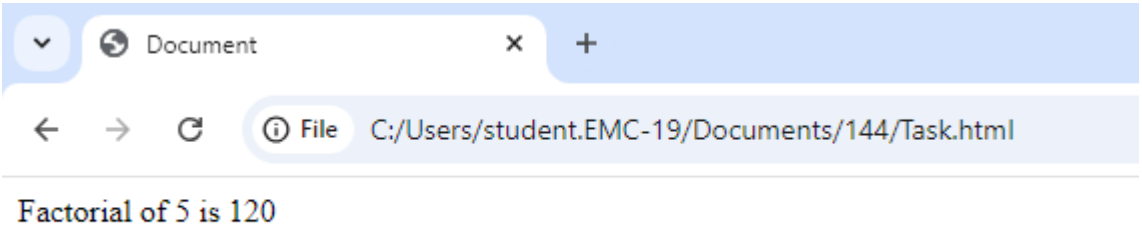
Advanced JavaScript

1. Recursion:

Task 1:

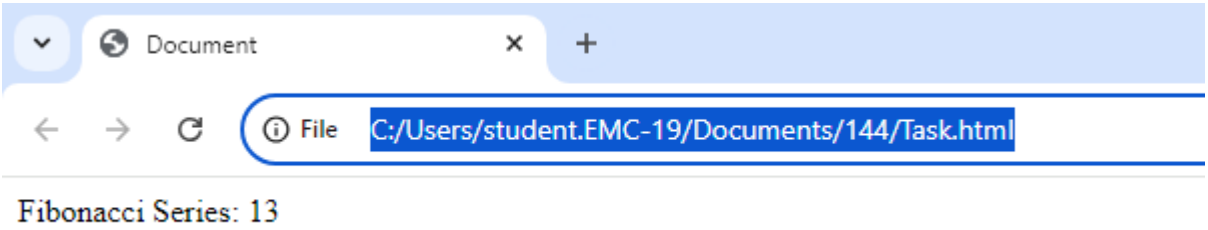
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let n = parseInt(prompt("Enter a number: "));
   function fact(n) {
      if (n === 0 || n === 1) {
         return 1;
      }
      return n * fact(n - 1);
   }
   document.write("Factorial of "+n+" is "+fact(n));
   </script>
</body>
</html>
```



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>
      let n = 7;
   function fibonacci(n) {
      if (n <= 1) {
         return n;
```

```
    }
    return fibonacci(n-1) + fibonacci(n-2);
  }
  document.write("Fibonacci Series: " + fibonacci(n));
  </script>
</body>
</html>
```

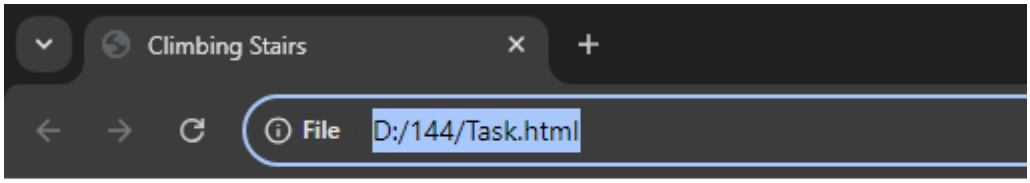


Task 3:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Climbing Stairs</title>
</head>
<body>
  <script>
    let n = 7;

    function StairsToClimb(n) {
      if (n <= 1) {
        return 1;
      }
      if (n === 2) {
        return 2;
      }
      if (n === 3) {
        return 4;
      }
      return StairsToClimb(n - 1) + StairsToClimb(n - 2) + StairsToClimb(n - 3);
    }

    document.write("Total number of ways to climb " + n + " steps: " + StairsToClimb(n));
  </script>
</body>
</html>
```

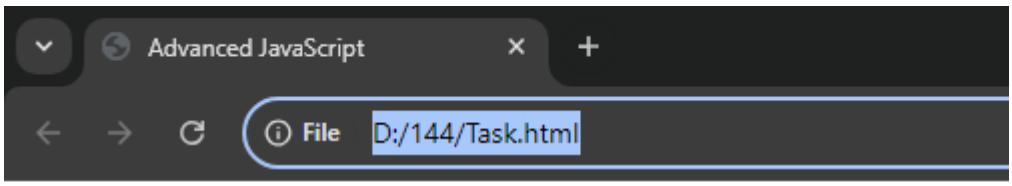


Total number of ways to climb 7 steps: 44

Task 4:

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width">
   <title>Advanced JavaScript</title>
</head>
<body>
   <script>
      function flatten(arr) {
         return arr.reduce((a, b) =>
            a.concat(Array.isArray(b) ? flatten(b) : b),
            []
         );
      }

      const nestedArray = [3, [2, 1], [6, [5, 7]]];
      document.writeln("The flattened array is: ");
      document.writeln(flatten(nestedArray));
   </script>
</body>
</html>
```



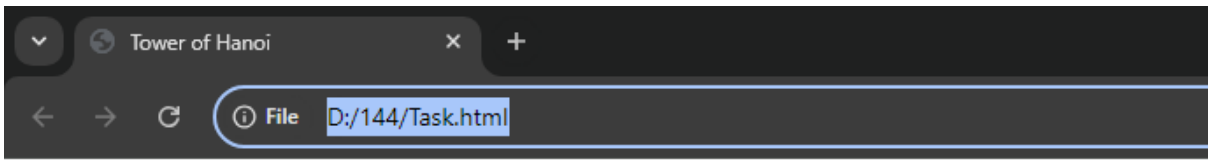
The flattened array is: 3,2,1,6,5,7

Task 5:

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

```
</head>
<body>
   <script>
      function towerOfHanoi(n, from, to, aux) {
         if (n === 1) {
            document.writeln("Move disk 1 from "+from+" to "+to+"<br>");
            return;
         }
         towerOfHanoi(n - 1, from, aux, to);
         document.writeln("Move disk "+n+ " from "+from+" to "+to+"<br>");
         towerOfHanoi(n - 1, aux, to, from);
      }

      const numberOfDisks = 3;
      towerOfHanoi(numberOfDisks, 'A', 'C', 'B');
   </script>
</body>
</html>
```



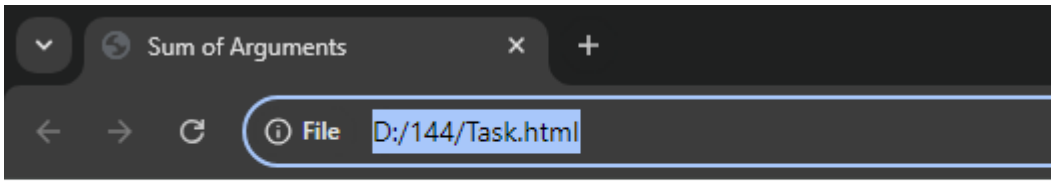
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C

2. JSON and variable length arguments/spread syntax:

Task 1:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Sum of Arguments</title>
</head>
<body>
   <script>
      function Arg() {
         let total = 0;
         for (let i = 0; i < arguments.length; i++) {
            total += arguments[i];
         }
         return total;
```

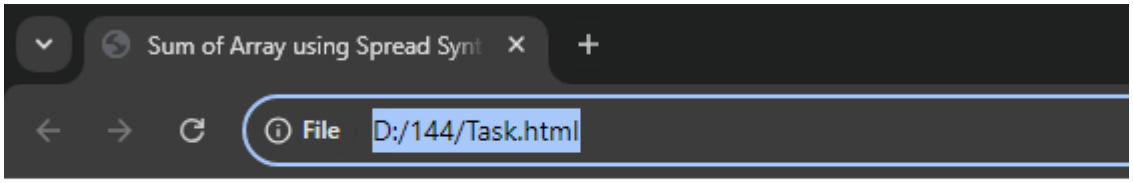
```
        }
        document.writeln("Sum of arguments: " + Arg(1, 2, 3, 4, 5));
    </script>
</body>
</html>
```



Sum of arguments: 15

Task 2:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sum of Array using Spread Syntax</title>
</head>
<body>
    <script>
        function sumOfNumbers(...numbers) {
            return numbers.reduce((total, num) => total + num, 0);
        }
        let numberArray = [5 , 8 , 3 , 1 , 3];
        document.writeln("Sum of numbers: " + sumOfNumbers(...numberArray));
    </script>
</body>
</html>
```



Sum of numbers: 20

Task 3:

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

```
</head>
<body>
   <script>
      function deepClone(obj) {
         return JSON.parse(JSON.stringify(obj));
      }

      const originalObject = {
         name: "John",
         age: 30,
         address: {
            street: "123 Main St",
            city: "Anytown"
         }
      };

      const clonedObject = deepClone(originalObject);
      clonedObject.address.city = "NewCity";
      document.writeln("Original Object: " + JSON.stringify(originalObject) + "<br>");
      document.writeln("Cloned Object: " + JSON.stringify(clonedObject) + "<br>");
   </script>
</body>
</html>
```

Original Object: {"name":"John","age":30,"address":{"street":"123 Main St","city":"Anytown"}}
Cloned Object: {"name":"John","age":30,"address":{"street":"123 Main St","city":"NewCity"}}

Task 4:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Merge Objects with Spread Syntax</title>
</head>
<body>
   <script>
      function mergeObjects(obj1, obj2) {
         return { ...obj1, ...obj2 };
      }

      const object1 = { name: "John", age: 30 };
      const object2 = { city: "New York", country: "USA" };
```

```
        const mergedObject = mergeObjects(object1, object2);

        document.writeln("Merged Object: " + JSON.stringify(mergedObject));
    </script>
</body>
</html>
```

Merged Object: {"name":"John","age":30,"city":"New York","country":"USA"}

Task 5:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Serialize and Parse Object</title>
</head>
<body>
    <script>
        const person = {
            name: "John",
            age: 30,
            city: "New York"
        };
        const jsonString = JSON.stringify(person);
        document.writeln("Serialized JSON string: " + jsonString + "<br>");
        const parsedObject = JSON.parse(jsonString);
        document.writeln("Parsed object: " + JSON.stringify(parsedObject));
    </script>
</body>
</html>
```

Serialized JSON string: {"name":"John","age":30,"city":"New York"}
Parsed object: {"name":"John","age":30,"city":"New York"}