

Q1: Array functions in JavaScript with examples.

**push():** **adds one or more elements to the end of an array**

let fruits = ["apple", "banana"];

fruits.push("orange"); // adds "orange" to the end of the array

console.log(fruits); // output: ["apple", "banana", "orange"]

**pop():** **removes the last element from an array and returns it**

let fruits = ["apple", "banana", "orange"];

let lastFruit = fruits.pop(); // removes "orange" and returns it

console.log(fruits); // output: ["apple", "banana"]

console.log(lastFruit); // output: "orange"

**shift(): removes the first element from an array and returns it**

let fruits = ["apple", "banana", "orange"];

let firstFruit = fruits.shift(); // removes "apple" and returns it

console.log(fruits); // output: ["banana", "orange"]

console.log(firstFruit); // output: "apple"

**unshift(): adds one or more elements to the beginning of an array**

let fruits = ["banana", "orange"];

fruits.unshift("apple"); // adds "apple" to the beginning of the array

console.log(fruits); // output: ["apple", "banana", "orange"]

splice(): adds or removes elements from an array at a specified index

**splice(): adds or removes elements from an array at a specified index**

let fruits = ["apple", "banana", "orange"];

fruits.splice(1, 1, "grape", "kiwi"); // removes "banana" and adds "grape" and "kiwi" at index 1

console.log(fruits); // output: ["apple", "grape", "kiwi", "orange"]

**slice(): returns a new array containing a portion of an existing array**

let fruits = ["apple", "banana", "orange", "grape", "kiwi"];

let citrusFruits = fruits.slice(2, 4); // returns a new array containing "orange" and "grape"

console.log(citrusFruits); // output: ["orange", "grape"]

**concat(): combines two or more arrays into a new array**

let fruits = ["apple", "banana"];

let moreFruits = ["orange", "grape", "kiwi"];

let allFruits = fruits.concat(moreFruits); // combines the two arrays into a new array

console.log(allFruits); // output: ["apple", "banana", "orange", "grape", "kiwi"]

**indexOf(): returns the index of the first occurrence of a specified element in an array, or -1 if it is not found**

let fruits = ["apple", "banana", "orange"];

let index = fruits.indexOf("banana"); // returns 1

console.log(index); // output: 1

Q2: String functions in JavaScript with example.

**length: returns the length of a string**

let str = "hello world";

console.log(str.length); // output: 11

**toUpperCase(): returns a string in uppercase**

let str = "hello world";

console.log(str.toUpperCase()); // output: "HELLO WORLD"

**toLowerCase(): returns a string in lowercase**

let str = "HELLO WORLD";

console.log(str.toLowerCase()); // output: "hello world"

**charAt(): returns the character at a specified index in a string**

let str = "hello world";

console.log(str.charAt(1)); // output: "e"

**indexOf(): returns the index of the first occurrence of a specified substring in a string, or -1 if it is not found**

let str = "hello world";

console.log(str.indexOf("world")); // output: 6

**substring(): returns a portion of a string between two specified indices**

let str = "hello world";

console.log(str.substring(0, 5)); // output: "hello"

**Question 3: create a chess board**

import React from 'react';

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

const ChessBoard = () => {

// Create a 2D array to represent the chess board

const board = Array(8).fill().map(() => Array(8).fill());

// Fill the board with alternating black and white squares

for (let i = 0; i < 8; i++) {

for (let j = 0; j < 8; j++) {

board[i][j] = (i + j) % 2 === 0 ? 'white' : 'black';

}

}

// Map the board array to create the chess board UI

const boardUI = board.map((row, rowIndex) => (

<View key={rowIndex} style={styles.row}>

{row.map((square, colIndex) => (

<View key={colIndex} style={[styles.square, { backgroundColor: square }, ((rowIndex === 0) && (colIndex === 0)) ? styles.topLeft : null]} />

))}

</View>

));

return (

<View style={styles.container}>

{boardUI}

</View>

);

};

const styles = StyleSheet.create({

container: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

},

row: {

flexDirection: 'row',

},

square: {

width: 40,

height: 40,

borderWidth: 1,

borderColor: 'black',

},

topLeft: {

borderTopWidth: 3,

borderLeftWidth: 3,

borderColor: 'red',

},

});

export default function App() {

return <ChessBoard />;

}