1. Write a JavaScript function that reverse a number.

```
function reverseNumber(num) {
    let reversedNum =
parseInt(num.toString().split('').reverse().join(''))
;
    return reversedNum;
}

console.log(reverseNumber(12345));
```

2. Write a JavaScript function that checks whether a passed string is palindrome or not?

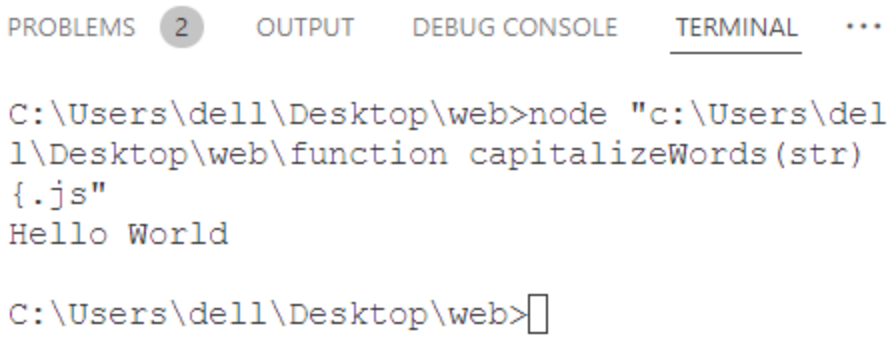
```
function isPalindrome(str) {

    let cleanStr =
str.toLowerCase().replace(/[\W_]/g, '');
    let reversedStr =
cleanStr.split('').reverse().join('');
```

```
    return cleanStr === reversedStr;
}


console.log(isPalindrome("madam"));
console.log(isPalindrome("hello"));
```

3. Write a JavaScript function that accepts a string as a parameter and converts the first

letter of each word of the string in upper case.

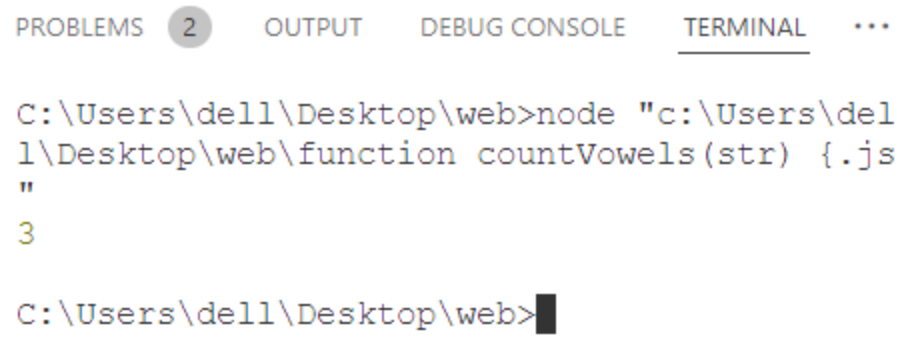
```
function capitalizeWords(str) {
    return str.replace(/\b\w/g, function(char) {
        return char.toUpperCase();
    });
}


console.log(capitalizeWords("hello world")); //
Output: "Hello World"
```



5. Write a JavaScript function that accepts a string as a parameter and counts the number of

vowels within the string.

```
function countVowels(str) {
    const vowels = 'aeiouAEIOU';
    let count = 0;
    for (let char of str) {
        if (vowels.includes(char)) {
            count++;
        }
    }
    return count;
}

console.log(countVowels("programming")); // Output: 3
(o, a, i)
```

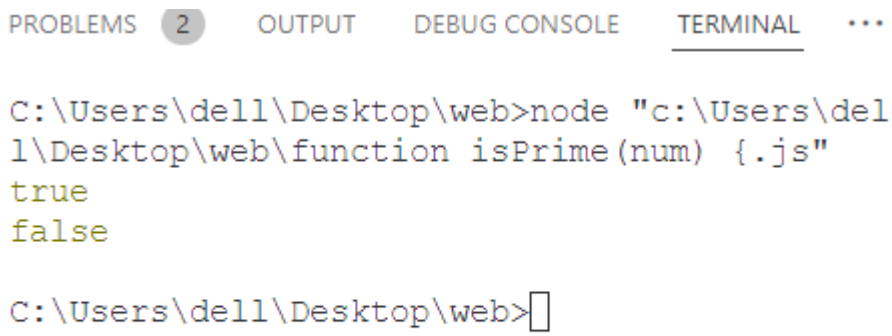


6. Write a JavaScript function that accepts a number as a parameter and check the number

is prime or not.

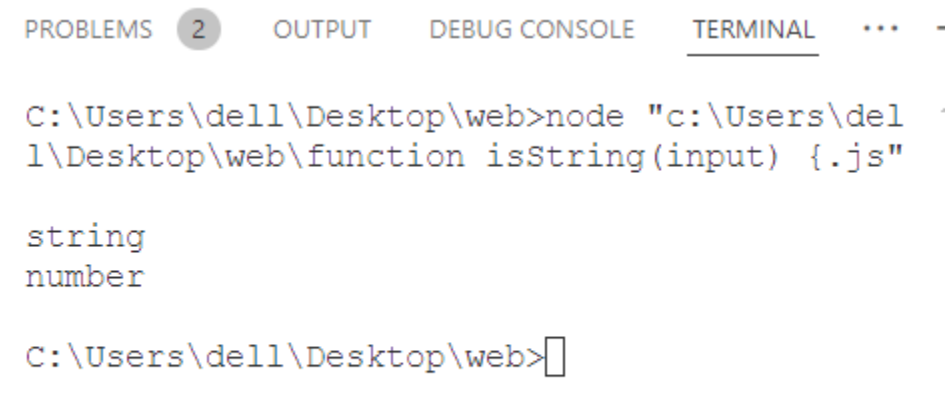
```
function isPrime(num) {
    if (num <= 1) {
        return false;
    }
    if (num <= 3) {
        return true;
    }
    if (num % 2 === 0 || num % 3 === 0) {
        return false;
    }
    let i = 5;
    while (i * i <= num) {
        if (num % i === 0 || num % (i + 2) === 0) {
            return false;
        }
        i += 6;
    }
    return true;
}
```

```
console.log(isPrime(11));
console.log(isPrime(15));
```



7. Write a JavaScript function which accepts an argument and returns the type.

```
function isString(input) {
  return typeof input
}
console.log(isString("Hello")); // true
console.log(isString(42)); // false
```



8. Write a JavaScript function which will take an array of numbers stored and find the

second lowest and second greatest numbers, respectively. Sample array : [1,2,3,4,5]

```
function findSecondNumbers(arr) {
    if (arr.length < 2) {
        return "Array length should be at least 2";
    }
    let sortedArr = arr.sort((a, b) => a - b);
    let secondLowest = sortedArr[1];
    let secondGreatest = sortedArr[arr.length - 2];
    return [secondLowest, secondGreatest];
}

console.log(findSecondNumbers([1, 2, 3, 4, 5]));
```

```
PROBLEMS 2   OUTPUT   DEBUG CONSOLE   TERMINAL   ···   -

C:\Users\dell\Desktop\web>node "c:\Users\del
l\Desktop\web\tempCodeRunnerFile.js"
[ 2, 4 ]

C:\Users\dell\Desktop\web>
```

9. Write a JavaScript function to check whether an `input` is an array or not

console.log(is_array([1, 2, 4, 0]));

```
function isArray(input) {
    return Array.isArray(input);
}

console.log(isArray("saugat")); // Output: false
console.log(isArray([1, 2, 4, 0])); // Output: true
```

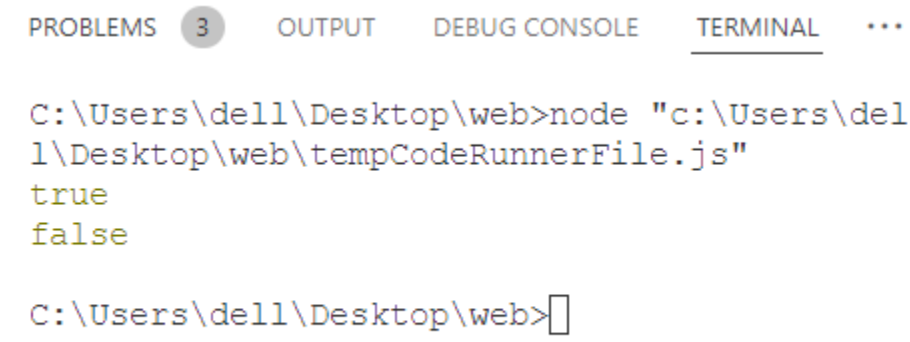
10. Write a simple JavaScript program to join all elements of the following array into a

string.

```
let array = [1, "red", 3, 4, 5];
let result = array.join(" ");
console.log(result);
```

12. Write a JavaScript function to check whether an `input` is a date object or not.

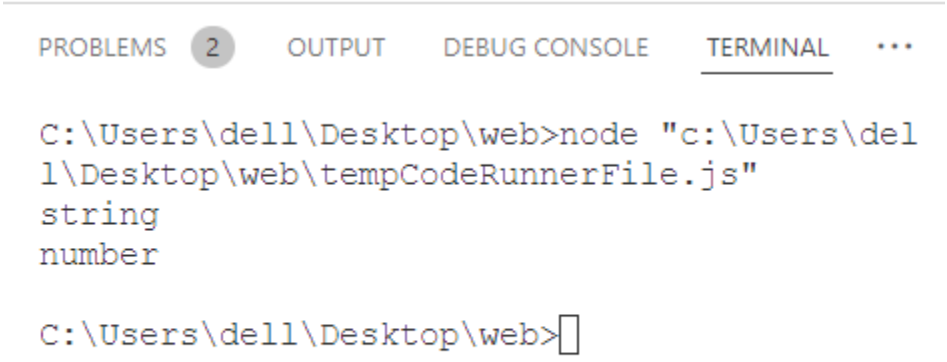
```
function isDate(input) {
  return input instanceof Date;
}

console.log(isDate(new Date())); // true
console.log(isDate("2024-07-23")); // false
```



13. Write a JavaScript function to check whether an `input` is a string or not

```
function isString(input) {
```

```
  return typeof input
}
console.log(isString("Hello")); // true
console.log(isString(42)); // false
```



14. Write a JavaScript program to list the properties of a JavaScript object.

Sample object:

var student = {

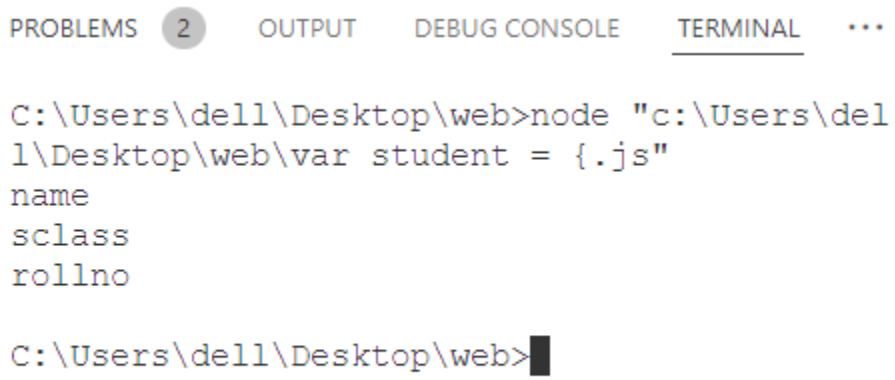
name :"saugat",,

sclass : vi,

rollno : 12 };

```
var student = {
  name: "saugat khanal",
  sclass: "xii",
  rollno: 85,
};
```

```
function listProperties(obj) {
  for (var prop in obj) {
    if (obj.hasOwnProperty(prop)) {
      console.log(prop);
    }
  }
}


listProperties(student);
```



15. Write a JavaScript program to delete the rollno property from the following object.

Also print the object before or after deleting the property.

Sample object:

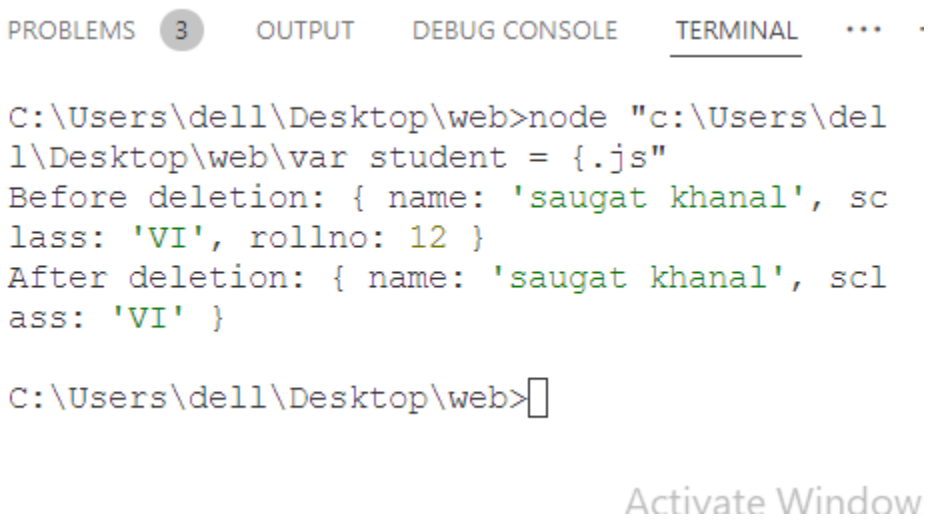
var student = {

name : "David Rayy",

sclass : "VI",

rollno : 12 };

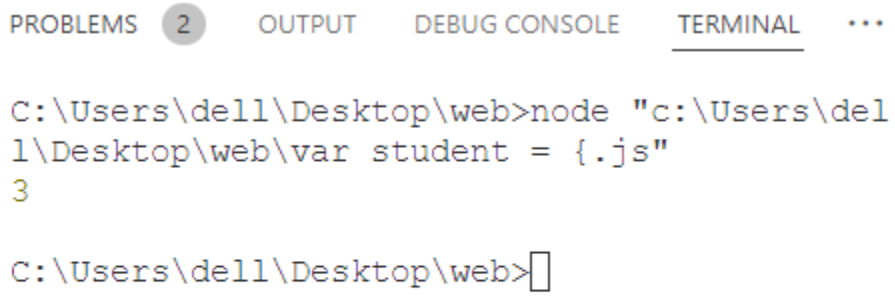
```
var student = {
   name: "saugat",
   sclass: "VI",
   rollno: 12
};
console.log("Before deletion:", student);
delete student.rollno;
console.log("After deletion:", student);
```



16. Write a JavaScript program to get the length of a JavaScript object.

```
var student = {
   name: "saugat khanal",
   sclass: "VI",
```

```
    rollno: 12
};
function getObjectLength(obj) {
    return Object.keys(obj).length;
}
console.log(getObjectLength(student));
```



17. Write a JavaScript program of window object such as alert(),

prompt(),confirm(),open(),close()

```
window.alert("Hello, world!");
var userInput = window.prompt("Enter your name:");
var userConfirmed = window.confirm("Are you sure?");
```

18. Write a JavaScript program in String object

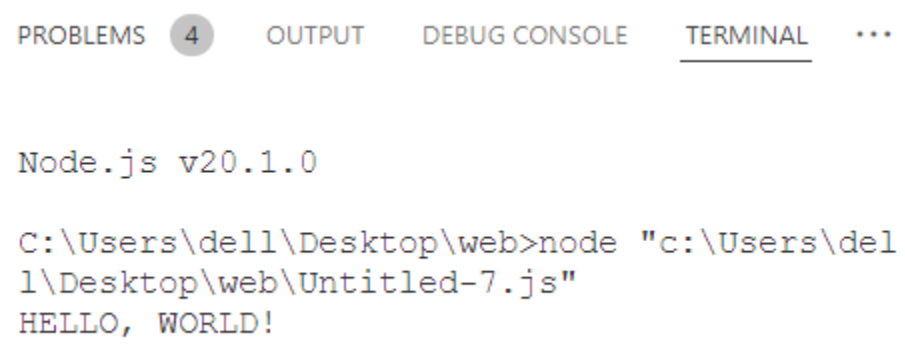
1)By string literal

2)By string object (using new keyword)

By string literals

var str1 = "Hello, world!";

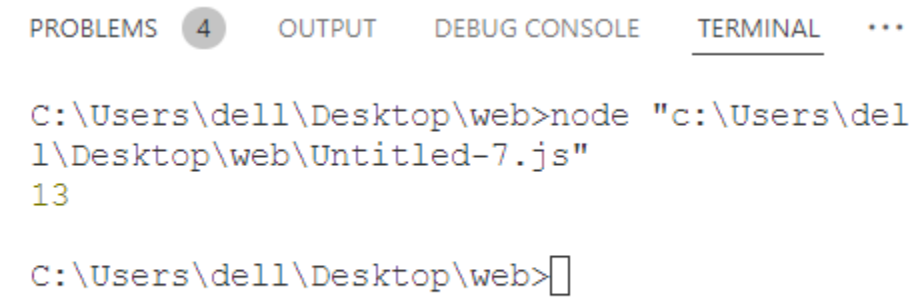
console.log(str1.toUpperCase());



by newkeyword

var str2 = new String("Hello, world!");

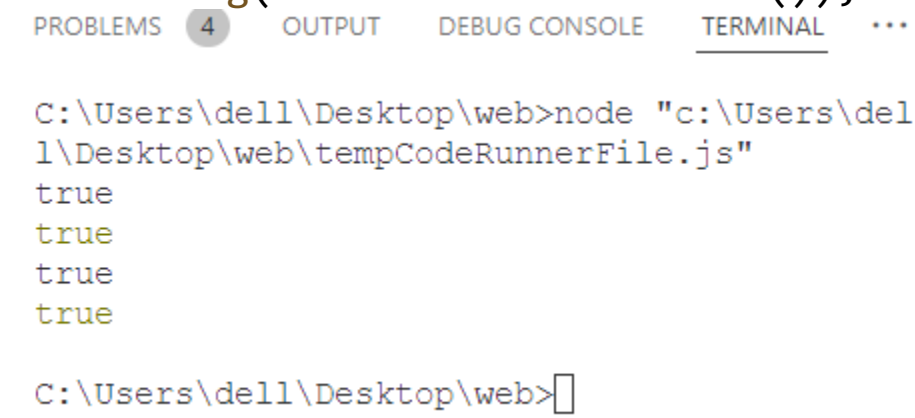
console.log(str2.length);



20. Write a JavaScript program in Boolean object such as toString, valueof.

```
var boolObj = new Boolean(true);
console.log(boolObj.toString());"
console.log(boolObj.valueOf());
```

```
var boolPrimitive = true;
console.log(boolPrimitive.toString());
console.log(boolPrimitive.valueOf());
```



41. Create an HTML page titled "MyWebsite" with a Meta tag specifying the author as "John

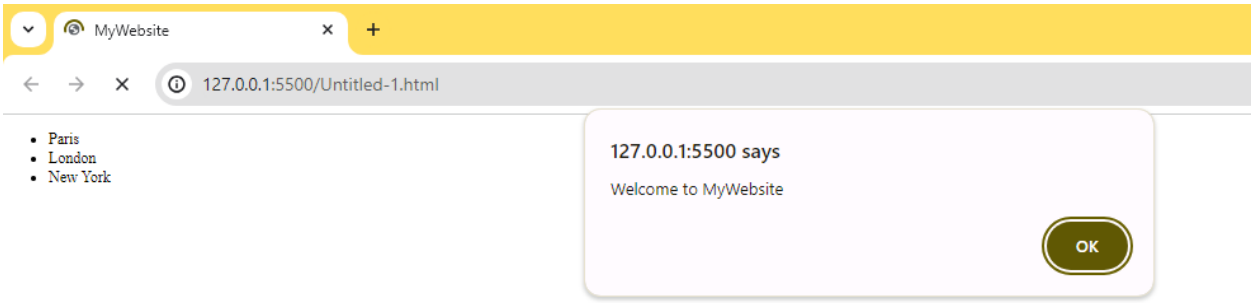
Doe". Include a div with the ID "content" that contains an unordered list of three cities:

Paris, London, and New York. After London, add a comment saying "Capital of

England." Additionally, add a script that displays an alert saying "Welcome to

MyWebsite" when the page loads.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta name="author" content="John Doe" />
    <title>MyWebsite</title>
  </head>
  <body>
    <div id="content">
      <ul>
        <li>Paris</li>
        <li>London</li>
        <!-- Capital of England -->
        <li>New York</li>
      </ul>
    </div>
    <script>
      window.onload = function () {
        alert("Welcome to MyWebsite");
      };
    </script>
  </body>
</html>
```



42. Develop an HTML document titled "Recipes" with a Meta tag indicating the author as

"Chef Smith". Inside a div labeled "recipeList", list three popular dishes: Spaghetti

Carbonara, Chicken Tikka Masala, and Caesar Salad. Link the Chicken Tikka Masala

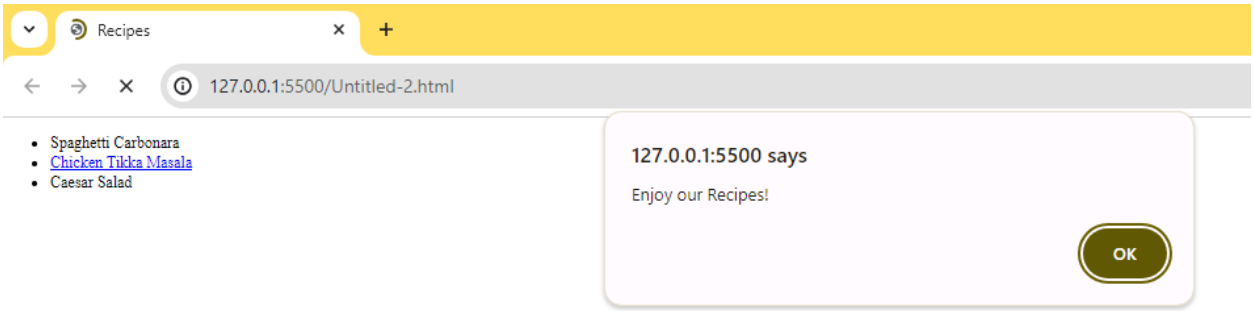
item to a recipe page on www.chickentikkamasala.com. Following the Caesar Salad item,

insert a comment stating "Healthy Option." Trigger an alert displaying "Enjoy our

Recipes!" upon the page's loading.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta name="author" content="Chef Smith" />
    <title>Recipes</title>
```

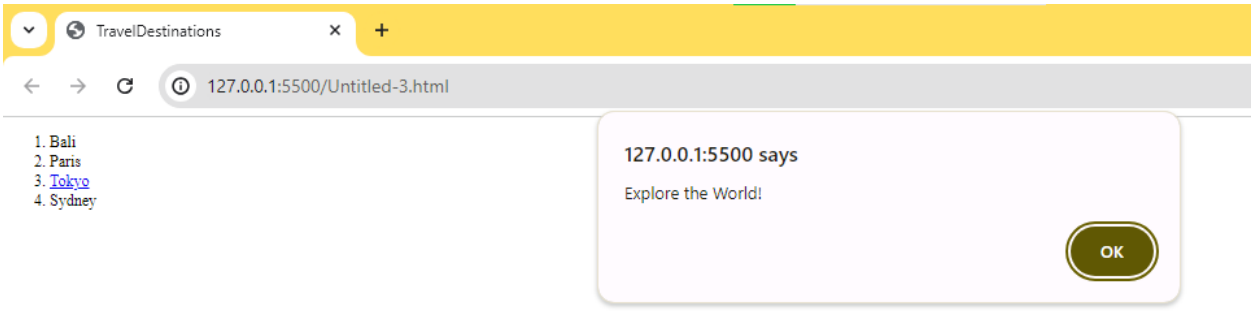
```
  </head>
  <body>
    <div id="recipeList">
      <ul>
        <li>Spaghetti Carbonara</li>
        <li>
          <a
href="http://www.chickentikkamasala.com">Chicken
Tikka Masala</a>
        </li>
        <li>
          Caesar Salad
          <!-- Healthy Option -->
        </li>
      </ul>
    </div>
    <script>
      window.onload = function () {
        alert("Enjoy our Recipes!");
      };
    </script>
  </body>
</html>
```



43. Design an HTML file named "TravelDestinations" featuring a Meta tag with the author set as "TravelerJane". Within the main container div, display an ordered list of vacation spots: Bali, Paris, Tokyo, and Sydney. Hyperlink the Tokyo item to www.visit-tokyo.com. Insert a comment after Sydney mentioning "Beautiful harbor views." Use JavaScript to present an alert message "Explore the World!" as soon as the page loads.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta name="author" content="TravelerJane" />
    <title>TravelDestinations</title>
  </head>
  <body>
    <div id="mainContainer">
      <ol>
        <li>Bali</li>
        <li>Paris</li>
        <li><a href="http://www.visit-
tokyo.com">Tokyo</a></li>
        <li>
          Sydney
          <!-- Beautiful harbor views -->
        </li>
      </ol>
    </div>
    <script>
      window.onload = function () {
        alert("Explore the World!");
      };
    </script>
  </body>
```

```
</html>
```



44. Write a JavaScript function named calculateArea that calculates the area of a rectangle.Declare variables length and width with values 5 and 3 respectively. Print the area using these variables.

```
function calculateArea() {
  var length = 5;
  var width = 3;
  var area = length * width;
  console.log("The area of the rectangle is: " +
area);
}

calculateArea();
```

45. Develop a JavaScript function named displayInfo that displays information about a user.Declare variables username with the value "JohnDoe", age with the value 30, and isSubscribed with the value true. Print the user information using these variables.

```
function displayInfo() {
  var username = "saugat";
  var age = 30;
  var isSubscribed = true;
  console.log("Username: " + username);
  console.log("Age: " + age);
  console.log("Subscribed: " + isSubscribed);
}

displayInfo();
```

46. What is client-side form validation? Develop a JavaScript function to validate a username field, ensuring it contains only letters and numbers and has a length between 4 and 12 characters. Also, validate a password field, requiring it to have a minimum length of 8 characters and at least one uppercase letter and one special character.

```
function validateUsername(username) {
  var regex = /^[a-zA-Z0-9]{4,12}$/;
  return regex.test(username);
}
```

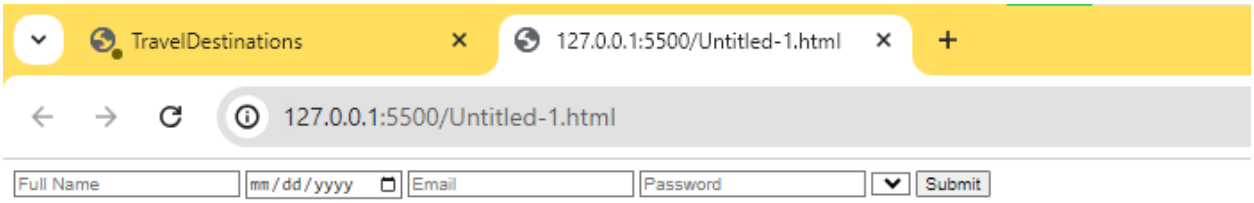
```
// Function to validate password

function validatePassword(password) {

  var passRegex = /^(?=.*[A-Z])(?=.*[\W_]).{8,}$/;

  return passRegex.test(password);

}
```

47. Explain the purpose of the canvas element in HTML5. Write JavaScript code to create a canvas element with the ID "myCanvas" and draw a rectangle with a width of 100 pixels and a height of 50 pixels, filled with a green color.

48. Design an HTML form for a user profile update. Include fields for the user's full name,date of birth, email, password, and a dropdown for selecting a country. The password must be at least 8 characters long and include at least one uppercase letter, one lowercase letter, and one special character. The email should be in a valid format. Write aJavaScript function to validate the form before submission.

```
<form id="profileForm">
  <input type="text" id="fullName" placeholder="Full
Name" />
  <input type="date" id="dob" placeholder="Date of
Birth" />
  <input type="email" id="email" placeholder="Email"
/>
  <input type="password" id="password"
placeholder="Password" />
  <!-- include dropdown for country -->
  <select id="country">
    <!-- options for countries -->
```

```
  </select>

  <button onclick="validateForm()">Submit</button>
</form>
```



function validateForm() {

}

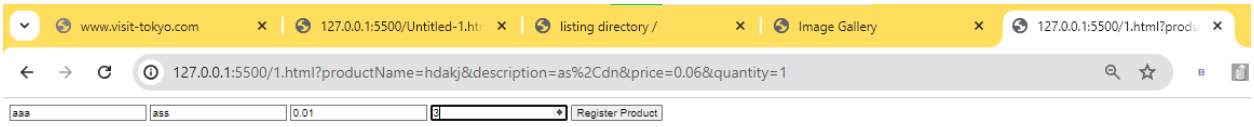
49. Create an HTML form for product registration. The form should contain fields for the product name, description, price, and quantity. The price field should only accept numeric values greater than zero, and the quantity should only accept positive integer values. Write a JavaScript function to validate the form data before submission.

<form id=productRegistrationForm>

    <input type=text name=productName >

    <input type=text name=description >

```
        <input type=number name=price min=0 step=".01">
        <input type=number name=quantity min=1>
        <button onclick='validateProductData()'>Register
Product</button>




</form>


function validateProductData (){
  // Get values from input fields - productName , description , price ,
quantity
  // Perform validation checks based on requirements specified.
}
```



50. Describe the XMLHttpRequest lifecycle states and their significance in AJAX operations. Develop a JavaScript function that demonstrates the handling of different XMLHttpRequest states and response statuses during an AJAX request.