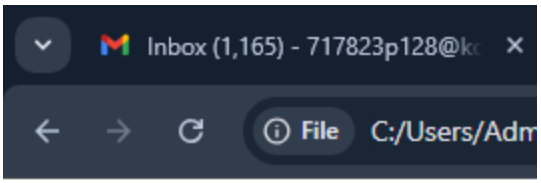
Promise, Promises chaining:

TASK 5

```
<html>
  <body>
    <script>
      function place(order) {
        return new Promise((resolve, reject) => {
          setTimeout(() => {
            document.writeln(`${order} order placed`);
            document.writeln("<br>");
            resolve(order);
          }, 1000)
        })
      }

      function preparefood(order) {
        return new Promise((resolve, reject) => {
          setTimeout(() => {
            document.writeln(`${order} food prepared`);
            document.writeln("<br>");
            resolve(order);
          }, 1000)
        })
      }

      function deliveryfood(order) {
        return new Promise((resolve, reject) => {
          setTimeout(() => {
            document.writeln(`${order} food delivered`);
            document.writeln("<br>");
            resolve(order);
          }, 1000)
        })
      }

      async function orderfood(fooditem) {
        const order = await place(fooditem);
        const preparedFood = await preparefood(order);
        const delivered = await deliveryfood(preparedFood);
        document.writeln("Order process complete!");
      }

      orderfood("Pizza");
    </script>
  </body>
</html>
```

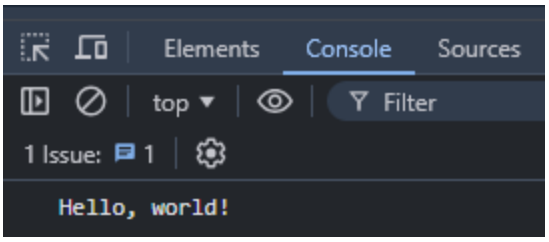


Pizza order placed
Pizza food prepared
Pizza food delivered
Order process complete!

Async/await:

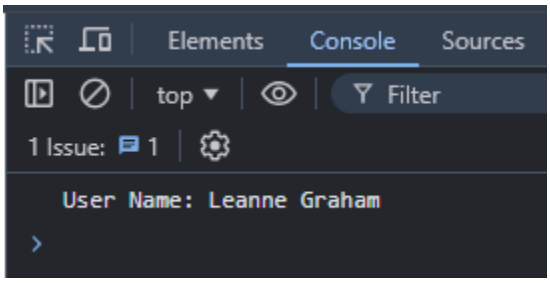
TASK 1

```
<html>
   <body>
      <script>
         async function fetchData() {
  return new Promise((resolve, reject) => {
   setTimeout(() => {
    const data = { message: 'Hello, world!' };
    resolve(data);
   }, 1000);
  });
}

async function main() {
  try {
   const data = await fetchData();
   console.log(data.message);
  } catch (error) {
   console.error(error);
  }
}

main();


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



TASK 2

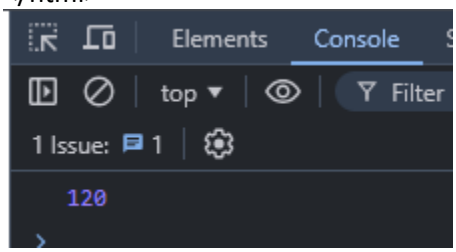
```
<html>
  <body>
    <script>
      async function fetchUserData(userId) {
 try {
  const response = await fetch(`https://jsonplaceholder.typicode.com/users/${userId}`);
  if (!response.ok) {
   throw new Error('Network response was not ok');
  }
  const userData = await response.json();

  console.log(`User Name: ${userData.name}`);
 } catch (error) {
  console.error('Error fetching data:', error);
 }
}

fetchUserData(1);

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



Recursion and stack:
TASK 1

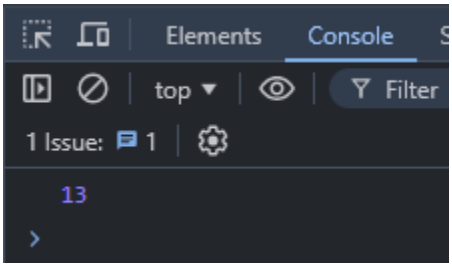
```
<html>
  <body>
    <script>
     function factorial(n){
```

```
        if(n === 0){
           return 1;
        }
        else{
           return n * factorial (n - 1);
        }
       }
console.log(factorial(5));
    </script>
  </body>
</html>
```



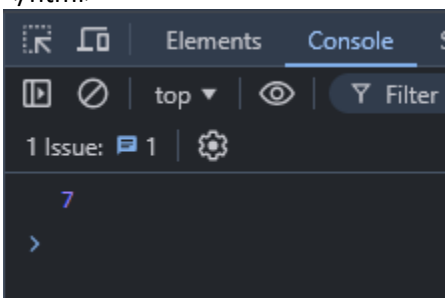
TASK 2

```
<html>
  <body>
    <script>
     function fibonacci(n){
      if(n === 0){
         return 0;
      }
      else if(n === 1){
         return 1;
      }
      else{
         return fibonacci(n - 1) + fibonacci(n - 2);
      }
     }
console.log(fibonacci(7));
    </script>
  </body>
</html>
```



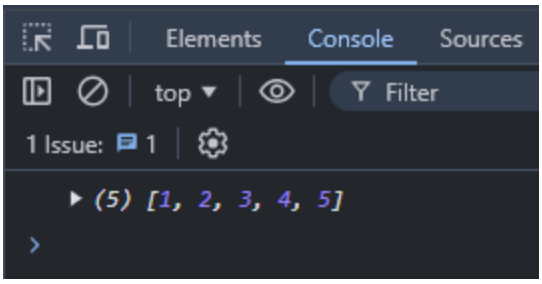
TASK 3

```
<html>
   <body>
      <script>
       function climb(n){
         if(n === 1){
            return 1;
         }
         else if(n === 2){
            return 2;
         }
         else if(n === 0){
            return 1;
         }
         else{
            return climb(n - 1) + climb(n - 2) + climb(n - 3);
         }
       }
console.log(climb(4));
      </script>
   </body>
</html>
```



TASK 4

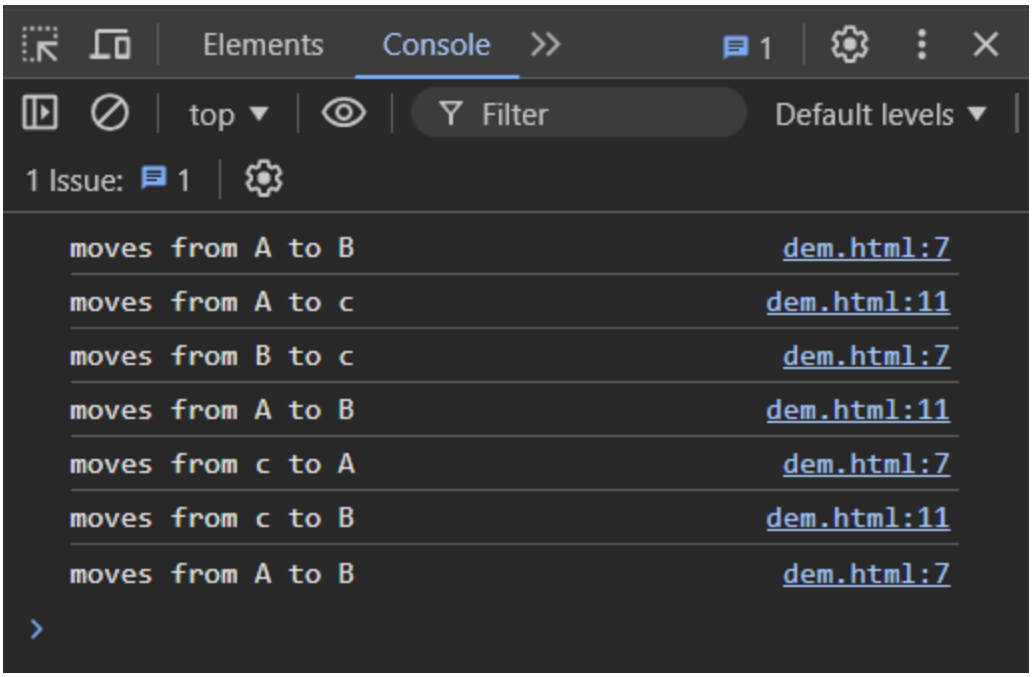
```
<html>
   <body>
      <script>
       function flattenArray(arr) {
  return arr.flat(Infinity);
}
console.log(flattenArray([1, [2, [3, [4]]], 5]));

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



Recursion and stack:

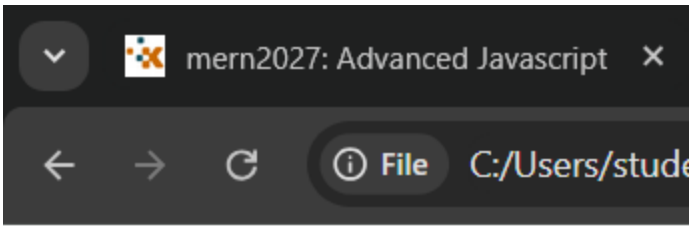
TASK 5

```
<html>
 <body>
  <script>
    function towerofHanoi(n,source,destination,auxillary)
    {
      if(n === 1){
        console.log(`moves from ${source} to ${destination}`);
        return;
      }
      towerofHanoi(n - 1,source,auxillary,destination);
      console.log(`moves from ${source} to ${destination}`);

      towerofHanoi(n - 1,auxillary,destination,source);
    }
    const n = 3;
      towerofHanoi(n,'A','B','c');
  </script>
 </body>
</html>
```



Browser: DOM Basics:
TASK 1

```
<!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>
  <h1 id="header">Headerr</h1>
  <button onclick="changecontent()">change the content</button>
  <script>
    function changecontent(){
      document.getElementById("header").textContent = "new textContent";
    }
  </script>
</body>
</html>
```

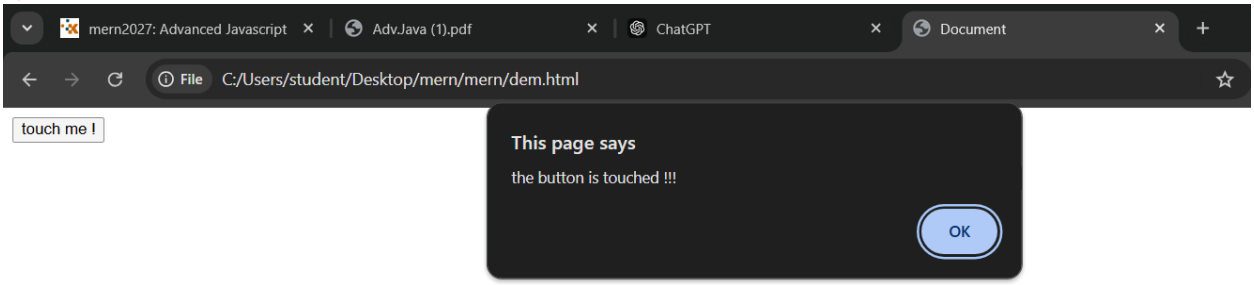


new textContent

change the content

TASK 2

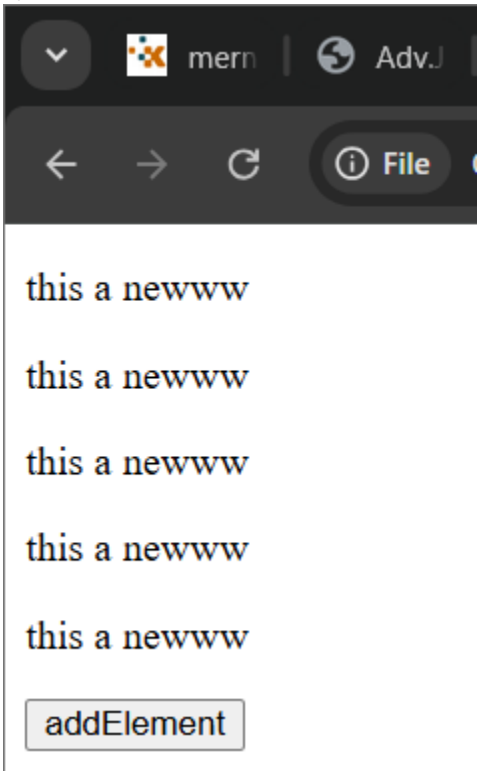
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <button  id="clickbutton">click me !</button>
   <script>
         document.getElementById("clickbutton").addEventListener("click",function(){

            alert("the button is clicked !!!")

         });

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



TASK 3

```
<!DOCTYPE html>
<html lang="en">
```

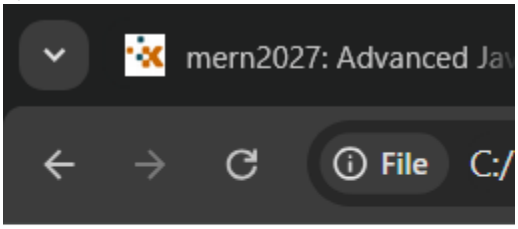
```
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <div id="container"></div>
   <button onclick="addElement()">addElement</button>
   <script>
         function addElement(){
            let newe = document.createElement("p");
            newe.textContent = "this a newww";

            document.getElementById("container").appendChild(newe);
         }
   </script>
</body>
</html>
```



TASK 4

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

```
  <title>Task 4</title>
</head>
<body>
  <div id="toggleElement">This is a toggled element!</div>
  <button onclick="toggleVisibility()">Toggle Visibility</button>

  <script>
    function toggleVisibility() {
      let element = document.getElementById("toggleElement");

      if (element.style.display === "none") {
        element.style.display = "block";
      } else {
        element.style.display = "none";
      }
    }
  </script>
</body>
</html>
```



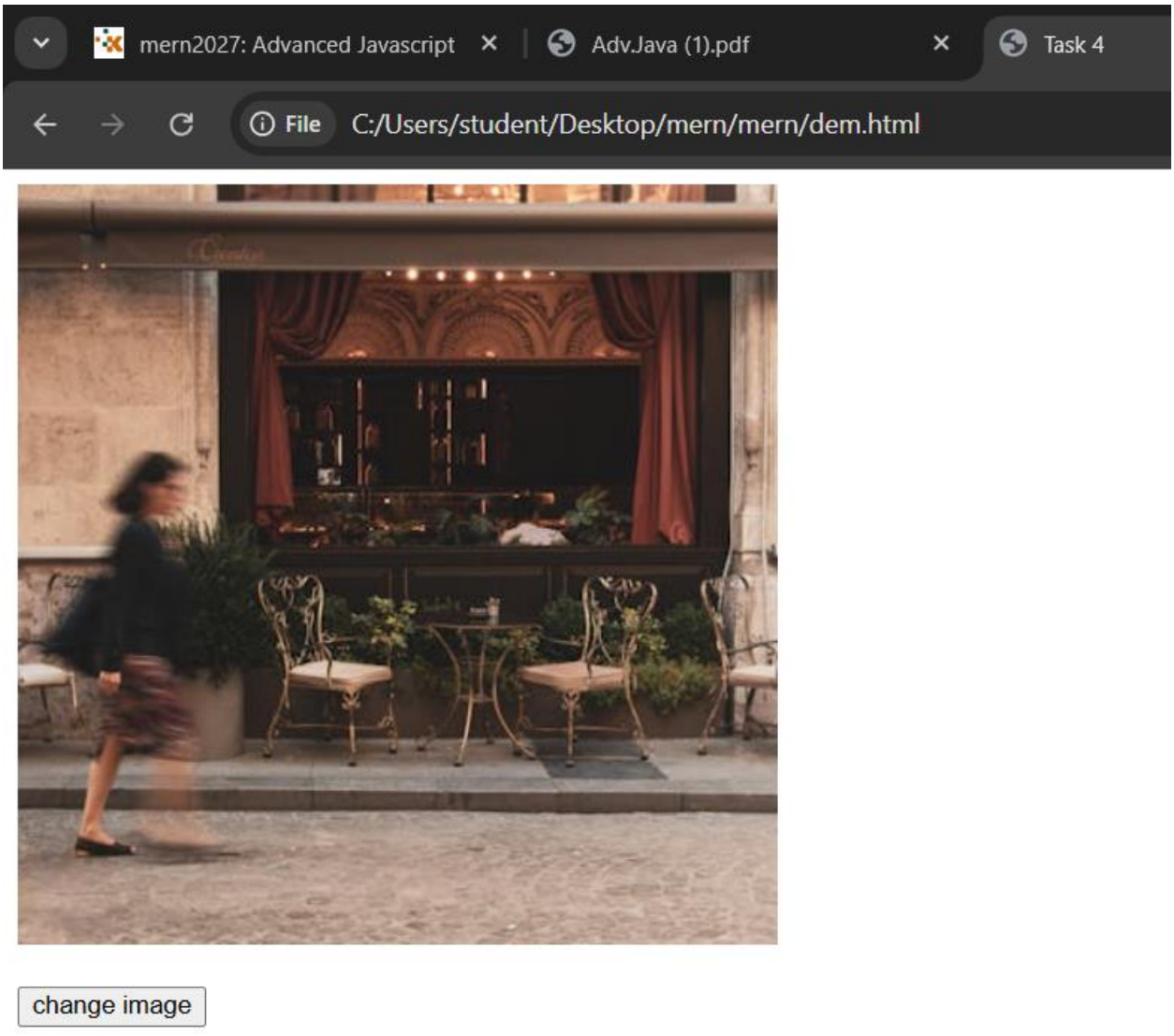
This is a toggled element!

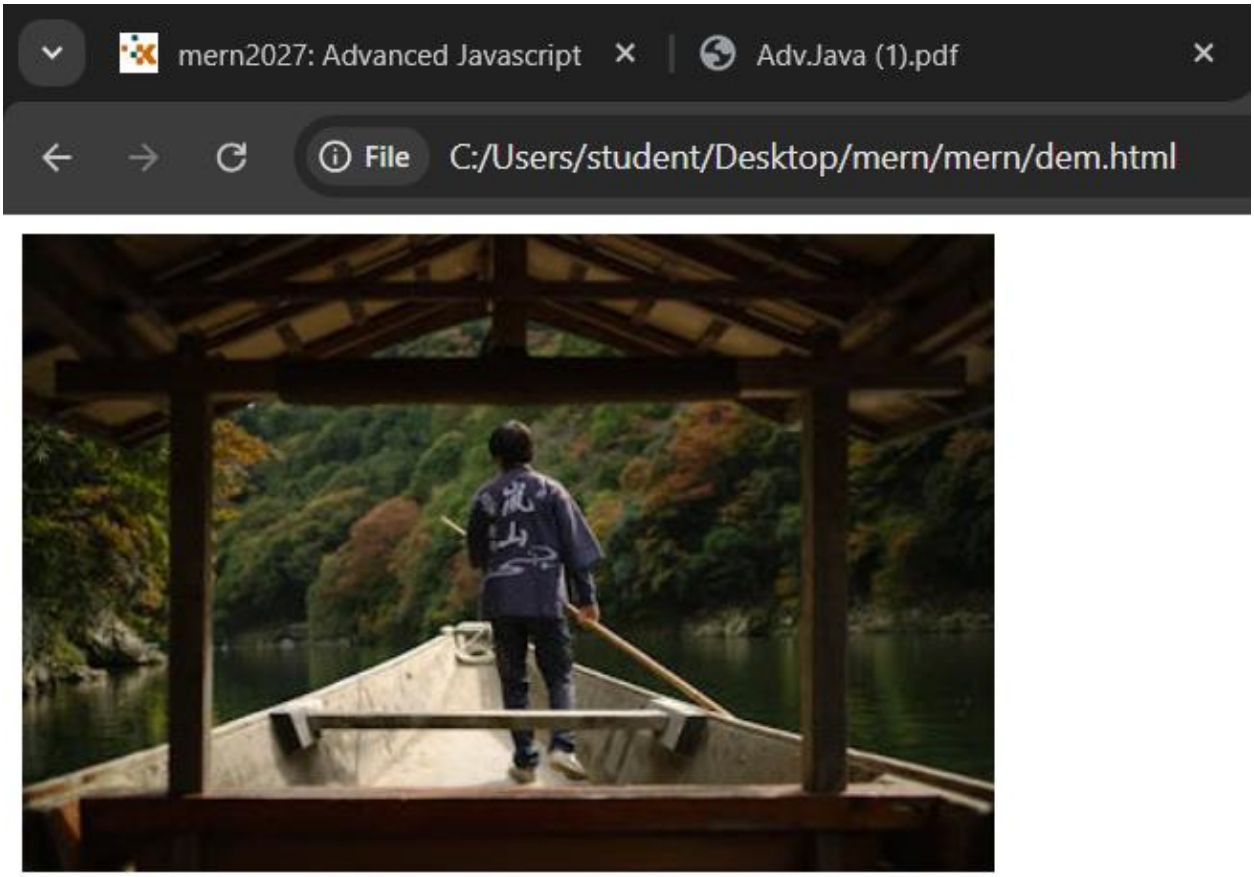
Toggle Visibility

TASK 5

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Task 4</title>
</head>
<body>
  <img id="image" src="https://images.pexels.com/photos/23996559/pexels-photo-23996559/free-photo-of-woman-walking-in-front-of-restaurant.jpeg?auto=compress&cs=tinysrgb&w=400&lazy=load" alt="Placeholder">
  <br><br><button onclick="changeimg()">change image</button>

  <script>
    function changeimg() {
      let image = document.getElementById("image");
```

```
    image.setAttribute("src","https://images.pexels.com/photos/15154328/pexels-photo-15154328/free-photo-of-a-person-standing-on-the-boat.jpeg?auto=compress&cs=tinysrgb&w=400&lazy=load");
    image.setAttribute("alt","neww");
    console.log(image.src);
    console.log(image.alt);
   }
  </script>
</body>
</html>
```




mern2027: Advanced Javascript
Adv.Java (1).pdf
File
C:/Users/student/Desktop/mern/mern/dem.html
change image