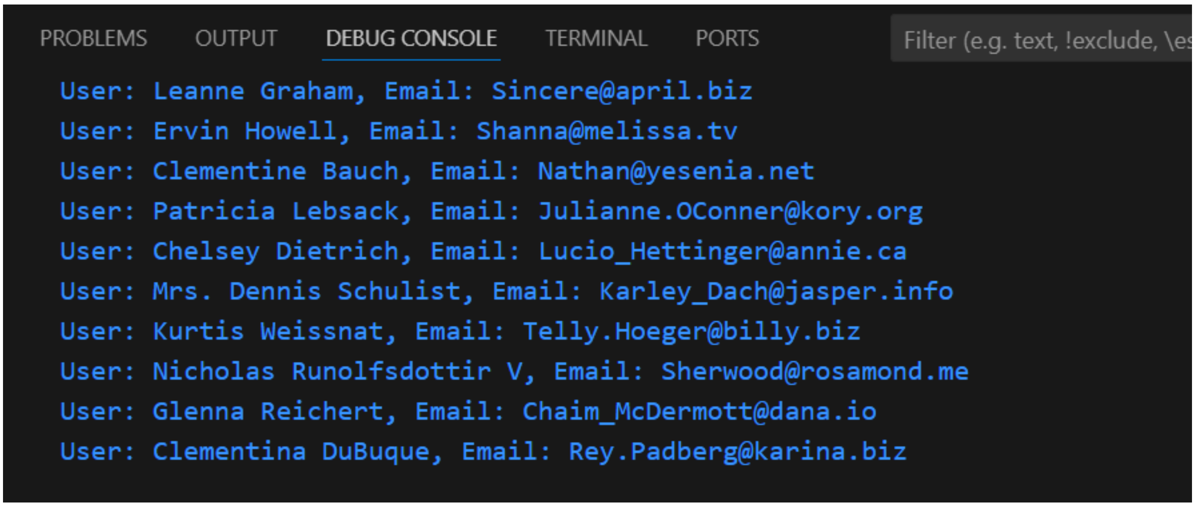
Task 2:

```
<!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>
        async function Data() {
            try {
                const response = await fetch('https://jsonplaceholder.typicode.com/users');
                if (!response.ok) {
                    throw new Error("Network response was not ok");
                }
                const data = await response.json();
                data.forEach(user => {
                    console.log("User: "+user.name+", Email: "+user.email);
                });
            } catch (error) {
                console.error("There was an error fetching the data:", error);
            }
        }
        Data();
    </script>
</body>
</html>
```

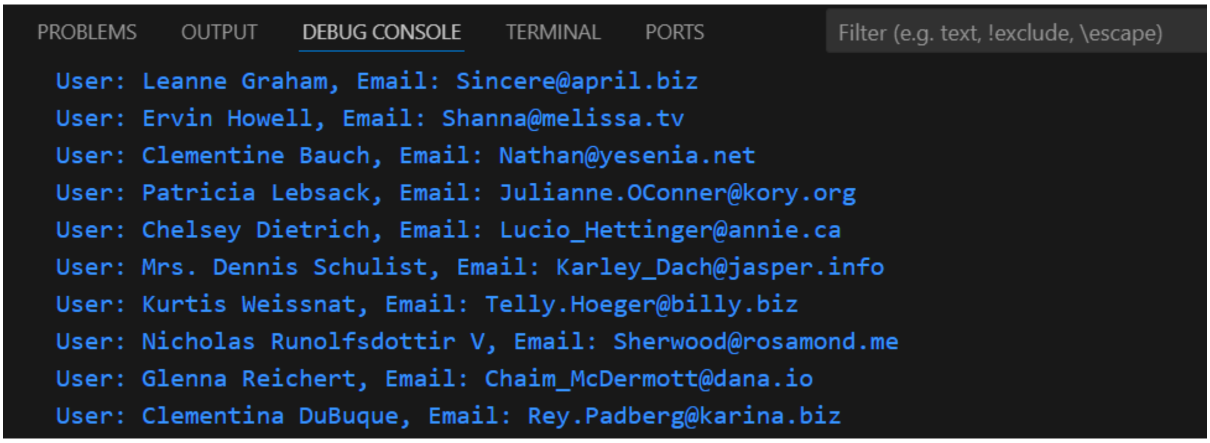


```
User: Leanne Graham, Email: Sincere@april.biz
User: Ervin Howell, Email: Shanna@melissa.tv
User: Clementine Bauch, Email: Nathan@yesenia.net
User: Patricia Lebsack, Email: Julianne.OConner@kory.org
User: Chelsey Dietrich, Email: Lucio_Hettinger@annie.ca
User: Mrs. Dennis Schulist, Email: Karley_Dach@jasper.info
User: Kurtis Weissnat, Email: Telly.Hoeger@billy.biz
User: Nicholas Runolfsdottir V, Email: Sherwood@rosamond.me
User: Glenna Reichert, Email: Chaim_McDermott@dana.io
User: Clementina DuBuque, Email: Rey.Padberg@karina.biz
```

Task 3:

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

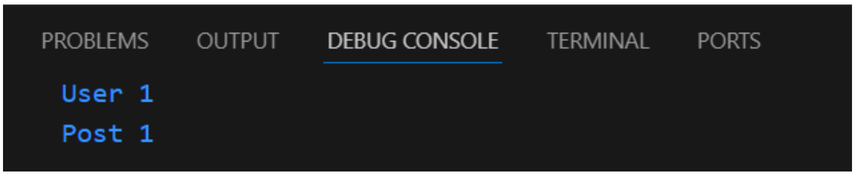
```
</head>
<body>
  <script>
    async function Data() {
      try {
        const response = await fetch('https://jsonplaceholder.typicode.com/users');

        if (!response.ok) {
          throw new Error("HTTP error! Status: "+(response.status));
        }

        const data = await response.json();
        data.forEach(user => {
          console.log("User: "+(user.name)+", Email: "+(user.email));
        });
      } catch (error) {
        console.error('There was an error fetching the data:', error.message);
      }
    }
    Data();
  </script>

</body>
</html>
```



```
User: Leanne Graham, Email: Sincere@april.biz
User: Ervin Howell, Email: Shanna@melissa.tv
User: Clementine Bauch, Email: Nathan@yesenia.net
User: Patricia Lebsack, Email: Julianne.OConner@kory.org
User: Chelsey Dietrich, Email: Lucio_Hettinger@annie.ca
User: Mrs. Dennis Schulist, Email: Karley_Dach@jasper.info
User: Kurtis Weissnat, Email: Telly.Hoeger@billy.biz
User: Nicholas Runolfsdottir V, Email: Sherwood@rosamond.me
User: Glenna Reichert, Email: Chaim_McDermott@dana.io
User: Clementina DuBuque, Email: Rey.Padberg@karina.biz
```

Task 4:

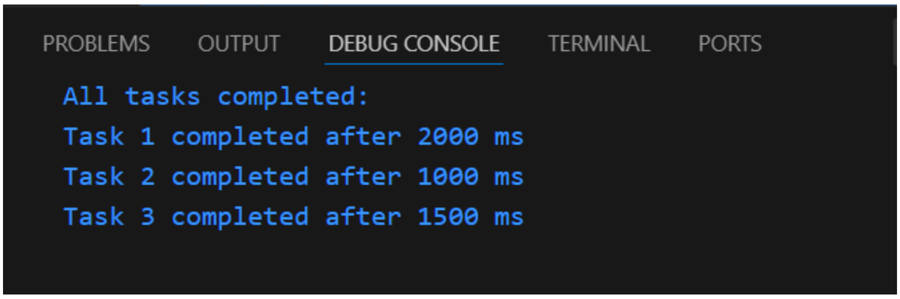
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    function GetUser(id) {
      return new Promise((resolve) => {
        setTimeout(() => {
                resolve("User " + id);
            }, 1000);
        });
    }
    function GetPost(id) {
        return new Promise((resolve) => {
            setTimeout(() => {
                resolve("Post " + id);
            }, 1500);
        });
    }
    async function GetData() {
        try {
            const [user, post] = await Promise.all([
                GetUser(1),
                GetPost(1)
            ]);
            console.log(user);
            console.log(post);
        } catch (error) {
            console.error("Error fetching data:", error);
        }
    }
    GetData();
  </script>
</body>
</html>
```



```
User 1
Post 1
```

Task 5:

```
<!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>
    function asyncTask(name, TimeTaken) {
        return new Promise((resolve) => {
            setTimeout(() => {
                resolve(`${name} completed after ${TimeTaken} ms`);
            }, TimeTaken);
        });
    }
    async function waitForAllTasks() {
        try {
            const results = await Promise.all([
                asyncTask("Task 1", 2000),
                asyncTask("Task 2", 1000),
                asyncTask("Task 3", 1500)
            ]);
            console.log("All tasks completed:");
            results.forEach((result) => console.log(result));
        } catch (error) {
            console.error("An error occurred:", error);
        }
    }
    waitForAllTasks();
  </script>
</body>
</html>
```



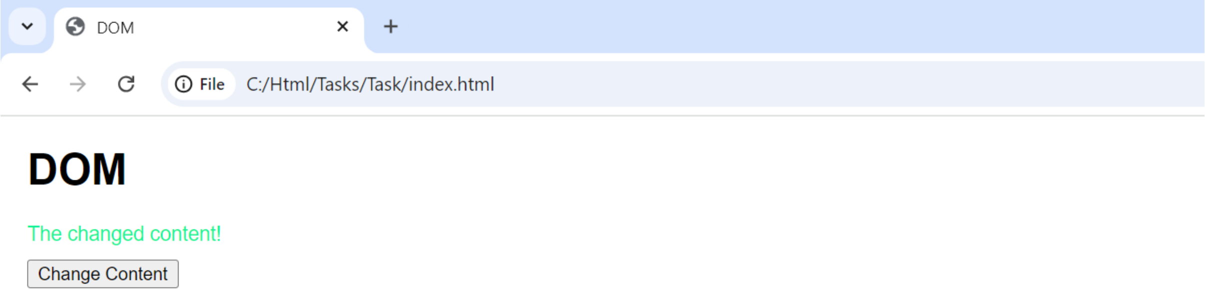
```
All tasks completed:
Task 1 completed after 2000 ms
Task 2 completed after 1000 ms
Task 3 completed after 1500 ms
```

7. Modules Introduction, Export and Import:

Task 1:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DOM</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        margin: 20px;
      }
      #message {
        font-size: 15px;
        color: rgb(16, 243, 149);
        margin: 10px 0;
      }
    </style>
  </head>
  <body>
    <h1>DOM</h1>
    <p id="message">The original content.</p>
    <button onclick="changeContent()">Change Content</button>
    <script>
```

```
      function changeContent() {
        var element = document.getElementById("message");
        element.textContent = "The changed content!";
      }
    </script>
  </body>
</html>
```

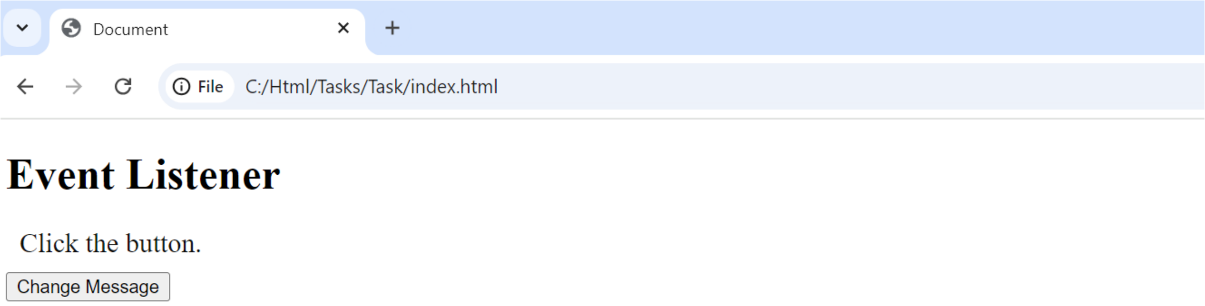


DOM

The changed content!

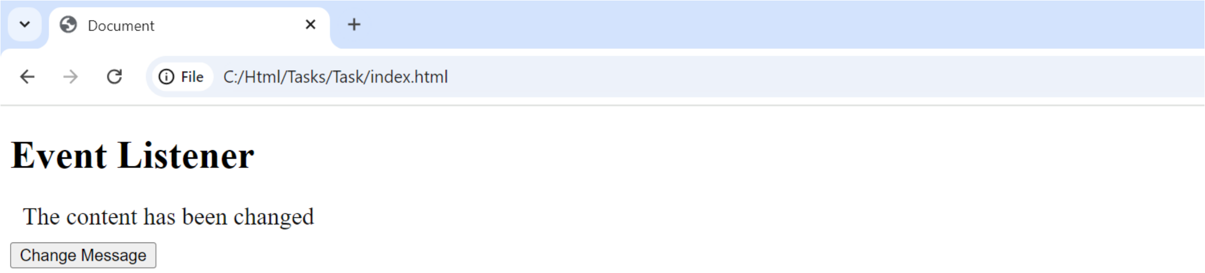
Task 2:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
      #message {
        font-size: 20px;
        color: rgb(17, 17, 17);
        margin: 10px;
      }
    </style>
  </head>
  <body>
    <h1> Event Listener </h1>
    <p id="message">Click the button.</p>
    <button id="changeMessageButton">Change Message</button>
    <script>
      const button = document.getElementById("changeMessageButton");
      const messageElement = document.getElementById("message");
      button.addEventListener("click", function() {
        messageElement.textContent = "The content has been changed ";
      });
    </script>
  </body>
</html>
```



Event Listener

Click the button.

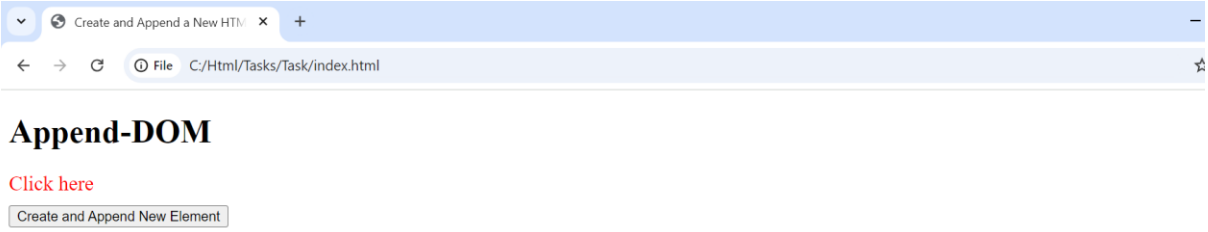


Event Listener

The content has been changed

Task 3:

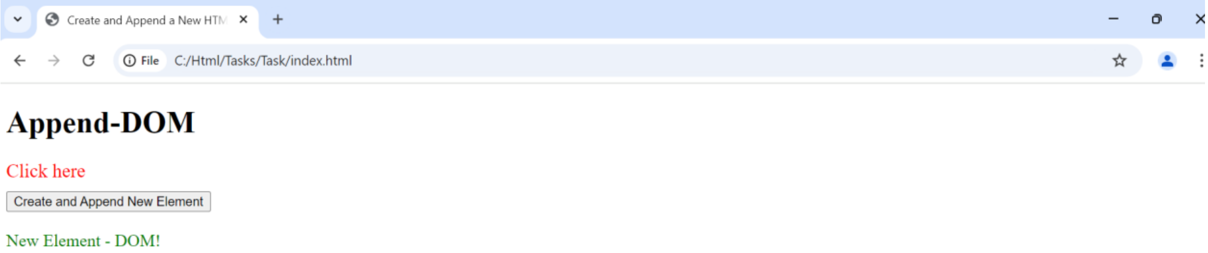
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Create and Append a New HTML Element</title>
    <style>
      #message {
        font-size: 20px;
        color: red;
        margin: 10px 0;
      }
      #newContainer {
        margin-top: 20px;
      }
    </style>
  </head>
  <body>

    <h1>Append-DOM</h1>
    <div id="message">Click here</div>
    <button id="createElementButton">Create and Append New Element</button>
    <div id="newContainer"></div>
    <script>
      const button = document.getElementById("createElementButton");
      const container = document.getElementById("newContainer");

      button.addEventListener("click", function() {
        const newElement = document.createElement("A");
        newElement.textContent = "New Element - DOM!";
        newElement.style.color = "green";
        newElement.style.fontSize = "18px";
        container.appendChild(newElement);
      });
    </script>
  </body>
</html>
```


Create and Append a New HTM
File C:/Html/Tasks/Task/index.html

Append-DOM

Click here

Create and Append New Element


Create and Append a New HTM
File C:/Html/Tasks/Task/index.html

Append-DOM

Click here

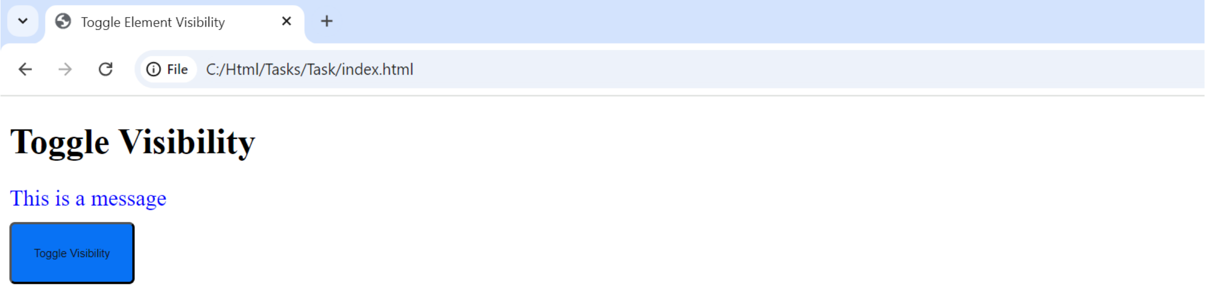
Create and Append New Element

New Element - DOM!

Task 4:

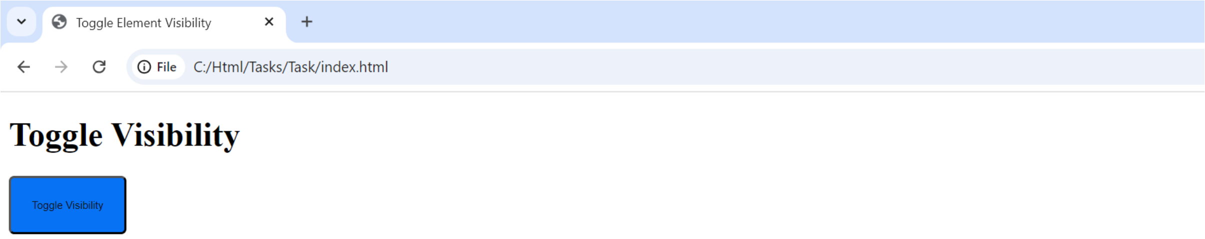
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Toggle Element Visibility</title>
    <style>
      #toggleMessage {
        font-size: 20px;
        color: blue;
        margin: 10px 0;
        display: block;
      }
      #toggleButton {
        padding: 20px 20px;
        font-size: 10px;
        background-color: rgb(8, 114, 244);
        color: rgb(17, 17, 17);
        border-radius: 5px;
      }
      #toggleButton:hover {
        background-color: rgb(243, 6, 46);
      }
    </style>
  </head>
  <body>
    <h1>Toggle Visibility</h1>
    <div id="toggleMessage">This is a message</div>
    <button id="toggleButton">Toggle Visibility</button>
    <script>
```

```
      const toggleButton = document.getElementById("toggleButton");
      const toggleMessage = document.getElementById("toggleMessage");
      function toggleVisibility() {
        if (toggleMessage.style.display === "none") {
          toggleMessage.style.display = "block";
        } else {
          toggleMessage.style.display = "none";
        }
      }
      toggleButton.addEventListener("click", toggleVisibility);
    </script>
  </body>
</html>
```



Toggle Visibility

This is a message

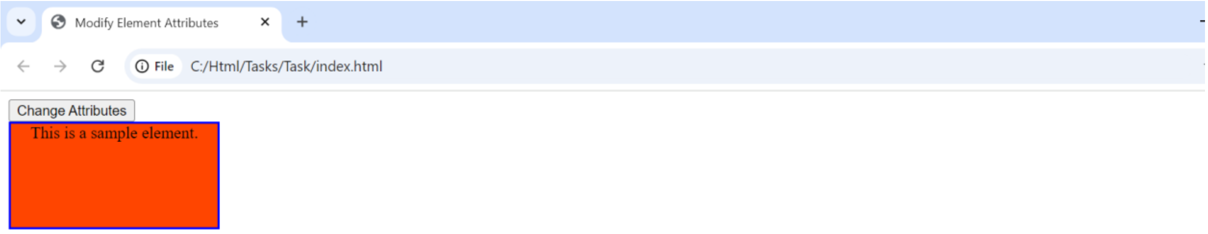


Toggle Visibility

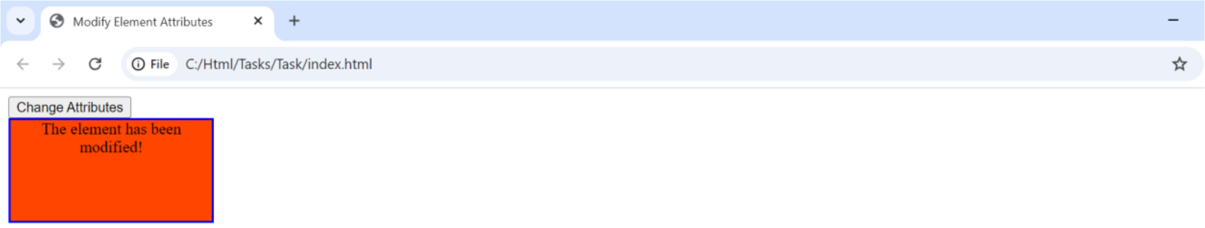
Task 5:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Modify Element Attributes</title>
  <style>
    #myElement {
      width: 200px;
      height: 100px;
      background-color: orangered;
      text-align: center;
      border: 2px solid blue;
    }
  </style>
</head>
<body>
  <button onclick="changeAttributes()">Change Attributes</button>
  <div id="myElement" class="box" title="Original Title">This is a sample element.</div>
  <script>
```

```
    function changeAttributes() {
      const element = document.getElementById("myElement");
      console.log("Current class:", element.getAttribute("class"));
      console.log("Current title:", element.getAttribute("title"));
      element.setAttribute("class", "modified-box");
      element.setAttribute("title", "Modified Title");
      element.textContent = "The element has been modified!";
    }
  </script>
</body>
</html>
```



This is a sample element.



The element has been modified!