style>
    /* Styling for body */
    body {
      font-family: Arial, sans-serif;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh; /* Set height to 100% of viewport height */
      margin: 0; /* Remove default margin */
    }
    /* Styling for password container */
    .password-container {
      text-align: center; /* Center align text within container */
    }
  </style>
</head>
<body>

<!-- Container for password input and checkbox -->
<div class="password-container">
  <!-- Label for password input -->
  <label for="password">Password:</label>
  <!-- Password input field -->
  <input type="password" id="password" placeholder="Enter your password">
  <br>
  <!-- Checkbox to toggle password visibility -->
  <input type="checkbox" id="showPassword"> Show Password
</div>

<script>
  // Add event listener to the checkbox
  document.getElementById('showPassword').addEventListener('change', function() {
    // Get the password input element
    const passwordInput = document.getElementById('password');
    // Change input type to text if checkbox is checked, otherwise set it back to password
    passwordInput.type = this.checked ? 'text' : 'password';
  });
</script>

</body>
</html>
```

Output:

Password:
☐ Show Password

Password:
☒ Show Password

2. Create a program that will select all the classes and loop over and whenever i click the button the alert should show

Code:

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

  <!-- Three div elements with different classes -->
  <div class="class1">Class 1</div>
  <div class="class2">Class 2</div>
  <div class="class3">Class 3</div>

  <!-- Button to trigger the alert -->
  <button id="showAlert">Click me!</button>

  <script>
    // Add an event listener to the button
    document.getElementById('showAlert').addEventListener('click', function() {
      // Select all elements whose class attribute starts with "class"
      var elements = document.querySelectorAll('[class^="class"]');
      var classNames = [];

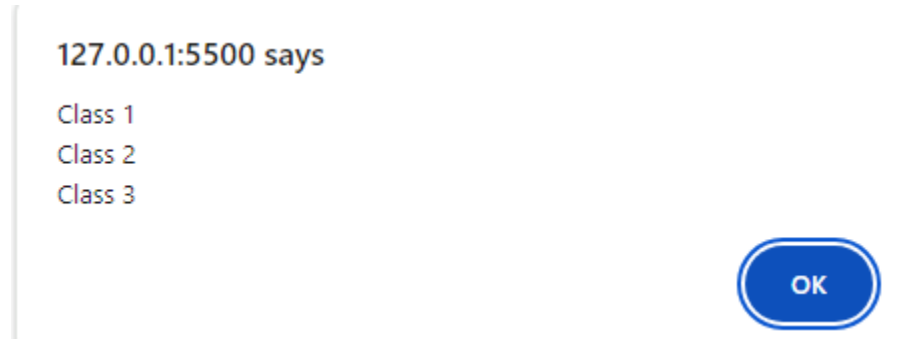
      // Loop through each selected element
      elements.forEach(function(element) {
        // Extract the text content of each element
        classNames.push(element.textContent);
      });

      // Show an alert with the concatenated text content of the selected elements
      alert(classNames.join('\n'));
    });
  </script>

</body>
</html>
```

Output:

Class 1
Class 2
Class 3



3. Create a responsive header using proper JavaScript

Code:

Output:



4. Create a form and validate using JavaScript

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Form Validation</title>
  <style>
    .error {
      color: red;
    }
  </style>
</head>
<body>

<h2>Registration Form</h2>

<form id="registrationForm">
  <div>
    <label for="username">Username:</label>
    <input type="text" id="username" name="username">
    <span id="usernameError" class="error"></span>
  </div>
  <div>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email">
    <span id="emailError" class="error"></span>
  </div>
  <div>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password">
    <span id="passwordError" class="error"></span>
  </div>
  <div>
    <label for="confirmPassword">Confirm Password:</label>
    <input type="password" id="confirmPassword" name="confirmPassword">
    <span id="confirmPasswordError" class="error"></span>
  </div>
  <div>
    <button type="submit">Submit</button>
  </div>
</form>
```

```

event.preventDefault(); // Prevent the form from submitting by default

// Get form inputs
const username = document.getElementById('username').value;
const email = document.getElementById('email').value;
const password = document.getElementById('password').value;
const confirmPassword = document.getElementById('confirmPassword').value;

// Regular expressions for validation
const usernameRegex = /^[a-zA-Z0-9_]{4,16}$/;
const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
const passwordRegex = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[a-zA-Z]).{8,}$/;

// Validation flags
let isValid = true;

// Validate username
if (!usernameRegex.test(username)) {
  document.getElementById('usernameError').textContent = 'Invalid username (4-16 characters, letters, numbers, underscore)';
  isValid = false;
} else {
  document.getElementById('usernameError').textContent = '';
}

// Validate email
if (!emailRegex.test(email)) {
  document.getElementById('emailError').textContent = 'Invalid email';
  isValid = false;
} else {
  document.getElementById('emailError').textContent = '';
}

// Validate password
if (!passwordRegex.test(password)) {
  document.getElementById('passwordError').textContent = 'Invalid password (at least 8 characters, one uppercase letter, one lowercase letter, one number)';
  isValid = false;
} else {
  document.getElementById('passwordError').textContent = '';
}

```

Output:

Registration Form

Username:

Email:

Password:

Confirm Password:

Registration Form

Username: Invalid username (4-16 characters, letters, numbers, underscore)

Email:

Password: Invalid password (at least 8 characters, one uppercase letter, one lowercase letter, one number)

Confirm Password:

5. Create a modal box using css and Js with three buttons

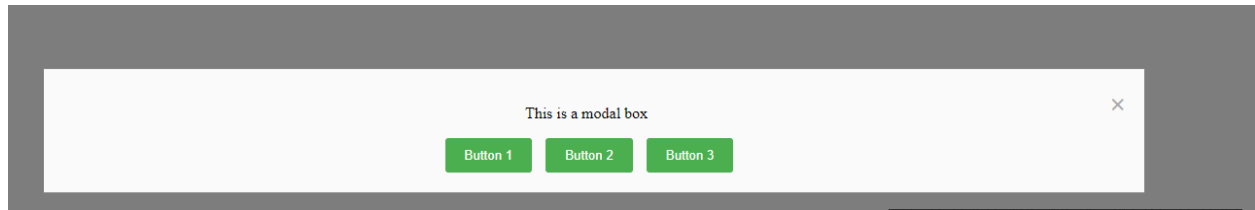
Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Modal Box</title>
  <style>
    /* Modal styles */
    .modal {
      display: none;
      position: fixed;
      z-index: 1;
      left: 0;
      top: 0;
      width: 100%;
      height: 100%;
      overflow: auto;
      background-color: rgba(0,0,0,0.5);
    }

    .modal-content {
      background-color: #fefefe;
      margin: 15% auto;
      padding: 20px;
      border: 1px solid #888;
      width: 80%;
      text-align: center;
    }

    /* Button styles */
    .button {
      padding: 10px 20px;
      background-color: #4CAF50;
      color: white;
      border: none;
      border-radius: 3px;
      cursor: pointer;
      margin-right: 10px;
    }
  </style>
</head>
<body>
  <div class="modal">
    <div class="modal-content">
      <div class="button">Button 1</div>
      <div class="button">Button 2</div>
      <div class="button">Button 3</div>
    </div>
  </div>
</body>
</html>
```

Output:



6. Use external js library to show slider

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Slider Example</title>
  <!-- Include slick-carousel CSS -->
  <link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/slick-carousel/slick/slick.css"/>
  <link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/slick-carousel/slick/slick-theme.css"/>
  <!-- Include your custom CSS if needed -->
  <style>
    .img{
      width: 30%;
    }
  </style>
</head>
<body>

<!-- Your HTML content goes here -->
<div class="slider">
  <div></div>
  <div></div>
  <div></div>
</div>

<!-- Include slick-carousel JavaScript -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/slick-carousel/slick/slick.min.js"></script>
```



```
<!-- Include slick-carousel JavaScript -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/slick-carousel/slick/slick.min.js"></script>
<script>
  // Initialize slick carousel
  $(document).ready(function(){
    $('.slider').slick({
      // Add your configuration options here
      autoplay: true,
      autoplaySpeed: 2000,
      dots: true,
      arrows: false
    });
  });
</script>

</body>
</html>
```

Output:



7.Prevent the browser when i click the form submit button

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Form Validation</title>
  <style>
    .error {
      color: red;
    }
  </style>
</head>
<body>

<h2>Registration Form</h2>

<form id="registrationForm">
  <div>
    <label for="username">Username:</label>
    <input type="text" id="username" name="username">
    <span id="usernameError" class="error"></span>
  </div>
  <div>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email">
    <span id="emailError" class="error"></span>
  </div>
  <div>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password">
    <span id="passwordError" class="error"></span>
  </div>
  <div>
    <label for="confirmPassword">Confirm Password:</label>
    <input type="password" id="confirmPassword" name="confirmPassword">
    <span id="confirmPasswordError" class="error"></span>
  </div>
  <div>
    <button type="submit" id="submitButton">Submit</button>
  </div>
</form>
```

```

script>
  document.getElementById('registrationForm').addEventListener('submit', function(event) {
    event.preventDefault(); // Prevent the default form submission behavior

    // Example validation: Check if username field is empty
    const username = document.getElementById('username').value;
    if (username.trim() === '') {
      document.getElementById('usernameError').textContent = 'Username is required';
      return; // Stop further execution if validation fails
    }

    alert('Form submitted successfully!');
  });
/script>

/body>
/html>

```

Output:

Registration Form

Username:
 Email:
 Password:
 Confirm Password:

127.0.0.1:5500 says
Form submitted successfully!

OK

MODULE: 4 (New Request)

1. What is JSON

Ans: JSON (JavaScript Object Notation) is a simple way to store and exchange data in a text format. It's commonly used in web development to send data between a server and a web browser.

2. What is promises

Ans: Promises in JavaScript are a way to handle asynchronous operations. They represent a value that may be available now, in the future, or never. Promises can be in one of three states: pending (initial state), fulfilled (operation completed successfully), or rejected (operation failed).

3. Write a program of promises and handle that promises also

Code:

```
<script>
// Function that returns a promise to simulate an asynchronous operation
function fetchData() {
  return new Promise((resolve, reject) => {
    // Simulating fetching data asynchronously
    setTimeout(() => {
      const data = Math.random(); // Simulated data
      if (data > 0.5) {
        resolve(data); // Resolve the promise with the data if successful
      } else {
        reject('Error: Data fetch failed'); // Reject the promise with an error message if failed
      }
    }, 1000); // Simulating a delay of 1 second
  });
}

// Using the promise
fetchData()
  .then(data => {
    console.log('Data fetched successfully:', data);
  })
  .catch(error => {
    console.error('Error fetching data:', error);
  });
</script>
```

Output:

✖ ▶ Error fetching data: Error: Data fetch failed

>

4. Use fetch method for calling an api <https://fakestoreapi.com/products>

Code:

```
<script>
// Using the fetch method to call the API
fetch('https://fakestoreapi.com/products')
  .then(response => {
    if (!response.ok) {
      throw new Error('Network response was not ok');
    }
    return response.json(); // Parse the response as JSON
  })
  .then(data => {
    console.log('Products:', data);
    // You can further process the data here
  })
  .catch(error => {
    console.error('Error fetching data:', error);
  });
</script>
```

Output:

```
Products:
▼ (20) [{"id": 1, "title": "Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops", "price": 109.95, "description": "Your perfect pack for everyday use and walks in the park. 15 inches in the padded sleeve, your everyday", "category": "men's clothing", "length": 20}, {"id": 2, "title": "Mens Casual Premium Slim Fit T-Shirts ", "price": 22.3, "description": "Slim-fitting style, contrast raglan long sleeve, three-button placket.", "category": "men's clothing", "length": 20}, {"id": 3, "title": "Mens Cotton Jacket", "price": 55.99, "description": "Great outerwear jackets for Spring/Autumn/Winter, ... and on in this Thanksgiving or Christmas Day.", "category": "men's clothing", "length": 20}, {"id": 4, "title": "Mens Casual Slim Fit", "price": 15.99, "description": "The color could be slightly different between on the product description.", "category": "men's clothing", "length": 20}, {"id": 5, "title": "John Hardy Women's Legends Naga Gold & Silver Dragon Station Chain Bracelet", "price": 695, "description": "From our Legends collection, the Naga was inspired by the mythical love and abundance, or outward for protection.", "category": "jewelry", "length": 20}, {"id": 6, "title": "Solid Gold Petite Micropave ", "price": 168, "description": "Satisfaction Guaranteed. Return or exchange any order within 30 days.", "category": "jewelry", "length": 20}, {"id": 7, "title": "White Gold Plated Princess", "price": 9.99, "description": "Classic Created Wedding Engagement Solitaire Diamond Promise, for wedding, anniversary, Valentine's Day...", "category": "jewelry", "length": 20}, {"id": 8, "title": "Pierced Owl Rose Gold Plated Stainless Steel Double", "price": 18.99, "description": "Rose Gold Plated Double Flared Tunnel Plug Earrings. Made of 316L Stainless Steel", "category": "jewelry", "length": 20}, {"id": 9, "title": "WD 2TB Elements Portable External Hard Drive - USB 3.0 ", "price": 64, "description": "USB 3.0 and USB 2.0 Compatibility Fast data transfer speeds* (hardware configuration required). Please note: For optimal performance, Windows 7 users are recommended to use Disk Management for new drives.", "category": "electronics", "length": 20}, {"id": 10, "title": "SanDisk SSD PLUS 1TB Internal SSD - SATA III 6 Gb/s", "price": 109, "description": "Easy upgrade for faster boot up, shutdown, application load and response. Up to 8TB of storage.", "category": "electronics", "length": 20}, {"id": 11, "title": "Silicon Power 64GB 128GB 256GB SSD 3D NAND A55 SLC Cache Performance Boost SATA III 2.5", "price": 109, "description": "3D NAND flash are applied to deliver high transfer speeds. Optimized performance and enhanced reliability.", "category": "electronics", "length": 20}, {"id": 12, "title": "WD 4TB Gaming Drive Works with Playstation 4 Portable External Hard Drive", "price": 114, "description": "Expand your PS4 gaming experience. Play anywhere. No need to compromise. 4TB of storage.", "category": "electronics", "length": 20}, {"id": 13, "title": "Acer SB220Q bi 21.5 inches Full HD (1920 x 1080) IPS Ultra-Thin", "price": 599, "description": "21.5 inches Full HD (1920 x 1080) widescreen IPS display. Vertical viewing angle, 178 degree. 75 hertz.", "category": "electronics", "length": 20}, {"id": 14, "title": "Samsung 49-Inch CH9000 Curved Gaming Monitor (LC49HG90DMUZD) - Super Ultrawide Screen QLED ", "price": 999.99, "description": "49 INCH SUPER ULTRAWIDE 32:9 CURVED GAMING MONITOR with ultra motion blur reduction. Free sync, G-Sync, HDR1000.", "category": "electronics", "length": 20}, {"id": 15, "title": "BEVELLESEN Women's 3-in-1 Snowboard Jacket Winter Coats", "price": 56.99, "description": "Note:The Jackets is US standard size. Please choose the size you need. Please choose the color you need. Please choose the size you need. Please choose the color you need.", "category": "clothing", "length": 20}, {"id": 16, "title": "Rock and Love Women's Removable Hooded Faux Leather Moto Biker Jacket", "price": 29.95, "description": "100% POLYURETHANE(shell) 100% POLYESTER(lining) 75%ASH ONLY // DO NOT BLEACH // LINE DRY // DO NOT IRON", "category": "clothing", "length": 20}, {"id": 17, "title": "Rain Jacket Women Windbreaker Striped Climbing Raincoats", "price": 39.99, "description": "Lightweight perfect for trip or casual wear---Long sleeve with elastic cuffs. Adjustable drawstrings give it a real styled look.", "category": "clothing", "length": 20}, {"id": 18, "title": "MBJ Women's Solid Short Sleeve Boat Neck V ", "price": 9.85, "description": "95% RAYON 5% SPANDEX. Made in USA or Imported. Do NOT wash in the dryer. Please choose the size you need. Please choose the color you need.", "category": "clothing", "length": 20}, {"id": 19, "title": "Opna Women's Short Sleeve Moisture", "price": 7.95, "description": "100% Polyester, Machine wash, 100% cationic polyesters, more feminine silhouette and Added Comfort", "category": "clothing", "length": 20}, {"id": 20, "title": "DAIRYDUY Womens T Shirt Casual Cotton Short", "price": 12.99, "description": "95% Cotton, 5% Spandex, Features: Casual, Short Sleeve, Home/Street. Season: Spring, Summer, Autumn, Winter.", "category": "clothing", "length": 20}], [{"prototype": "Array(0)"}]
```

5. Display all the product from the api in your HTML page

Code:

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

<h2>Product List</h2>

<ul id="productList">
  <!-- Product items will be added here dynamically -->
</ul>

<script>
  // Fetch products from the API
  fetch('https://fakestoreapi.com/products')
    .then(response => {
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      return response.json(); // Parse the response as JSON
    })
    .then(products => {
      const productList = document.getElementById('productList');

      // Loop through the products and create list items for each product
      products.forEach(product => {
        const listItem = document.createElement('li');
        listItem.textContent = `${product.title} - ${product.price}`; // Display product title and price
        productList.appendChild(listItem); // Add list item to the ul element
      });
    })
    .catch(error => {
      console.error('Error fetching data:', error);
    });
</script>

</body>
```

Output:

Product List

- Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops - \$109.95
- Mens Casual Premium Slim Fit T-Shirts - \$22.3
- Mens Cotton Jacket - \$55.99
- Mens Casual Slim Fit - \$15.99
- John Hardy Women's Legends Naga Gold & Silver Dragon Station Chain Bracelet - \$695
- Solid Gold Petite Micropave - \$168
- White Gold Plated Princess - \$9.99
- Pierced Owl Rose Gold Plated Stainless Steel Double - \$10.99
- WD 2TB Elements Portable External Hard Drive - USB 3.0 - \$64
- SanDisk SSD PLUS 1TB Internal SSD - SATA III 6 Gb/s - \$109
- Silicon Power 256GB SSD 3D NAND A55 SLC Cache Performance Boost SATA III 2.5 - \$109
- WD 4TB Gaming Drive Works with Playstation 4 Portable External Hard Drive - \$114
- Acer SB220Q bi 21.5 inches Full HD (1920 x 1080) IPS Ultra-Thin - \$599
- Samsung 49-Inch CHG90 144Hz Curved Gaming Monitor (LC49HG90DMNXZA) – Super Ultrawide Screen QLED - \$999.99
- BIYLACLESEN Women's 3-in-1 Snowboard Jacket Winter Coats - \$56.99
- Lock and Love Women's Removable Hooded Faux Leather Moto Biker Jacket - \$29.95
- Rain Jacket Women Windbreaker Striped Climbing Raincoats - \$39.99
- MJB Women's Solid Short Sleeve Boat Neck V - \$9.85
- Opna Women's Short Sleeve Moisture - \$7.95
- DANVOUY Womens T Shirt Casual Cotton Short - \$12.99