



MOBILE APPLICATION DEVELOPMENT ASSIGNMENT 02

NAME:

MUHAMMAD MUAAB SHOAIB

REG. NO:

FA20-BCS-074

CLASS:

BCS-6B

SUBMITTED TO:

MR. MUHAMMAD KAMRAN

DATE:

19-MAR-2023

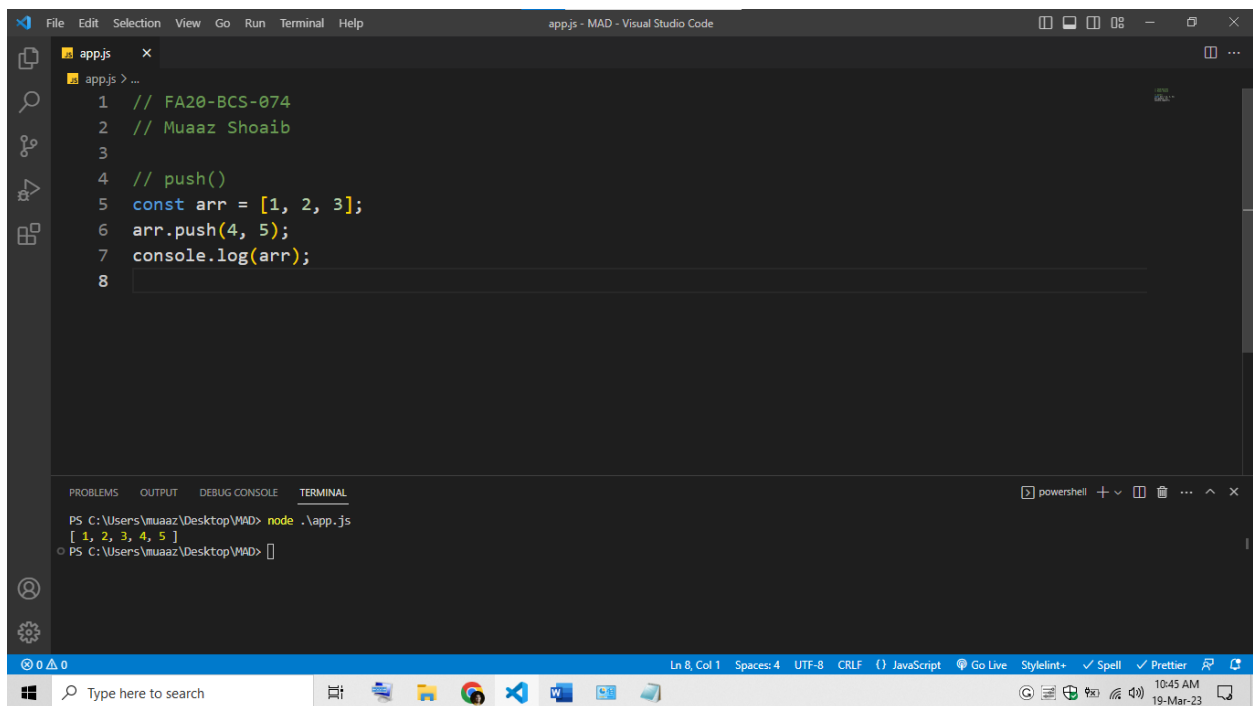
Q1: Array functions in JavaScript with examples.

push

This method adds one or more elements to the end of an array and returns the new length of the array.

```
// FA20-BCS-074
// Muaaz Shoaib

// push()
const arr = [1, 2, 3];
arr.push(4, 5);
console.log(arr);
```

A screenshot of the Visual Studio Code editor interface. The main editor window shows a file named 'app.js' with the following code:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // push()
5 const arr = [1, 2, 3];
6 arr.push(4, 5);
7 console.log(arr);
8
```

 The bottom panel shows the 'TERMINAL' tab with the command 'node .\app.js' executed, resulting in the output '[1, 2, 3, 4, 5]'. The status bar at the bottom indicates 'Ln 8, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and 'JavaScript'.

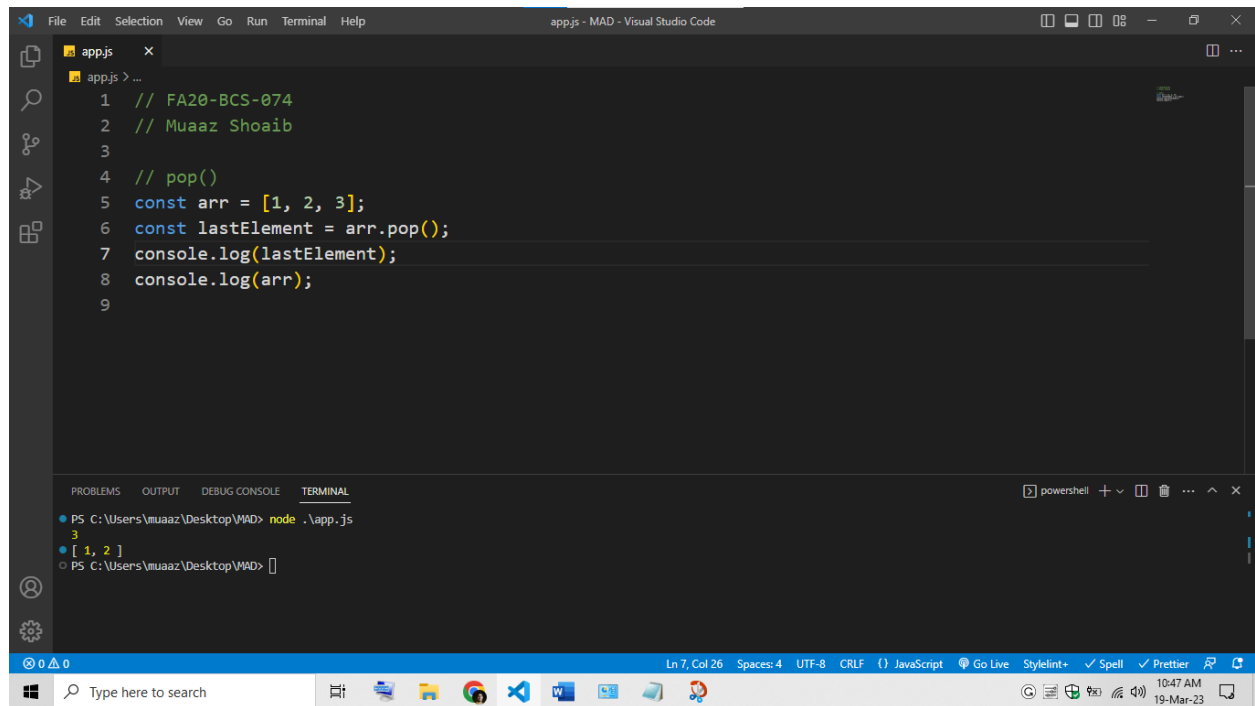
pop

This method removes the last element from an array and returns that element.

```
// FA20-BCS-074
// Muaaz Shoaib

// pop()
```

```
const arr = [1, 2, 3];
const lastElement = arr.pop();
console.log(lastElement);
console.log(arr);
```



The screenshot shows the Visual Studio Code editor with a file named `app.js` open. The code in the editor is as follows:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // pop()
5 const arr = [1, 2, 3];
6 const lastElement = arr.pop();
7 console.log(lastElement);
8 console.log(arr);
9
```

The terminal at the bottom shows the command `node .\app.js` being executed, with the output:

```
3
[ 1, 2 ]
```

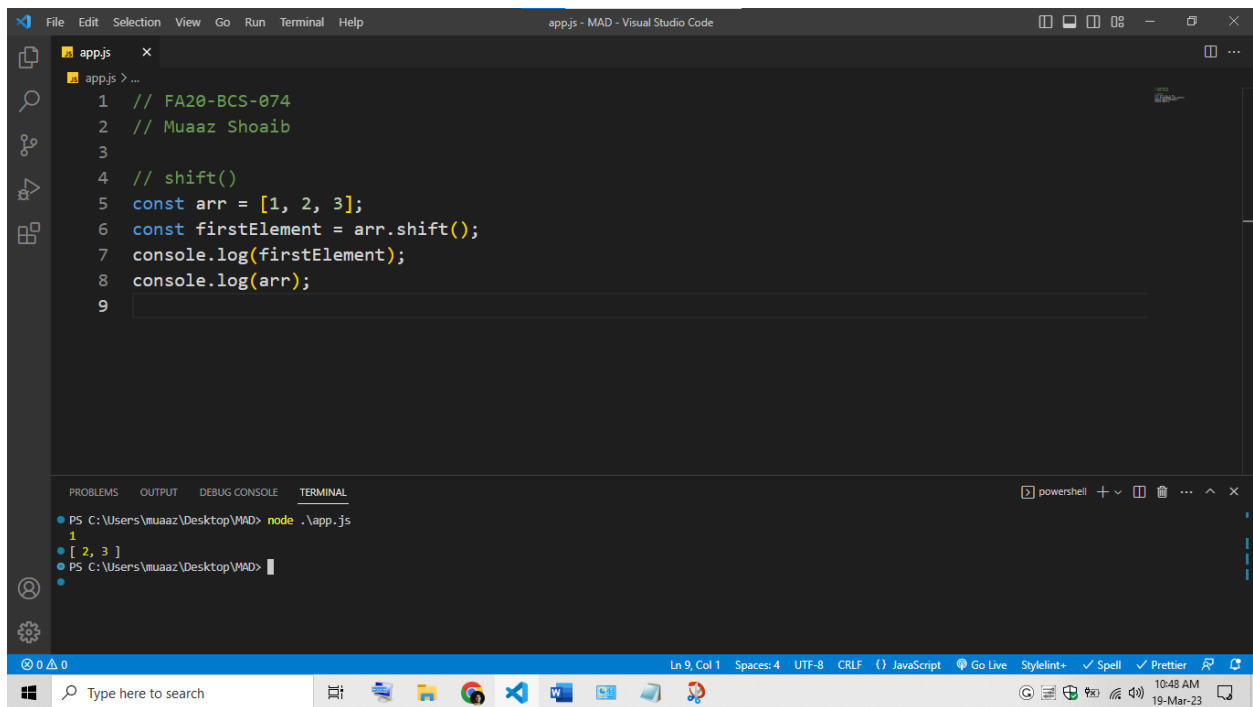
The status bar at the bottom indicates the current line is 7, column 26, with 4 spaces, UTF-8 encoding, and CRLF line endings. The file is identified as JavaScript.

shift

This method removes the first element from an array and returns that element.

```
// FA20-BCS-074
// Muaaz Shoaib

// shift()
const arr = [1, 2, 3];
const firstElement = arr.shift();
console.log(firstElement);
console.log(arr);
```



The screenshot shows the Visual Studio Code editor with a file named `app.js` open. The code in the editor is as follows:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // shift()
5 const arr = [1, 2, 3];
6 const firstElement = arr.shift();
7 console.log(firstElement);
8 console.log(arr);
9
```

The terminal at the bottom shows the command `node .\app.js` being executed, with the following output:

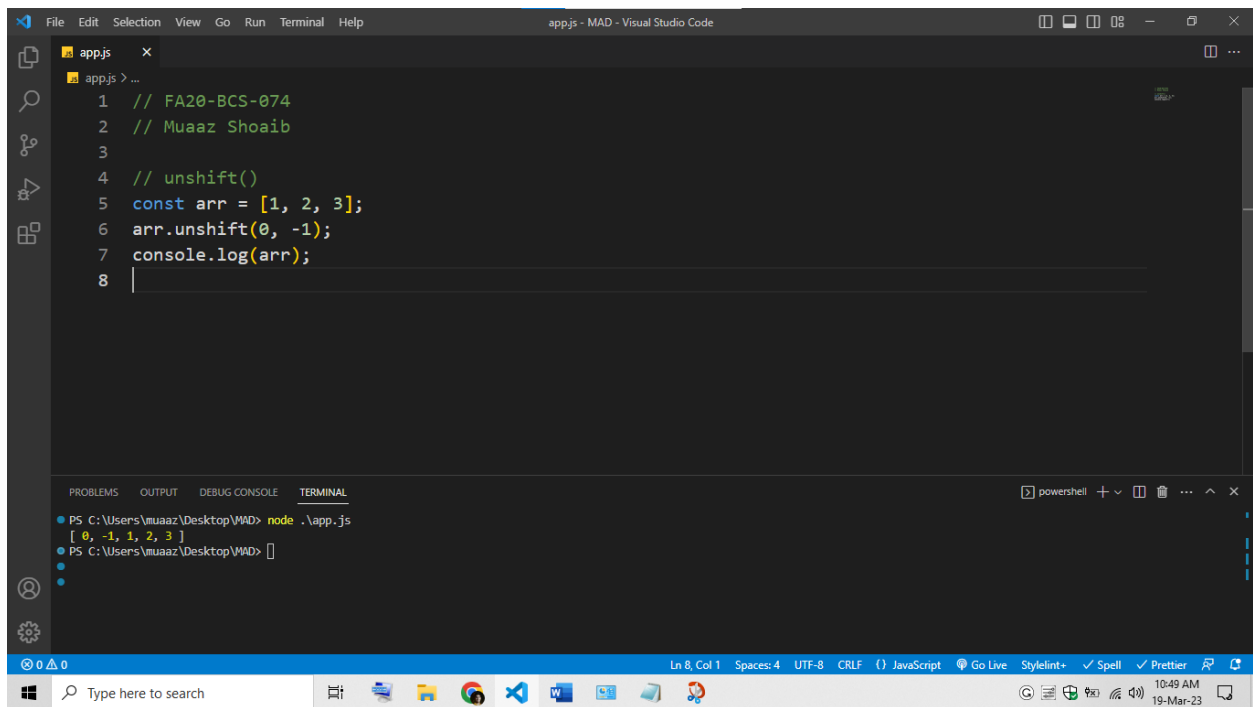
```
1
[ 2, 3 ]
PS C:\Users\muuaz\Desktop\WAD>
```

unshift

This method adds one or more elements to the beginning of an array and returns the new length of the array.

```
// FA20-BCS-074
// Muaaz Shoaib

// unshift()
const arr = [1, 2, 3];
arr.unshift(0, -1);
console.log(arr);
```



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following code:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // unshift()
5 const arr = [1, 2, 3];
6 arr.unshift(0, -1);
7 console.log(arr);
8
```

The terminal at the bottom shows the command `node .\app.js` being executed, resulting in the output:

```
[ 0, -1, 1, 2, 3 ]
```

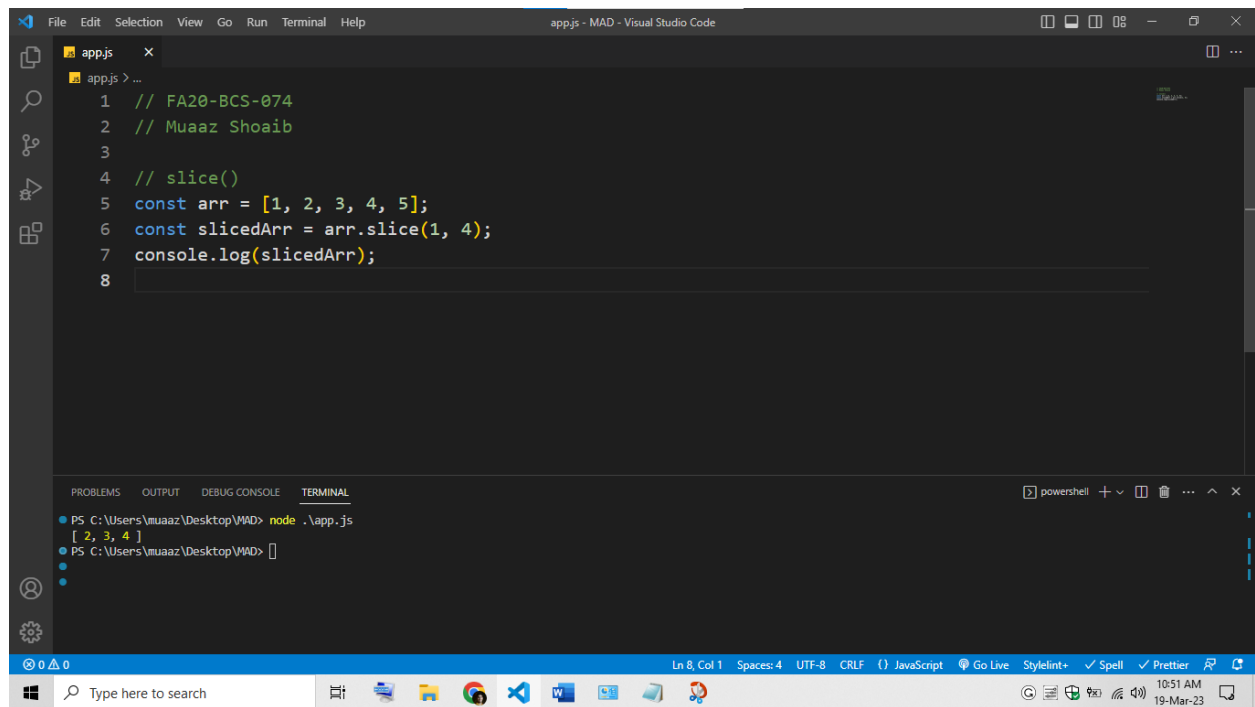
The status bar at the bottom indicates the file is at line 8, column 1, using UTF-8 encoding and CRLF line endings. It also shows various extensions like JavaScript, Go Live, Stylelint, Spell, and Prettier are installed.

slice

This method returns a shallow copy of a portion of an array into a new array.

```
// FA20-BCS-074
// Muaaz Shoaib

// slice()
const arr = [1, 2, 3, 4, 5];
const slicedArr = arr.slice(1, 4);
console.log(slicedArr);
```



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following code:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // slice()
5 const arr = [1, 2, 3, 4, 5];
6 const slicedArr = arr.slice(1, 4);
7 console.log(slicedArr);
8
```

The bottom panel shows the `TERMINAL` tab with the following output:

```
PS C:\Users\muuaz\Desktop\WAD> node .\app.js
[ 2, 3, 4 ]
PS C:\Users\muuaz\Desktop\WAD>
```

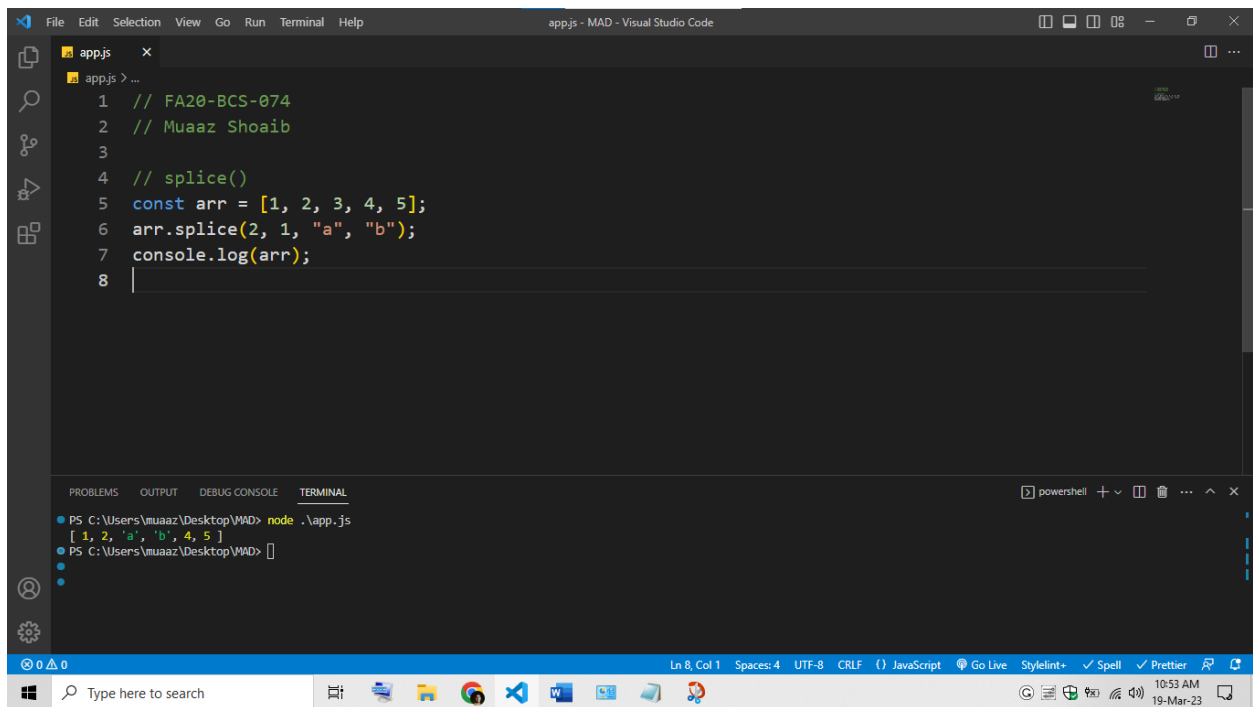
The status bar at the bottom indicates the current line and column as `Ln 8, Col 1`, and the file encoding as `UTF-8`. The Windows taskbar is visible at the bottom of the screen.

splice

This method changes the contents of an array by removing or replacing existing elements and/or adding new elements.

```
// FA20-BCS-074
// Muaaz Shoaib

// splice()
const arr = [1, 2, 3, 4, 5];
arr.splice(2, 1, "a", "b");
console.log(arr);
```



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following code:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // splice()
5 const arr = [1, 2, 3, 4, 5];
6 arr.splice(2, 1, "a", "b");
7 console.log(arr);
8
```

The terminal at the bottom shows the command `node .\app.js` being executed, resulting in the output:

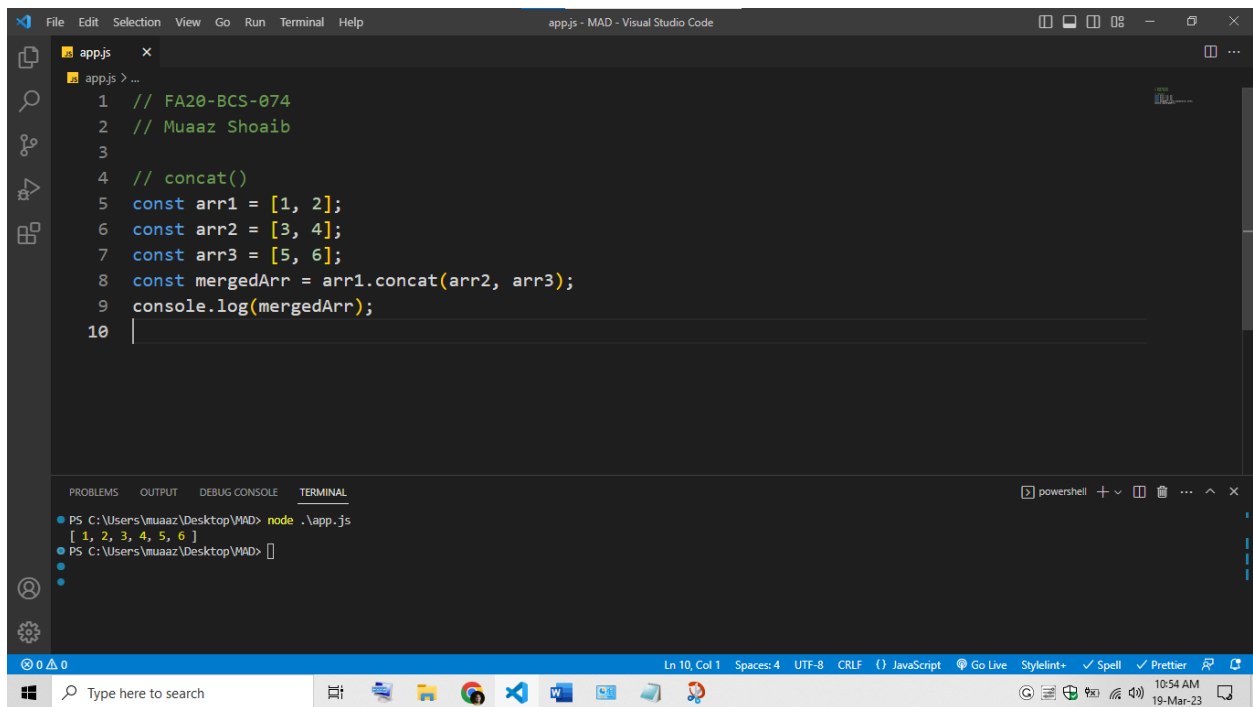
```
[ 1, 2, 'a', 'b', 4, 5 ]
```

concat

This method merges two or more arrays into a new array.

```
// FA20-BCS-074
// Muaaz Shoaib

// concat()
const arr1 = [1, 2];
const arr2 = [3, 4];
const arr3 = [5, 6];
const mergedArr = arr1.concat(arr2, arr3);
console.log(mergedArr);
```



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following code:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // concat()
5 const arr1 = [1, 2];
6 const arr2 = [3, 4];
7 const arr3 = [5, 6];
8 const mergedArr = arr1.concat(arr2, arr3);
9 console.log(mergedArr);
10
```

The terminal at the bottom shows the command `node .\app.js` being executed, resulting in the output:

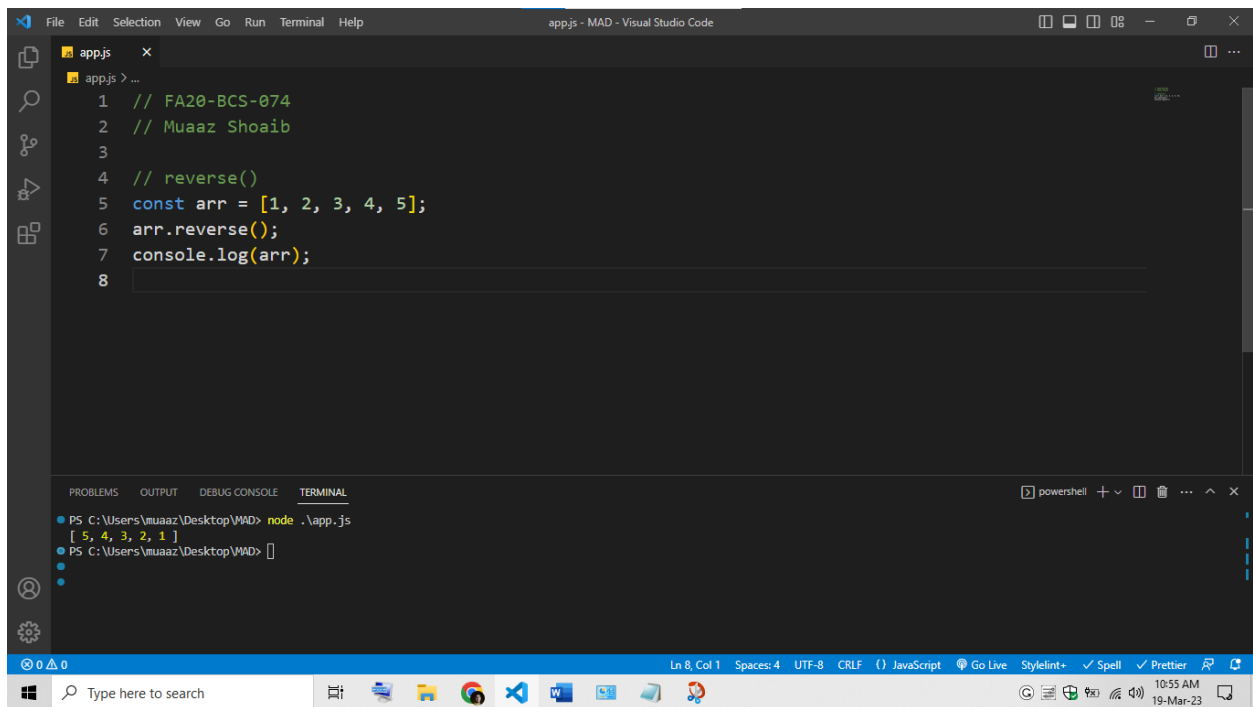
```
PS C:\Users\muuaz\Desktop\WAD> node .\app.js
[ 1, 2, 3, 4, 5, 6 ]
PS C:\Users\muuaz\Desktop\WAD>
```

reverse

This method reverses the order of the elements in an array.

```
// FA20-BCS-074
// Muaaz Shoaib

// reverse()
const arr = [1, 2, 3, 4, 5];
arr.reverse();
console.log(arr);
```

The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following code:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // reverse()
5 const arr = [1, 2, 3, 4, 5];
6 arr.reverse();
7 console.log(arr);
8
```

The bottom panel shows the `TERMINAL` tab with the following output:

```
PS C:\Users\muuaz\Desktop\WAD> node .\app.js
[ 5, 4, 3, 2, 1 ]
PS C:\Users\muuaz\Desktop\WAD>
```

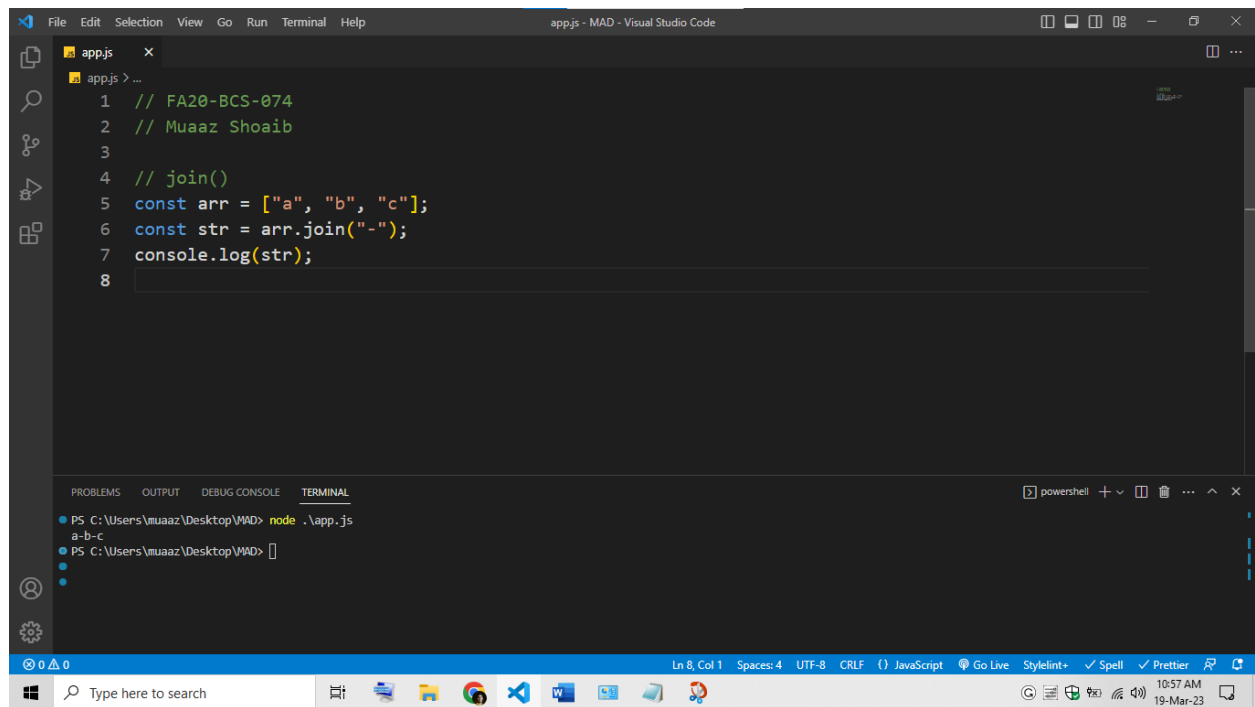
The status bar at the bottom indicates the current line and column as `Ln 8, Col 1`, and the file encoding as `UTF-8`.

join

This method joins all elements of an array into a string.

```
// FA20-BCS-074
// Muaaz Shoaib

// join()
const arr = ["a", "b", "c"];
const str = arr.join("-");
console.log(str);
```



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following code:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // join()
5 const arr = ["a", "b", "c"];
6 const str = arr.join("-");
7 console.log(str);
8
```

The bottom panel shows the `TERMINAL` tab with the following output:

```
PS C:\Users\muuaz\Desktop\WAD> node .\app.js
a-b-c
PS C:\Users\muuaz\Desktop\WAD>
```

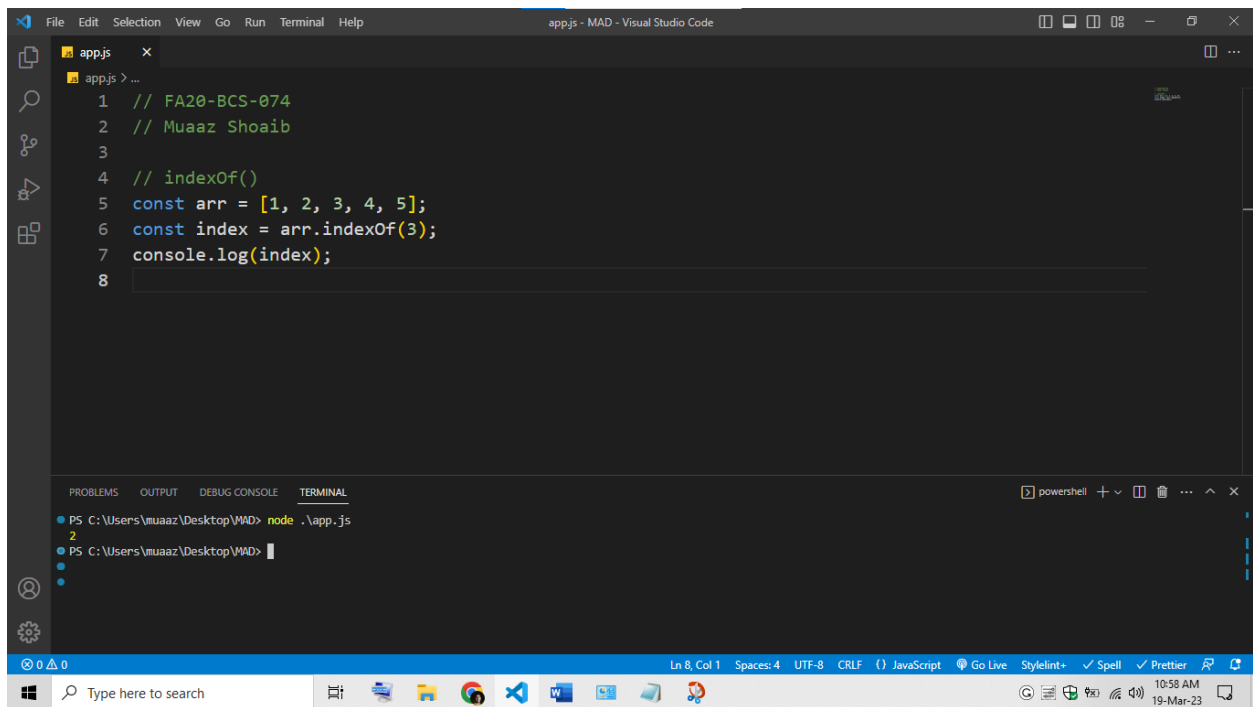
The status bar at the bottom indicates the current line and column as `Ln 8, Col 1`, and the file encoding as `UTF-8`.

indexOf

This method returns the first index at which a given element can be found in an array, or -1 if it is not present.

```
// FA20-BCS-074
// Muaaz Shoaib

// indexOf()
const arr = [1, 2, 3, 4, 5];
const index = arr.indexOf(3);
console.log(index);
```



The screenshot shows the Visual Studio Code editor interface. The main editor window displays a file named `app.js` with the following code:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // indexOf()
5 const arr = [1, 2, 3, 4, 5];
6 const index = arr.indexOf(3);
7 console.log(index);
8
```

The bottom panel shows the TERMINAL tab with the following output:

```
PS C:\Users\muaz\Desktop\WAD> node .\app.js
2
PS C:\Users\muaz\Desktop\WAD>
```

The status bar at the bottom indicates the current line and column (Ln 8, Col 1), the encoding (UTF-8), and the file type (JavaScript). The Windows taskbar is visible at the bottom of the screen.

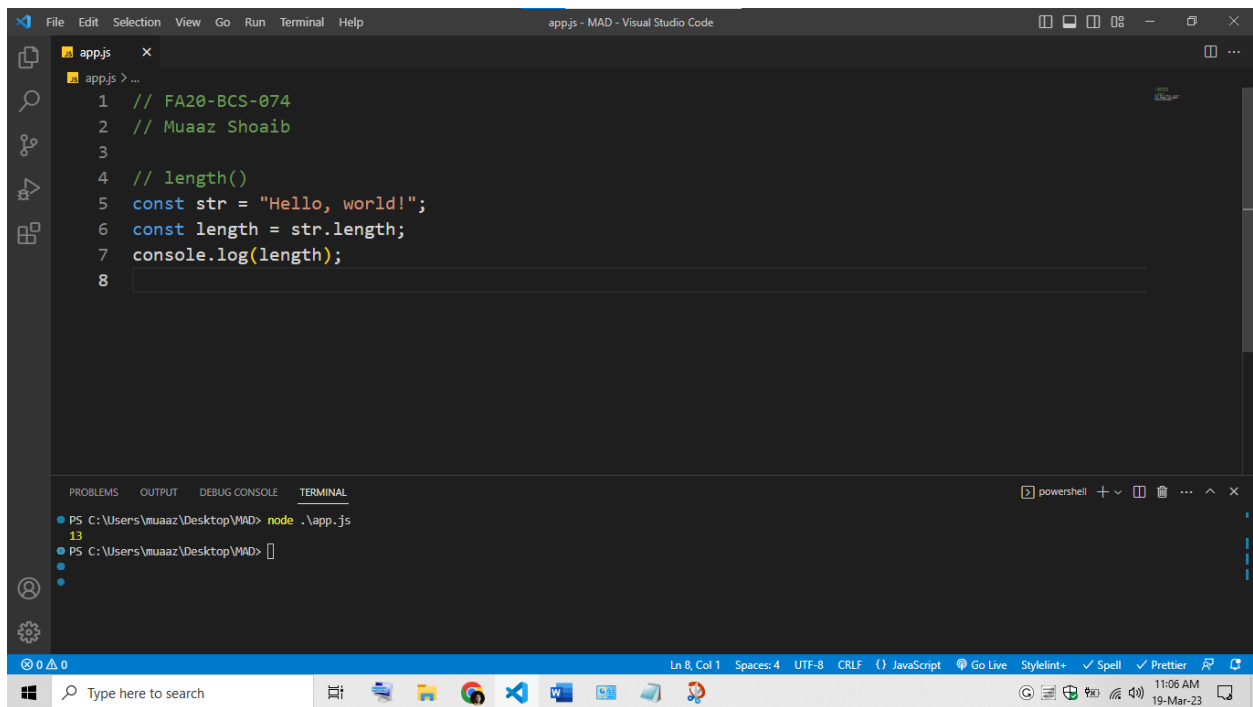
Q2: String functions in JavaScript with example.

length

This property returns the length of a string.

```
// FA20-BCS-074
// Muaaz Shoaib

// length()
const str = "Hello, world!";
const length = str.length;
console.log(length);
```



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following content:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // length()
5 const str = "Hello, world!";
6 const length = str.length;
7 console.log(length);
8
```

Below the editor, the TERMINAL panel is active, showing a PowerShell session:

```
PS C:\Users\muuaz\Desktop\WAD> node .\app.js
13
PS C:\Users\muuaz\Desktop\WAD>
```

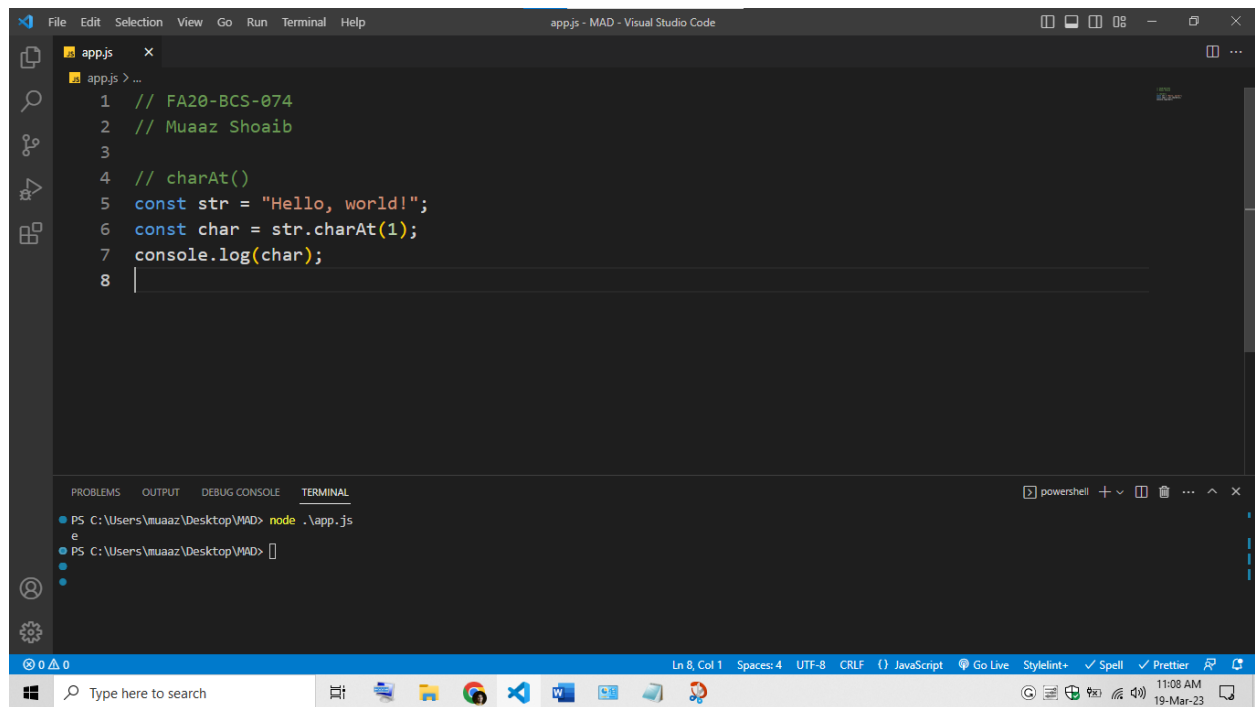
The status bar at the bottom indicates the current position is Line 8, Column 1, with 4 spaces, UTF-8 encoding, and CRLF line endings. It also shows various extensions like Go Live, Stylelint, Spell, and Prettier are installed.

charAt

This method returns the character at a specified index in a string.

```
// FA20-BCS-074
// Muaaz Shoaib

// charAt()
const str = "Hello, world!";
const char = str.charAt(1);
console.log(char);
```

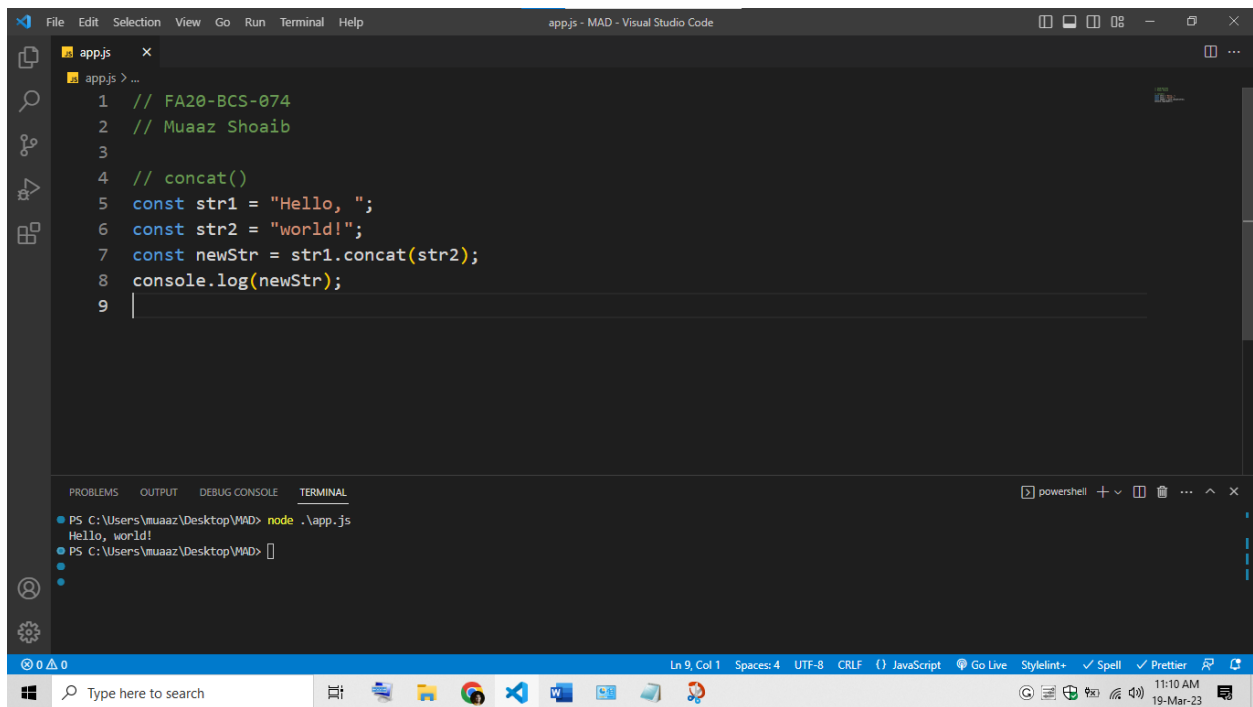


concat

This method concatenates two or more strings and returns the new string.

```
// FA20-BCS-074
// Muaaz Shoaib

// concat()
const str1 = "Hello, ";
const str2 = "world!";
const newStr = str1.concat(str2);
console.log(newStr);
```



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following content:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // concat()
5 const str1 = "Hello, ";
6 const str2 = "world!";
7 const newStr = str1.concat(str2);
8 console.log(newStr);
9
```

The bottom panel shows the `TERMINAL` tab with the following output:

```
PS C:\Users\muaz\Desktop\WAD> node .\app.js
Hello, world!
PS C:\Users\muaz\Desktop\WAD>
```

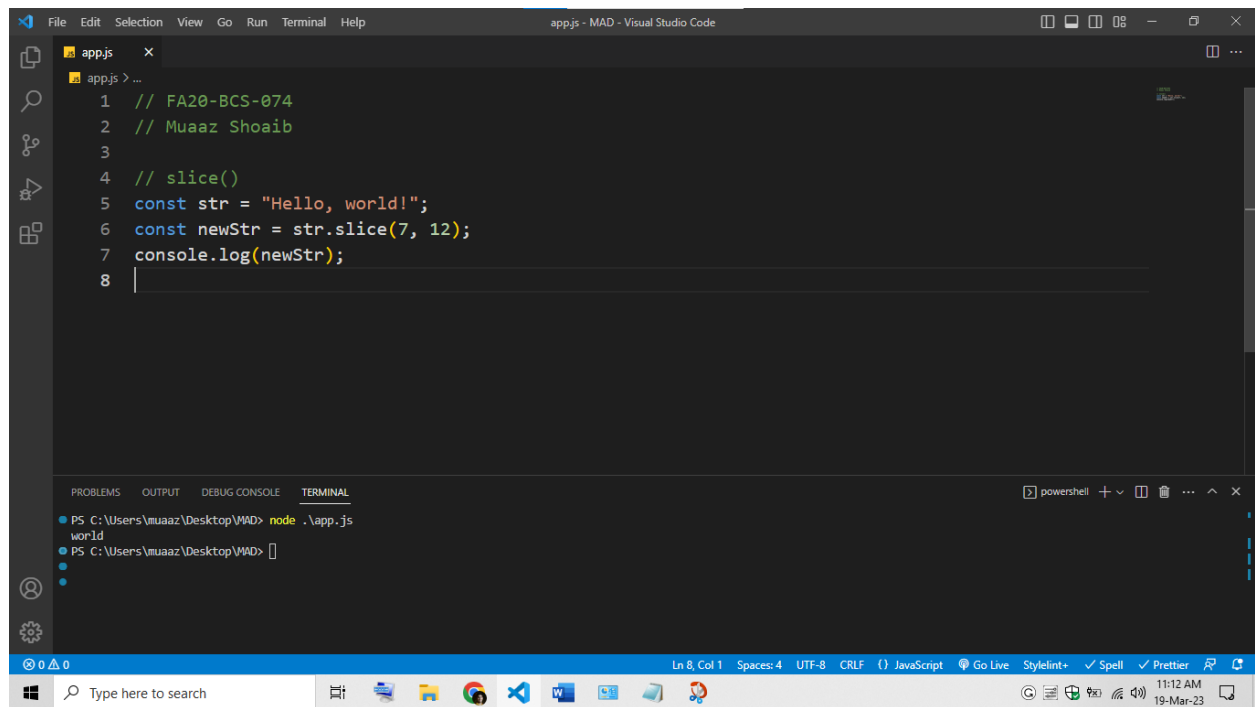
The status bar at the bottom indicates the current position is `Ln 9, Col 1`, the file encoding is `UTF-8`, and the line ending is `CRLF`. The language mode is `JavaScript`. The bottom of the window shows the Windows taskbar with the search bar and various application icons.

slice

This method returns a portion of a string into a new string.

```
// FA20-BCS-074
// Muaaz Shoaib

// slice()
const str = "Hello, world!";
const newStr = str.slice(7, 12);
console.log(newStr);
```



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following content:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // slice()
5 const str = "Hello, world!";
6 const newStr = str.slice(7, 12);
7 console.log(newStr);
8
```

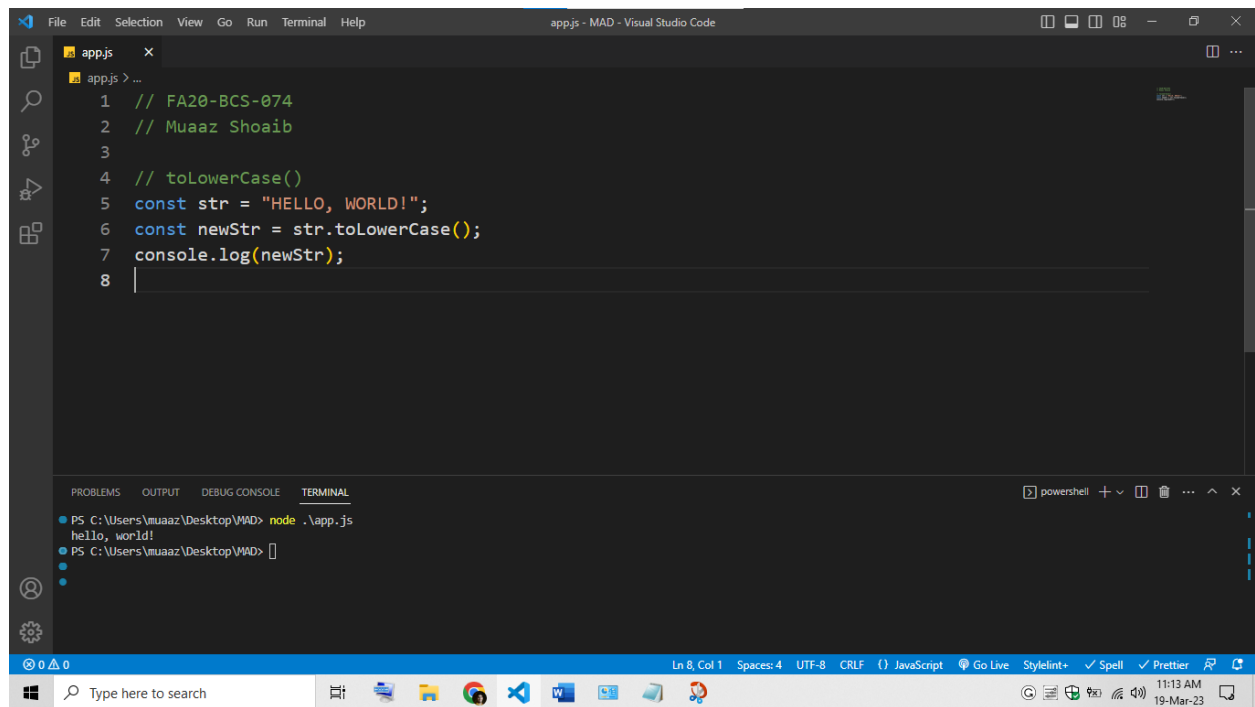
The terminal at the bottom shows the command `node .\app.js` being executed, resulting in the output `world`.

toLowerCase

This method returns a new string with all characters in lowercase.

```
// FA20-BCS-074
// Muaaz Shoaib

// toLowerCase()
const str = "HELLO, WORLD!";
const newStr = str.toLowerCase();
console.log(newStr);
```



```
File Edit Selection View Go Run Terminal Help
app.js - MAD - Visual Studio Code

app.js
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // toLowerCase()
5 const str = "HELLO, WORLD!";
6 const newStr = str.toLowerCase();
7 console.log(newStr);
8

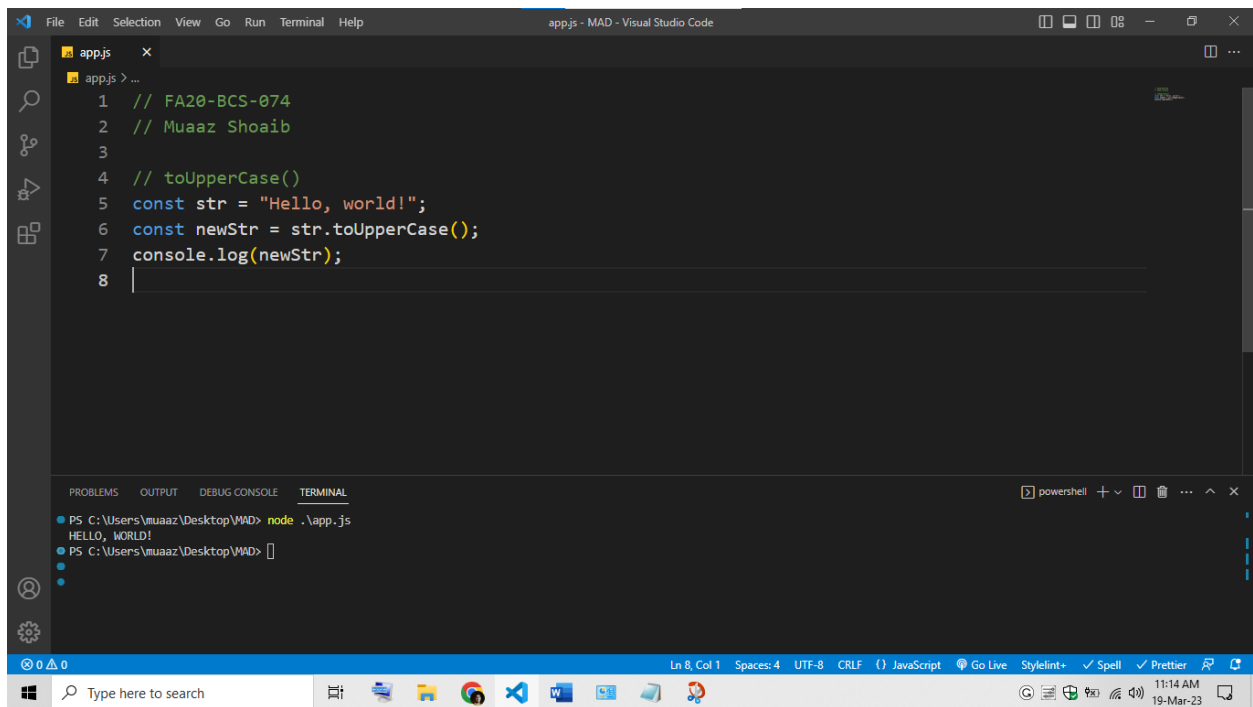
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\muuaz\Desktop\WAD> node .\app.js
hello, world!
PS C:\Users\muuaz\Desktop\WAD>
```

toUpperCase

This method returns a new string with all characters in uppercase.

```
// FA20-BCS-074
// Muaaz Shoaib

// toUpperCase()
const str = "Hello, world!";
const newStr = str.toUpperCase();
console.log(newStr);
```

The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following content:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // toUpperCase()
5 const str = "Hello, world!";
6 const newStr = str.toUpperCase();
7 console.log(newStr);
8
```

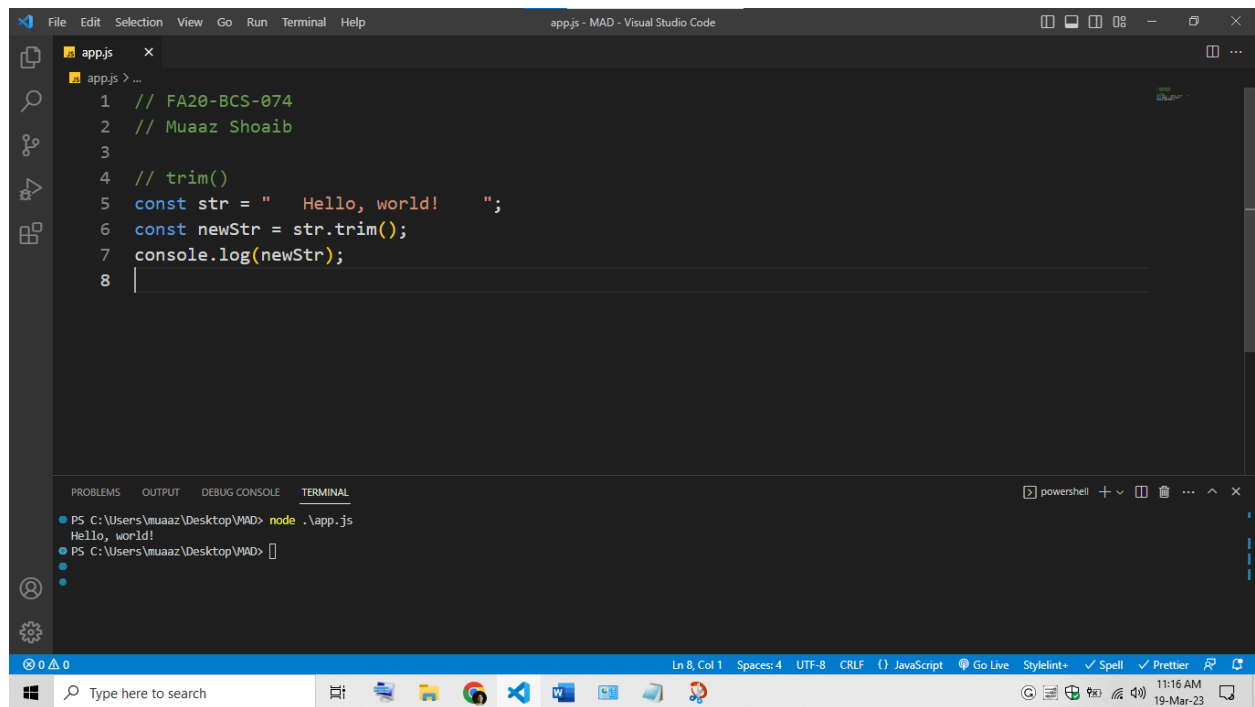
The terminal at the bottom shows the command `node .\app.js` being executed, resulting in the output `HELLO, WORLD!`.

trim

This method removes whitespace from both ends of a string.

```
// FA20-BCS-074
// Muaaz Shoaib

// trim()
const str = "  Hello, world!  ";
const newStr = str.trim();
console.log(newStr);
```



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following content:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // trim()
5 const str = "  Hello, world!  ";
6 const newStr = str.trim();
7 console.log(newStr);
8
```

The bottom panel shows the `TERMINAL` tab with the following output:

```
PS C:\Users\muaz\Desktop\WAD> node .\app.js
Hello, world!
PS C:\Users\muaz\Desktop\WAD>
```

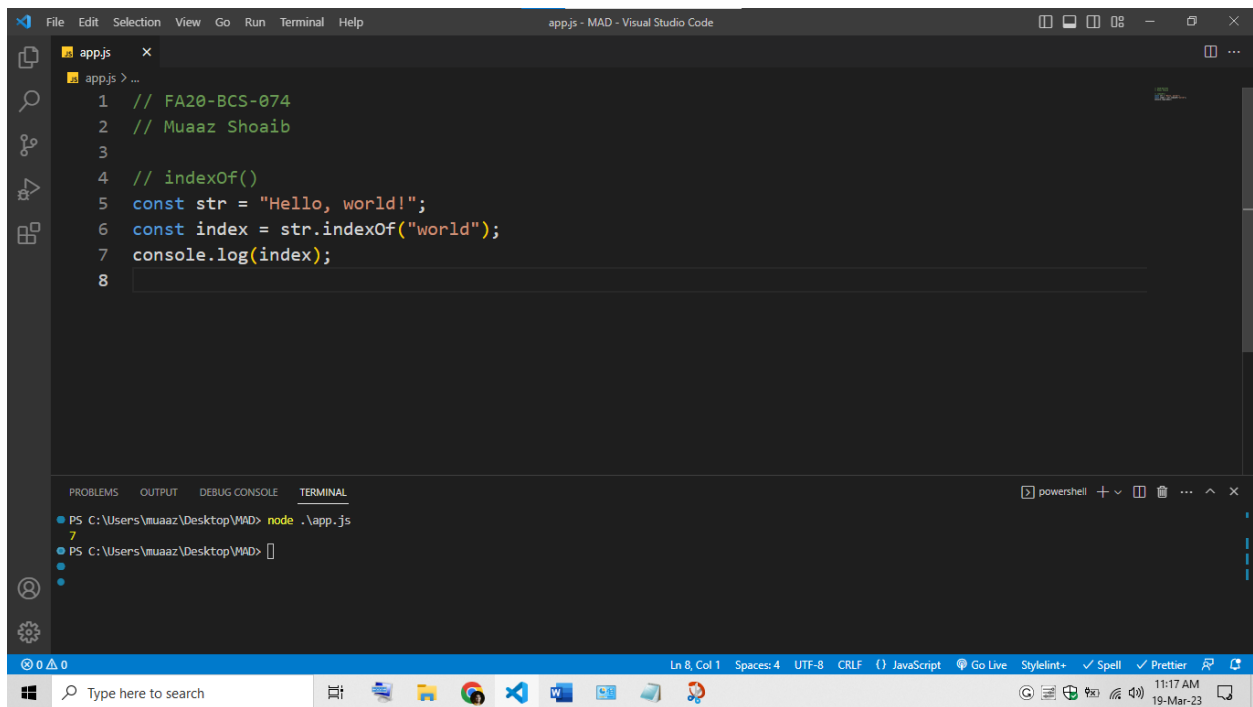
The status bar at the bottom indicates the current line and column as `Ln 8, Col 1`, and the file encoding as `UTF-8`.

indexOf

This method returns the index of the first occurrence of a specified value in a string, or -1 if it is not found.

```
// FA20-BCS-074
// Muaaz Shoaib

// indexOf()
const str = "Hello, world!";
const index = str.indexOf("world");
console.log(index);
```

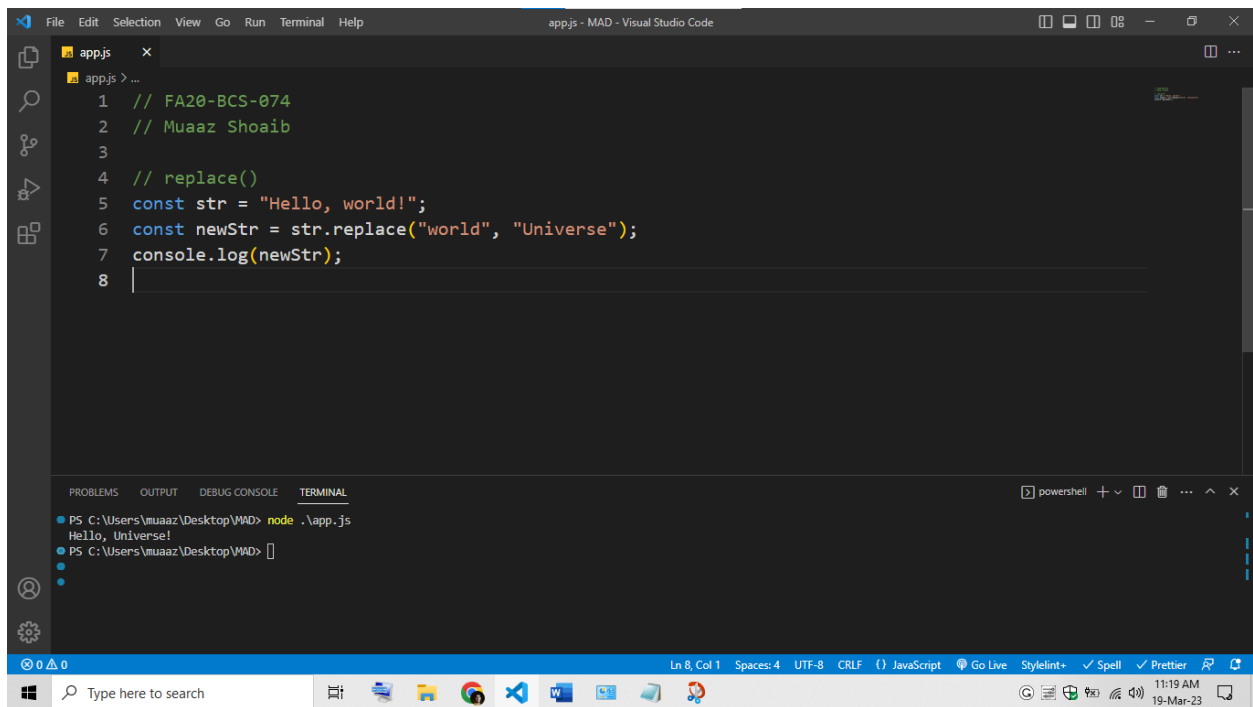


replace

This method replaces a specified value with another value in a string.

```
// FA20-BCS-074
// Muaaz Shoaib

// replace()
const str = "Hello, world!";
const newStr = str.replace("world", "Universe");
console.log(newStr);
```



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `app.js` with the following code:

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // replace()
5 const str = "Hello, world!";
6 const newStr = str.replace("world", "Universe");
7 console.log(newStr);
8
```

The terminal at the bottom shows the command `node .\app.js` being executed, resulting in the output `Hello, Universe!`.

split

This method splits a string into an array of substrings based on a specified separator.

```
// FA20-BCS-074
// Muaaz Shoaib

// split()
const str = "Hello, world!";
const arr = str.split(",");
console.log(arr);
```

```
1 // FA20-BCS-074
2 // Muaaz Shoaib
3
4 // split()
5 const str = "Hello, world!";
6 const arr = str.split(",");
7 console.log(arr);
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\muuaz\Desktop\WAD> node .\app.js
['Hello', ' world!']
PS C:\Users\muuaz\Desktop\WAD>

Ln 8, Col 1 Spaces: 4 UTF-8 CRLF () JavaScript Go Live Stylelint+ Spell Prettier

Q3: Design chess board UI in react native?

You will find a complete project by clicking on this link: https://github.com/muaazshoaib/chess_board

```
1 // Muaaz Shoaib
2 // FA20-BCS-074
3 // Design a chess board
4
5 import { StyleSheet, View } from "react-native";
6
7 const BOARD_SIZE = 8;
8
9 export default function App() {
10   const renderSquare = (row, col) => {
11     const isDark = (row + col) % 2 === 1;
12     const squareColor = isDark ? "green" : "white";
13
14     return <View style={ [styles.square, { backgroundColor: squareColor } ] } />;
15   };
16
17   const renderRow = (row) => {
18     return (
19       <View style={styles.row} />
20     );
21   };
22
23   return (
24     <View style={styles.container} />
25   );
26 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

r - reload the app
d - open developer menu
i - run on iOS
a - run on Android

Info Opening the app on Android...
Info Run CLI with --verbose flag for more details.

main 0 0 0 React Native Packager Spaces: 2 UTF-8 LF () JavaScript Go Live Stylelint+ Spell Prettier

```
// Muaaz Shoaib
// FA20-BCS-074
// Design a chess board

import { StyleSheet, View } from "react-native";

const BOARD_SIZE = 8;

export default function App() {
  const renderSquare = (row, col) => {
    const isDark = (row + col) % 2 === 1;
    const squareColor = isDark ? "green" : "white";

    return <View style={[styles.square, {
backgroundColor: squareColor }]} />;
  };

  const renderRow = (row) => {
    const squares = [];

    for (let col = 0; col < BOARD_SIZE; col++) {
      squares.push(renderSquare(row, col));
    }

    return <View style={styles.row}>{squares}</View>;
  };

  const rows = [];

  for (let row = 0; row < BOARD_SIZE; row++) {
```

```
    rows.push(renderRow(row));
  }

  return <View style={styles.board}>{rows}</View>;
}

const styles = StyleSheet.create({
  board: {
    flex: 1,
    flexDirection: "column",
    justifyContent: "center",
    alignItems: "center",
  },
  row: {
    flexDirection: "row",
    justifyContent: "center",
    alignItems: "center",
  },
  square: {
    width: 42,
    height: 42,
  },
});
```