ASSIGNMENT 02

NAME:

MUHAMMAD AWAIS ASLAM

REG. NO:

FA20-BCS-067

SUBMITTED TO:

MR. MUHAMMAD KAMRAN

DATE:

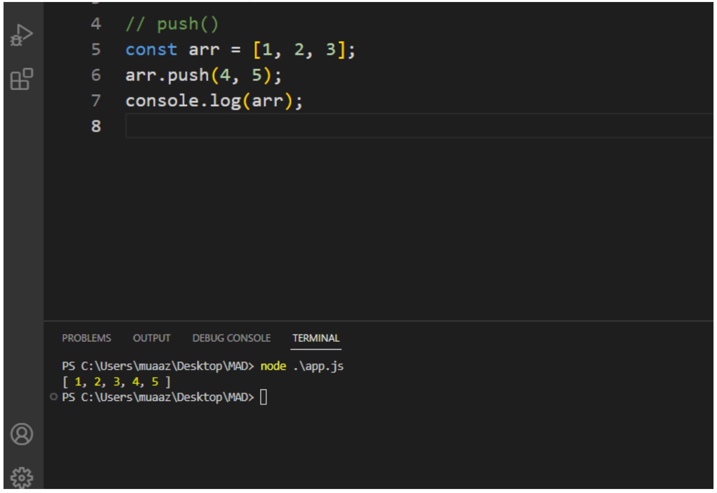
17-04-2023

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.

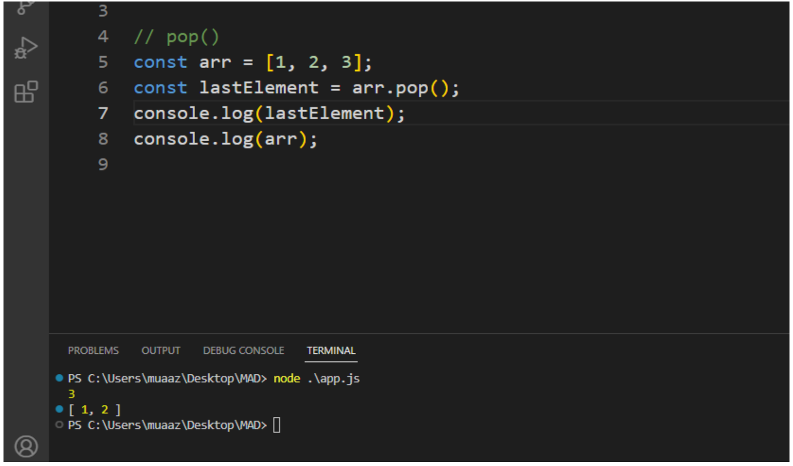
```
const arr = [1, 2, 3];
arr.push(4, 5);
console.log(arr);
```



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

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

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



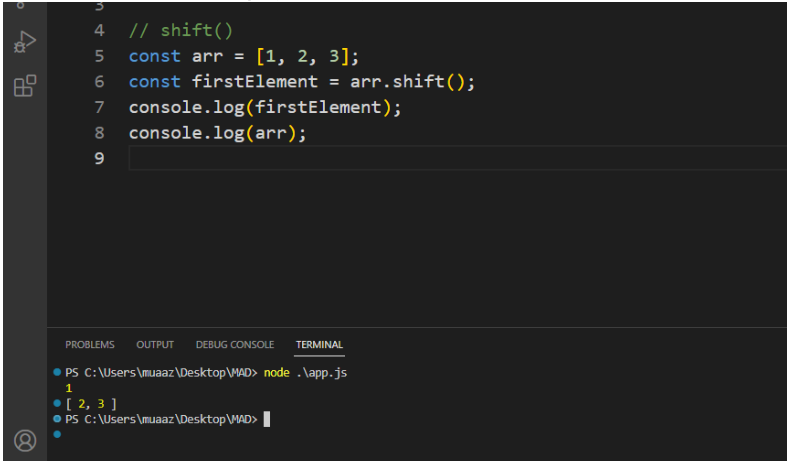
```
// pop()
const arr = [1, 2, 3];
const lastElement = arr.pop();
console.log(lastElement);
console.log(arr);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
3
[ 1, 2 ]
PS C:\Users\muaaz\Desktop\MAD> 
```

shift

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

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



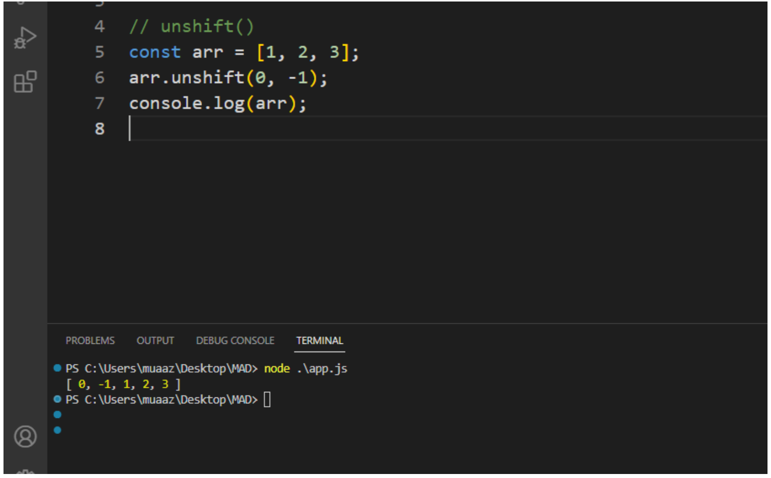
```
// shift()
const arr = [1, 2, 3];
const firstElement = arr.shift();
console.log(firstElement);
console.log(arr);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
1
[ 2, 3 ]
PS C:\Users\muaaz\Desktop\MAD>
```

unshift

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

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



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

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
[ 0, -1, 1, 2, 3 ]
PS C:\Users\muaaz\Desktop\MAD>
```

slice

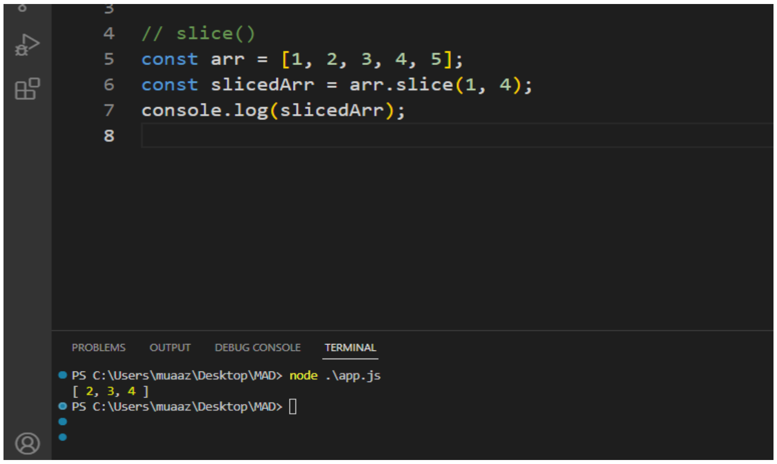
This method returns a shallow copy of a portion of an array into a new array.

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

splice

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

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



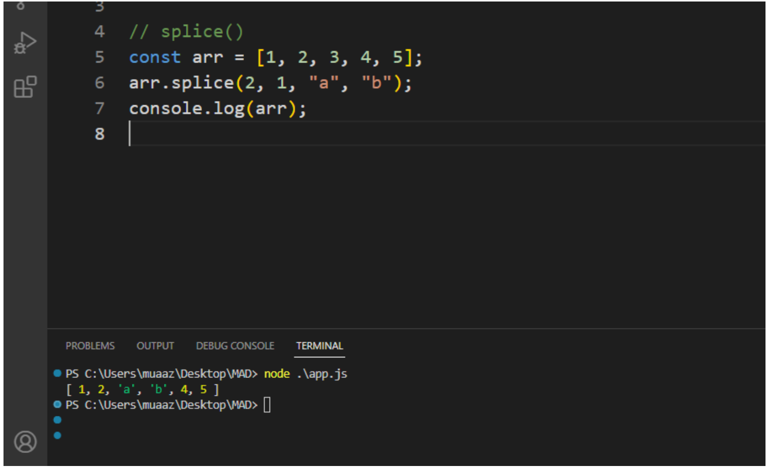
```
// slice()
const arr = [1, 2, 3, 4, 5];
const slicedArr = arr.slice(1, 4);
console.log(slicedArr);
```

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

splice

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

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



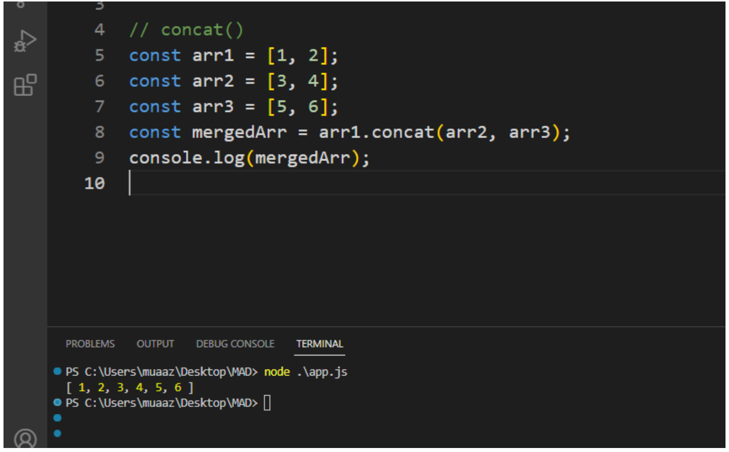
```
// splice()
const arr = [1, 2, 3, 4, 5];
arr.splice(2, 1, "a", "b");
console.log(arr);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
[ 1, 2, 'a', 'b', 4, 5 ]
PS C:\Users\muaaz\Desktop\MAD> 
```

concat

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

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



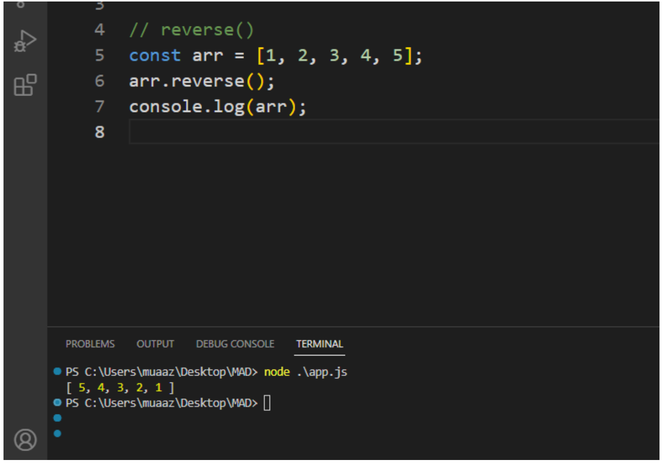
```
// concat()
const arr1 = [1, 2];
const arr2 = [3, 4];
const arr3 = [5, 6];
const mergedArr = arr1.concat(arr2, arr3);
console.log(mergedArr);
```

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

reverse

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

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



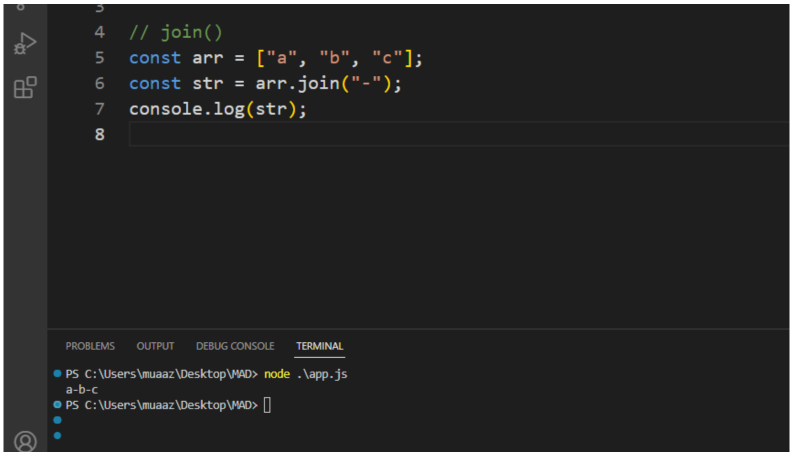
```
// reverse()
const arr = [1, 2, 3, 4, 5];
arr.reverse();
console.log(arr);
```

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

join

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

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



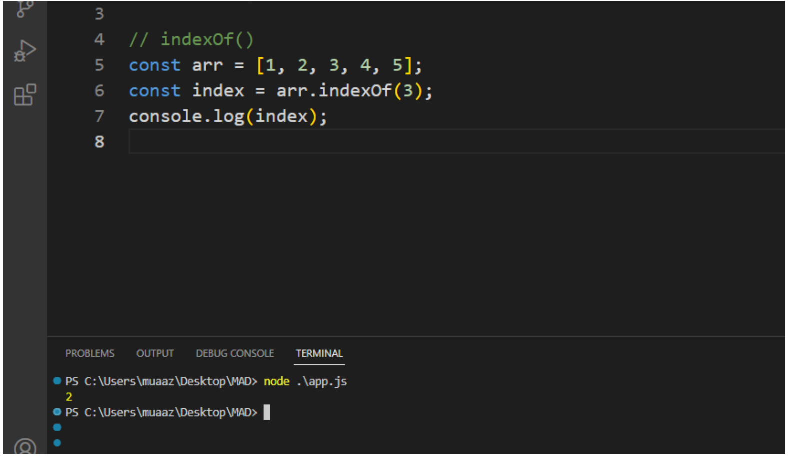
```
// join()
const arr = ["a", "b", "c"];
const str = arr.join("-");
console.log(str);
```

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

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.

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



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

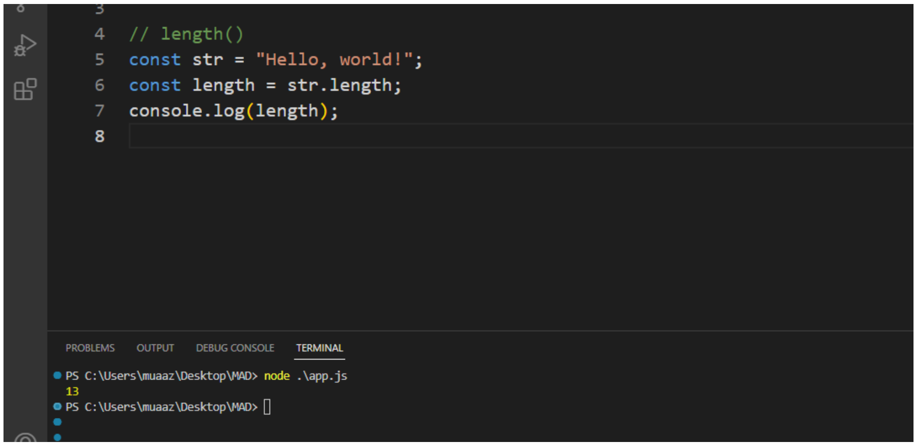
```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
2
PS C:\Users\muaaz\Desktop\MAD>
```

Q2: String functions in JavaScript with example.

length

This property returns the length of a string.

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



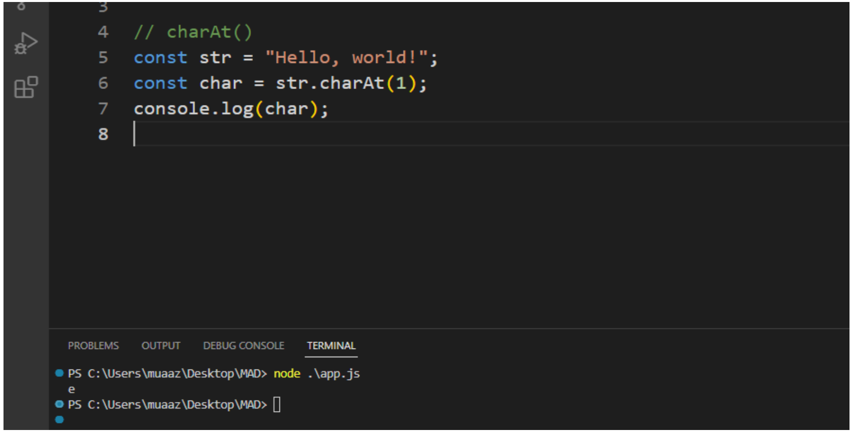
```
// length()
const str = "Hello, world!";
const length = str.length;
console.log(length);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
13
PS C:\Users\muaaz\Desktop\MAD> 
```

charAt

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

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



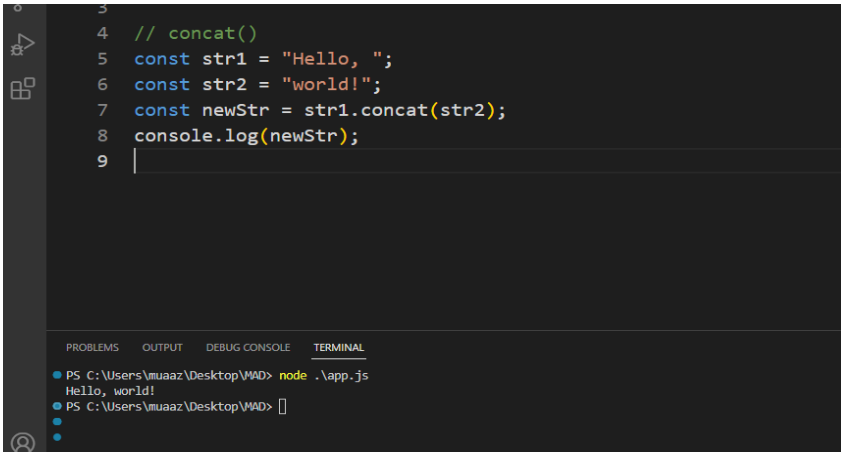
```
// charAt()
const str = "Hello, world!";
const char = str.charAt(1);
console.log(char);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
e
PS C:\Users\muaaz\Desktop\MAD> 
```

concat

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

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



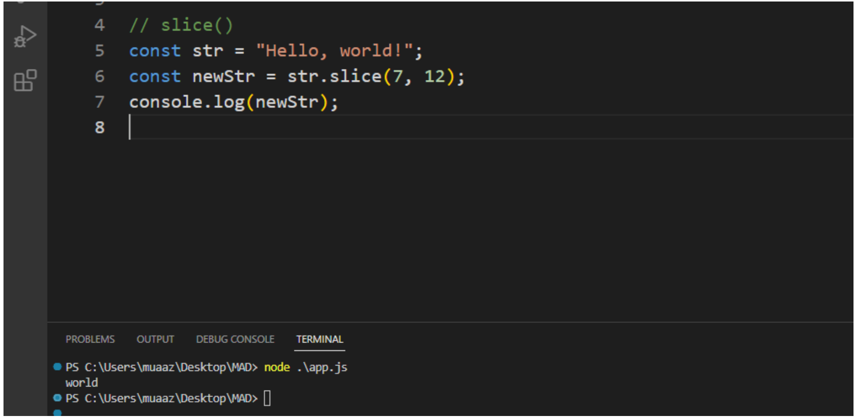
```
// concat()
const str1 = "Hello, ";
const str2 = "world!";
const newStr = str1.concat(str2);
console.log(newStr);
```

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

slice

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

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



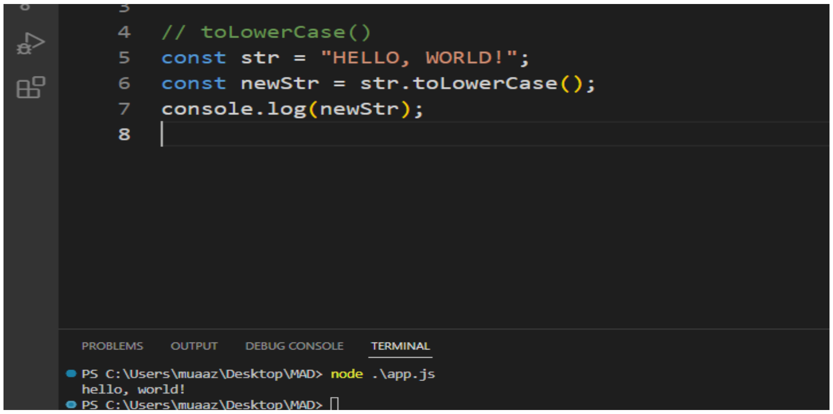
```
// slice()
const str = "Hello, world!";
const newStr = str.slice(7, 12);
console.log(newStr);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
world
PS C:\Users\muaaz\Desktop\MAD> 
```

toLowerCase

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

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



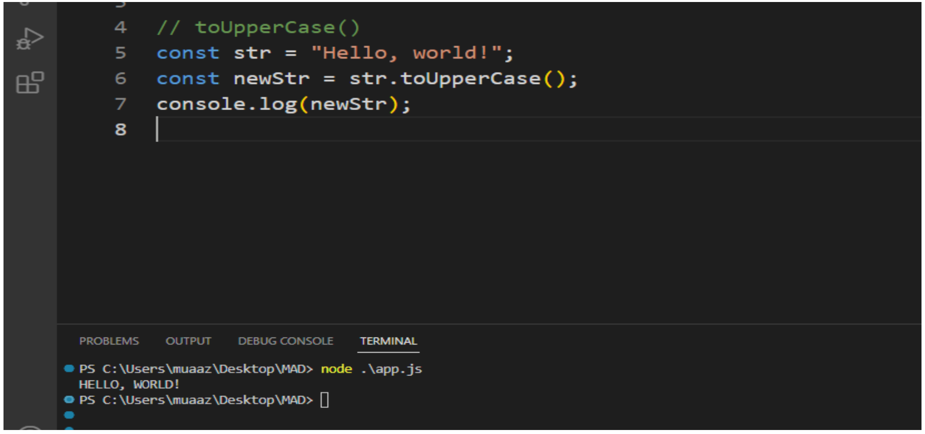
```
// toLowerCase()
const str = "HELLO, WORLD!";
const newStr = str.toLowerCase();
console.log(newStr);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
hello, world!
PS C:\Users\muaaz\Desktop\MAD>
```

toUpperCase

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

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



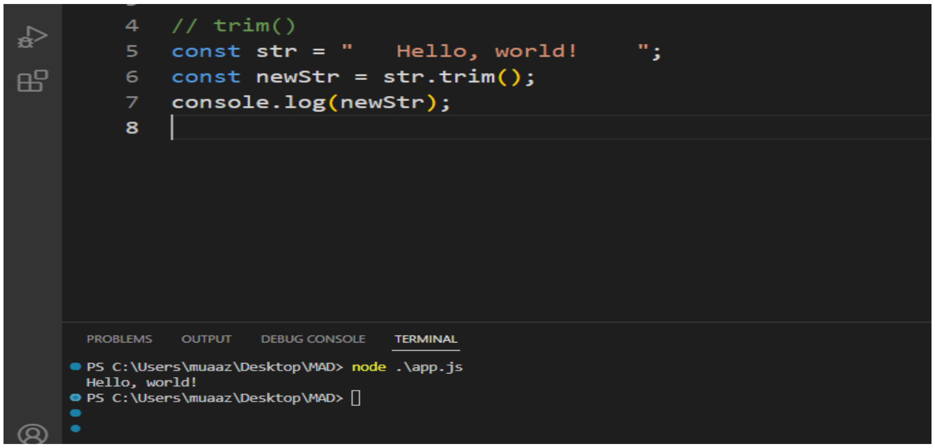
```
// toUpperCase()
const str = "Hello, world!";
const newStr = str.toUpperCase();
console.log(newStr);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
HELLO, WORLD!
PS C:\Users\muaaz\Desktop\MAD>
```

trim

This method removes whitespace from both ends of a string.

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



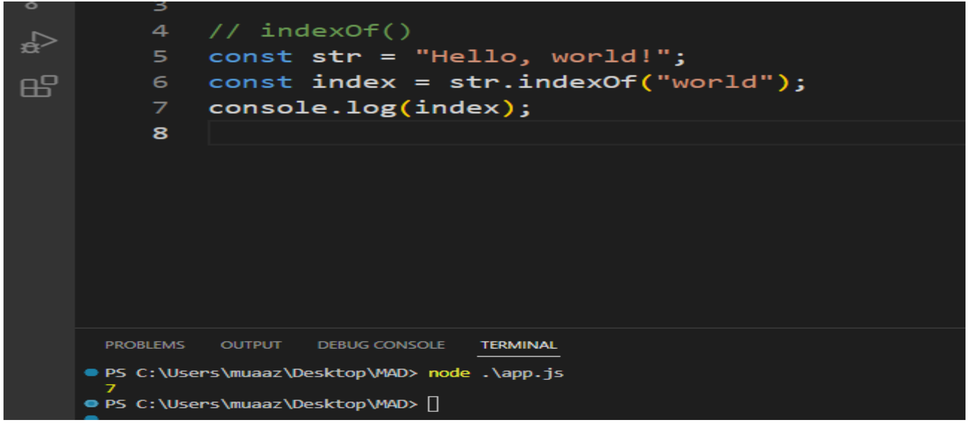
```
// trim()
const str = "    Hello, world!     ";
const newStr = str.trim();
console.log(newStr);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
Hello, world!
PS C:\Users\muaaz\Desktop\MAD> ▯
```

indexOf

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

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



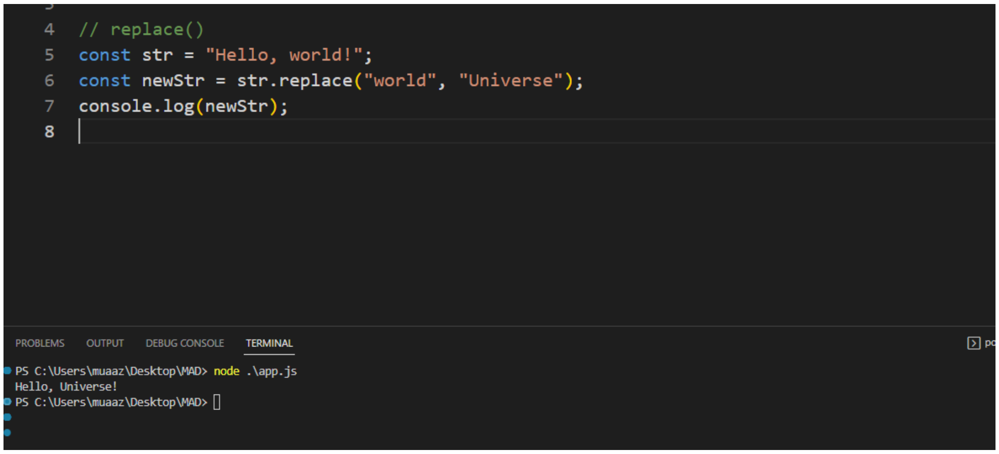
```
// indexOf()
const str = "Hello, world!";
const index = str.indexOf("world");
console.log(index);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
7
PS C:\Users\muaaz\Desktop\MAD> ▯
```

replace

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

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



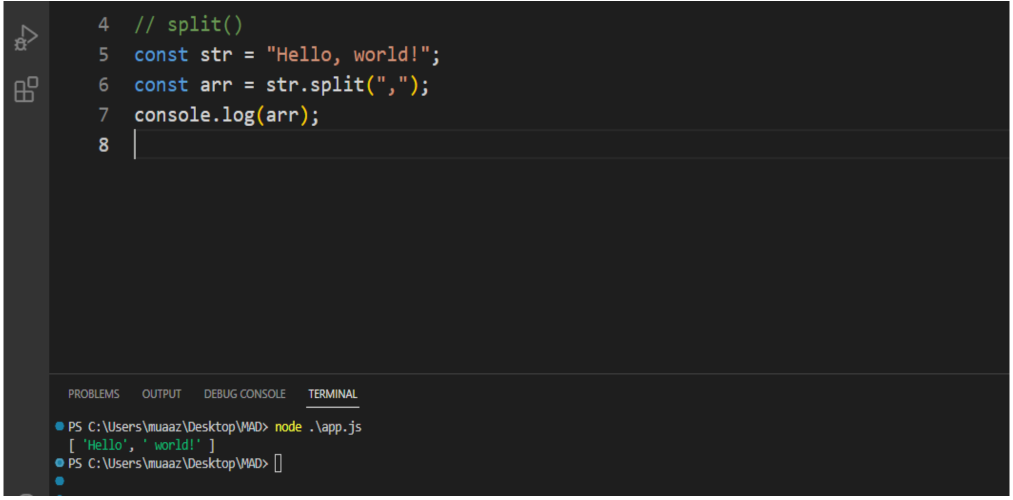
```
// replace()
const str = "Hello, world!";
const newStr = str.replace("world", "Universe");
console.log(newStr);
```

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
Hello, Universe!
PS C:\Users\muaaz\Desktop\MAD> 
```

split

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

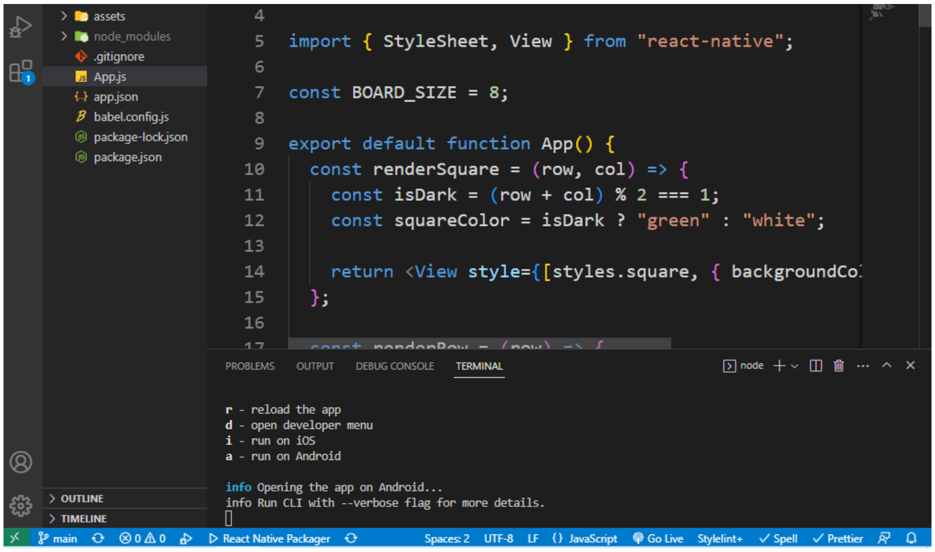
```
const str = "Hello, world!";
const arr = str.split(",");
console.log(arr);
```



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

```
PS C:\Users\muaaz\Desktop\MAD> node .\app.js
[ 'Hello', ' world!' ]
PS C:\Users\muaaz\Desktop\MAD> 
```

Q3: Design chess board UI in react native?



```
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, { backgroundCo
  };

  const renderRow = (row) => {
```

```
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.
```

```
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,
},
});
```