**NAME: Sharvayu Zade**

**PRN: 23070521135**

**SEC: B(B2)**

**Practical 6 part 2: Regular Expressions in JavaScript**

**Introduction**

Regular Expressions (regex) are patterns used to **search**, **match**, and **manipulate text**. They are extremely powerful for validating input, extracting data, and performing complex string operations.

**Example in real life:**

● Checking if an email address is valid.

● Finding all phone numbers in a document.

● Replacing multiple spaces with a single space.

**Basic Regex Syntax**

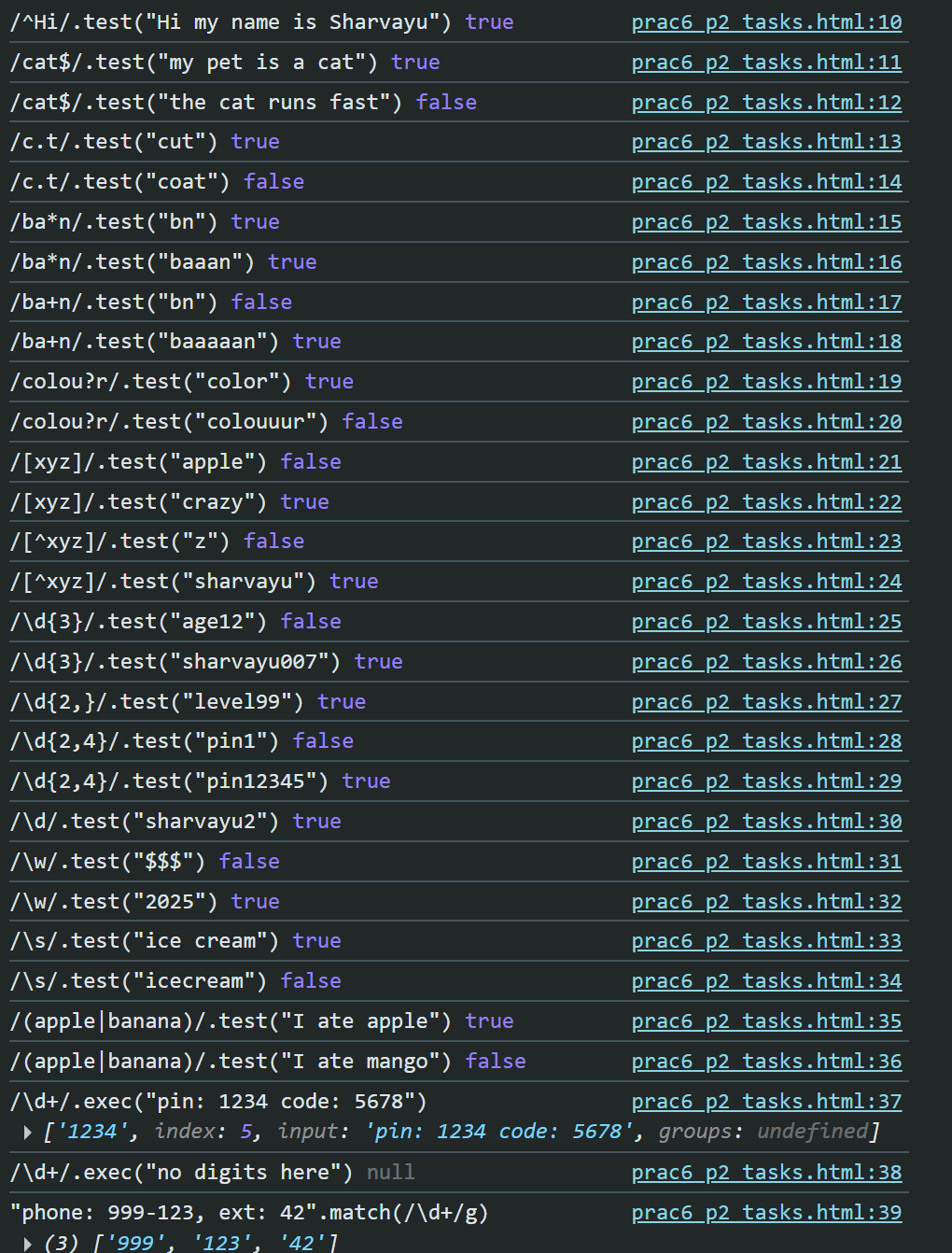
| **Symbol** | **Meaning** | **Example** |
| --- | --- | --- |
| ^ | Start of string | /^Hello/ matches "Hello world" |
| $ | End of string | /world$/ matches "Hello world" |
| . | Any character except newline | /h.t/ matches "hat", "hit" |
| \* | 0 or more occurrences | /go\*d/ matches "gd", "god", "good" |
| + | 1 or more occurrences | /go+d/ matches "god", "good" but **not** "gd" |
| ? | 0 or 1 occurrence | /colou?r/ matches "color" or "colour" |
| [abc]  [^abc]  {n} | Match a, b, or c  Match anything **except** a, b, or c  Exactly n times | /gr[ae]y/ matches "gray" or "grey"  /[^0-9]/ matches non-digits  /\d{3}/ matches three digits |

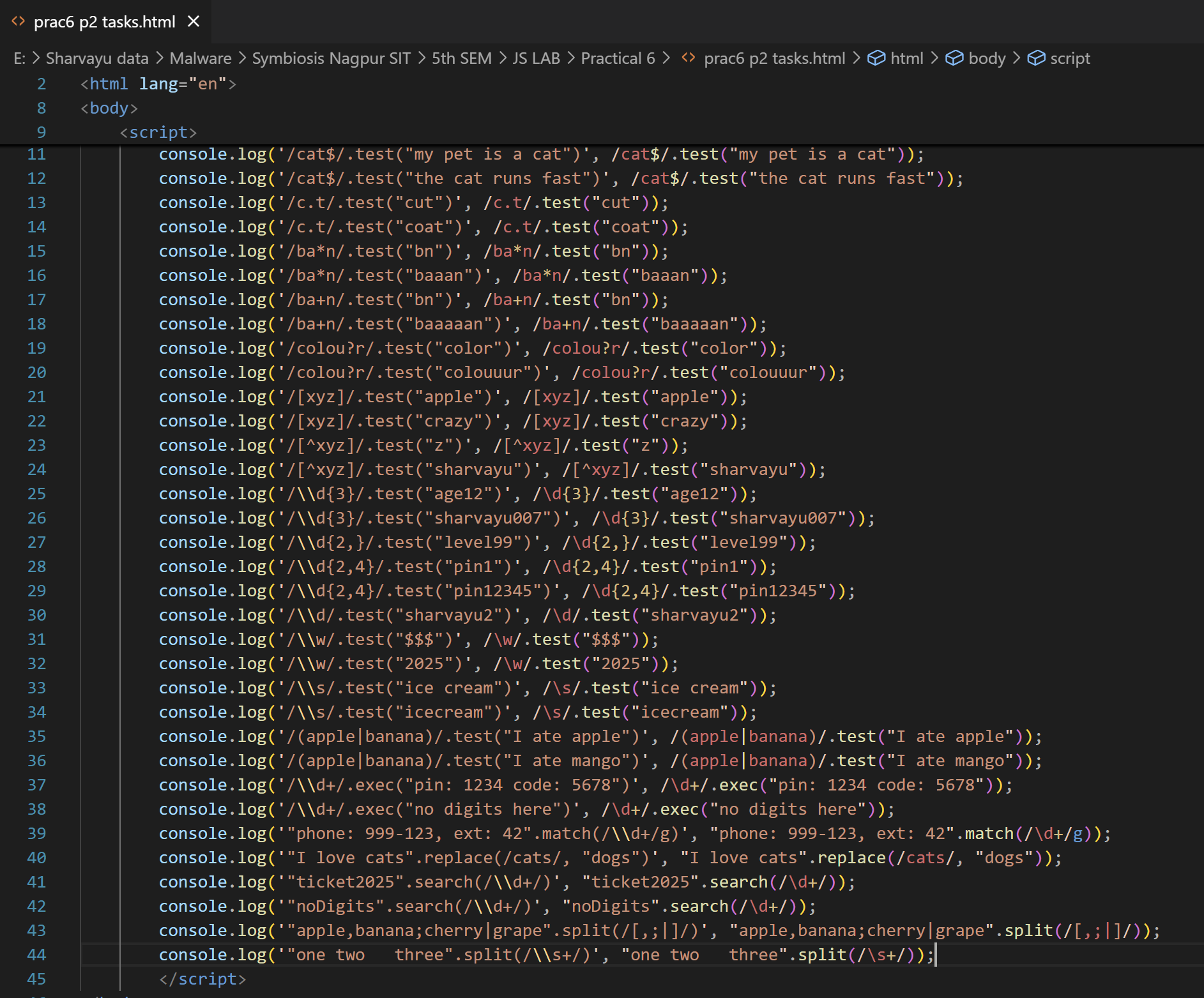
{n,} At least n times /\d{2,}/ matches two or more digits

| {n,m} | Between n and m times | /\d{2,4}/ matches 2, 3, or 4 digits |
| --- | --- | --- |
| \d | Digit (0-9) | /\d/ matches 1 |
| \w | Word character (letters, digits, underscore) | /\w/ matches A, 1 |
| \s | Whitespace (space, tab, newline) | /\s/ matches space |
| ` | ` | OR operator |

**JavaScript Regex Functions**

| **Method** | **Description** |
| --- | --- |
| test() | Returns true or false if regex matches. |
| exec() | Returns the first match as an array or null. |
| match() | Returns an array of all matches. |
| matchAll() | Returns an iterator with all matches + groups. |
| replace() | Replaces matched text with something else. |
| search() | Returns index of first match or -1. |
| split() | Splits string using a regex pattern. |



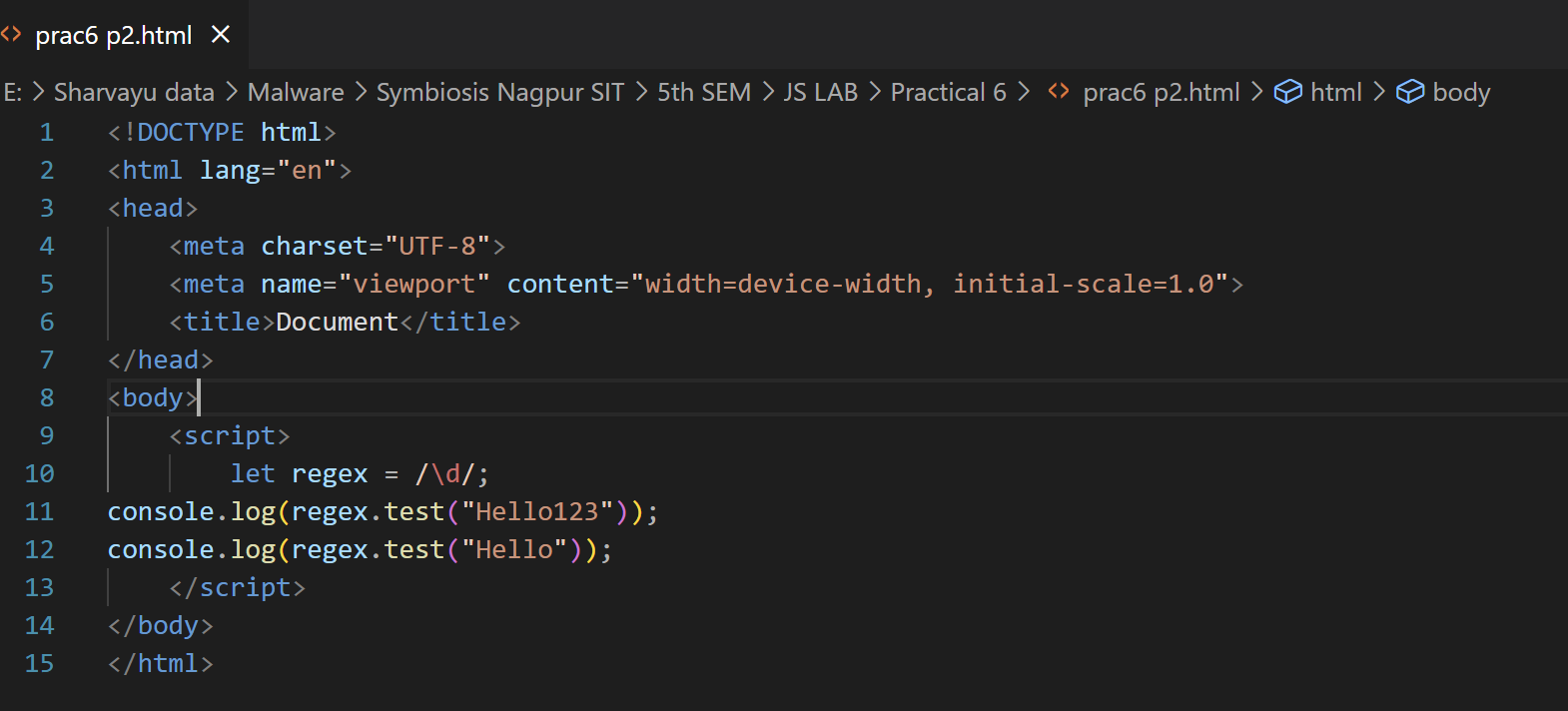
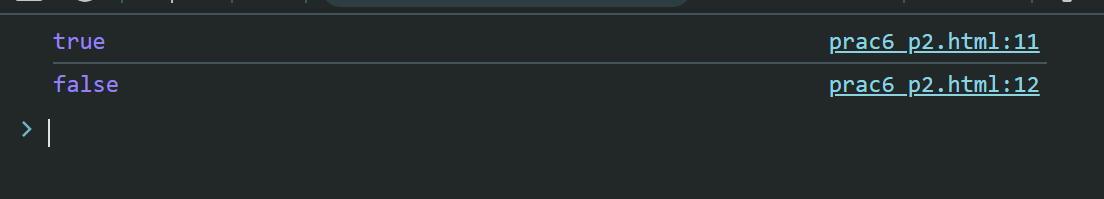


**Examples**

**Example 1: Test if a string contains digits**

let regex = /\d/;

console.log(regex.test("Hello123")); // true

console.log(regex.test("Hello")); // false 

**Example 2: Validate Email Address**

let emailRegex = /^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$/;

console.log(emailRegex.test("student@example.com")); // true

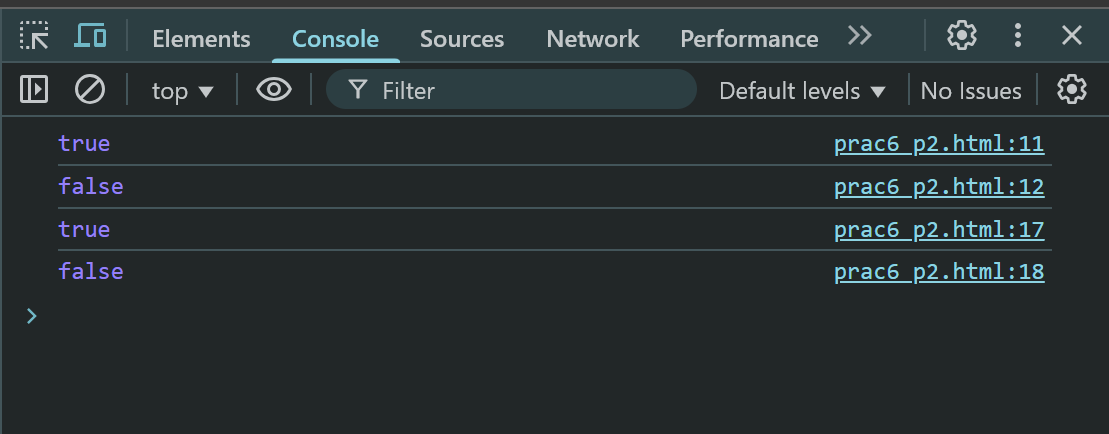
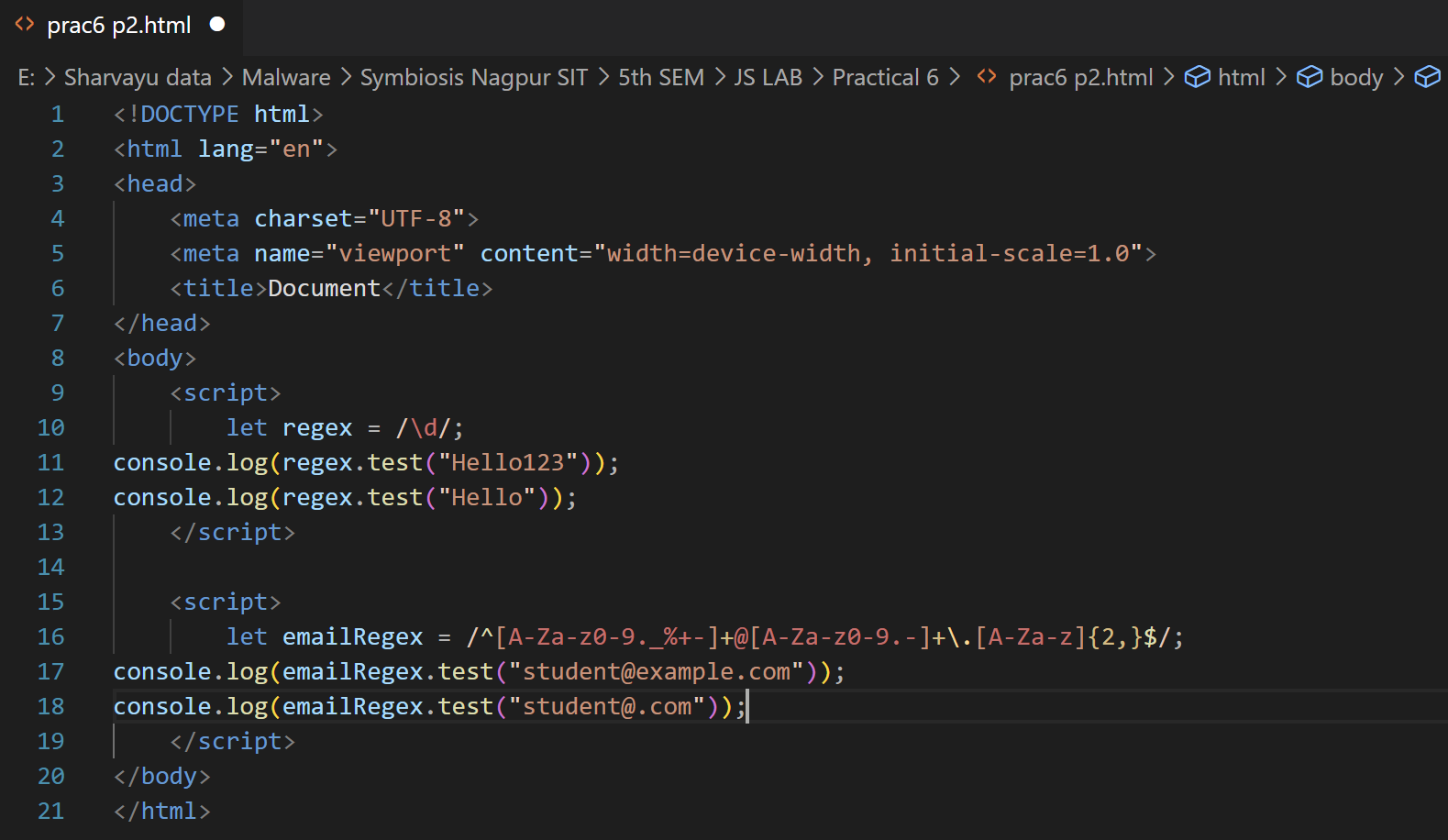
console.log(emailRegex.test("student@.com")); // false

**Explanation:**

● ^ → Start of string

● @ → Must contain @

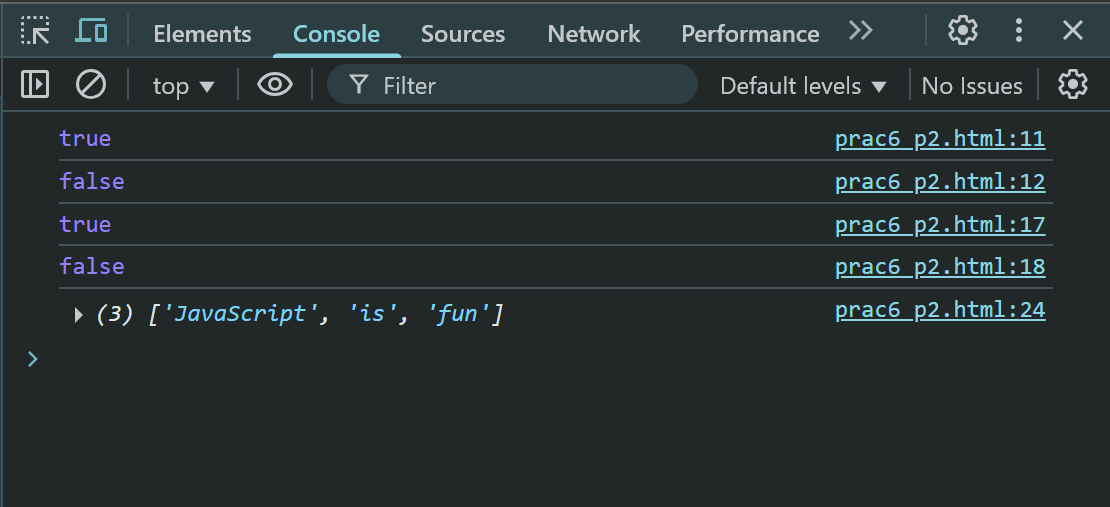
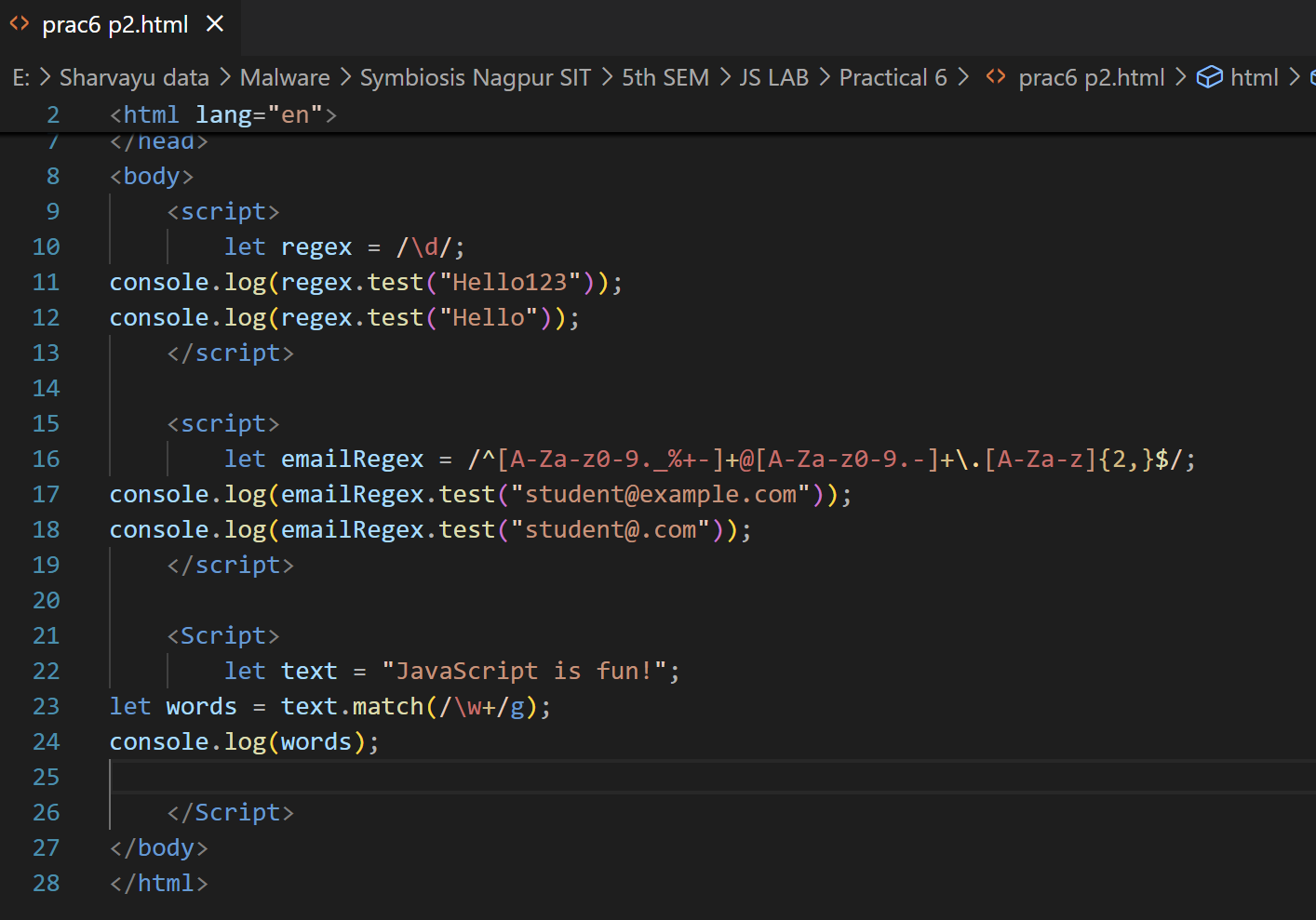
● . → Dot before domain

● $ → End of string 

**Example 3: Find all words in a string**

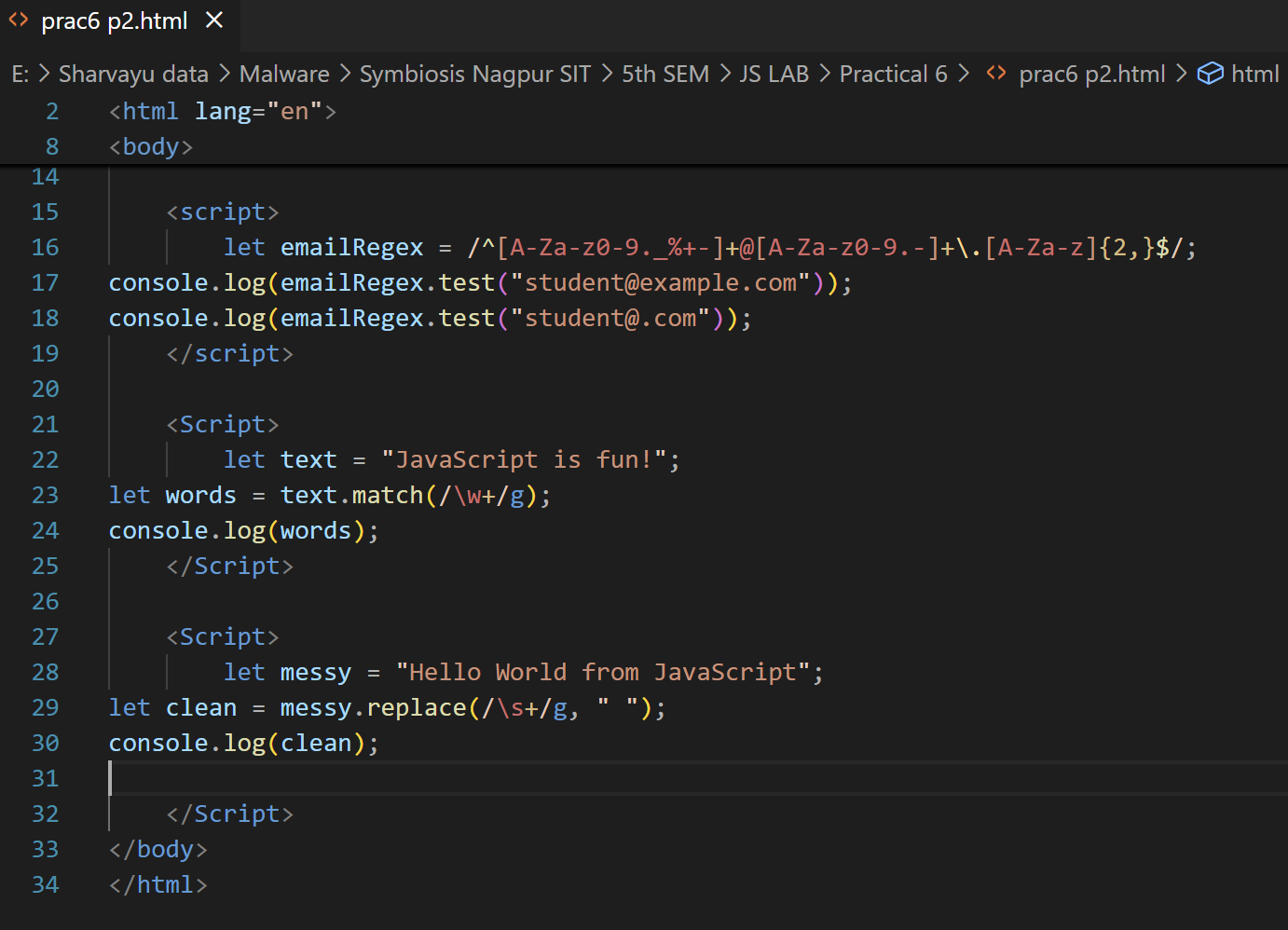
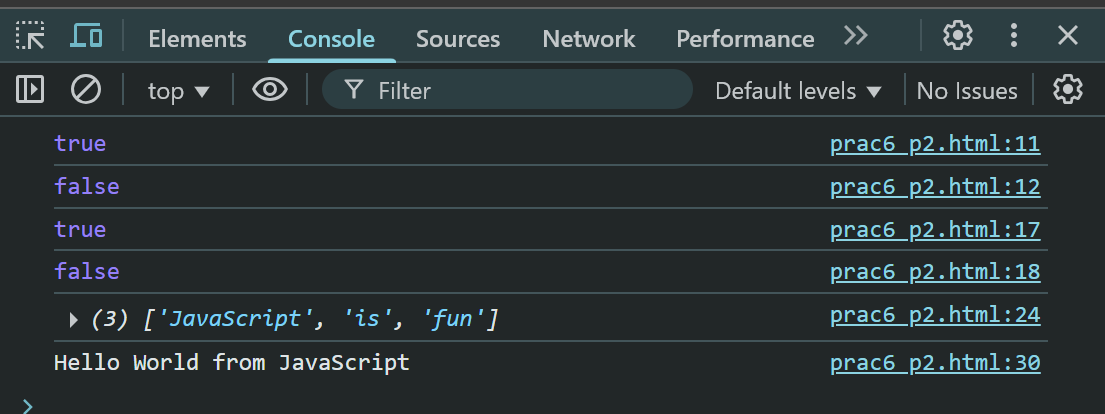
let text = "JavaScript is fun!";

let words = text.match(/\w+/g);

console.log(words); // ["JavaScript", "is", "fun"] 

**Example 4: Replace multiple spaces with single space**

let messy = "Hello World from JavaScript";

let clean = messy.replace(/\s+/g, " "); 

console.log(clean); // "Hello World from JavaScript"

**Lab Tasks**

**Task 1: Validate Mobile Number**

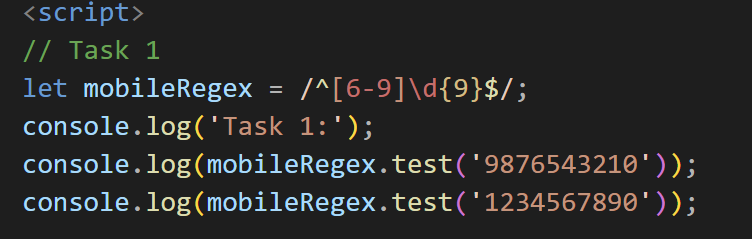
Create a regex to check if a number:

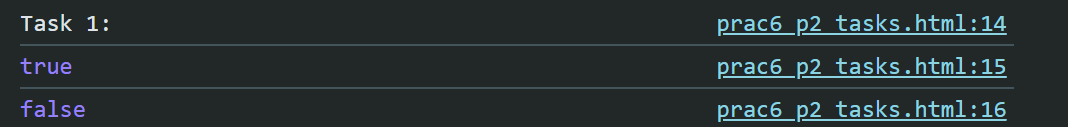
Starts with 6, 7, 8, or 9

Contains **exactly 10 digits**

**Example Valid:** 9876543210

**Example Invalid:** 1234567890



**Hint:**

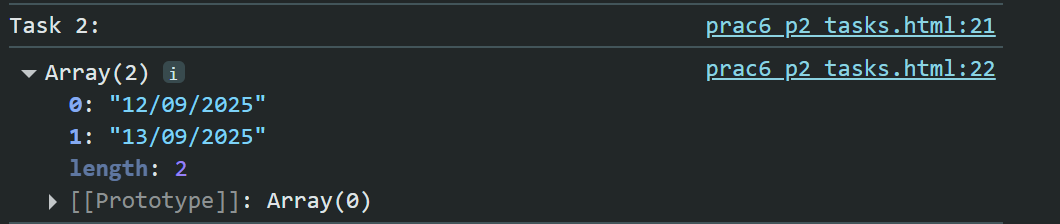
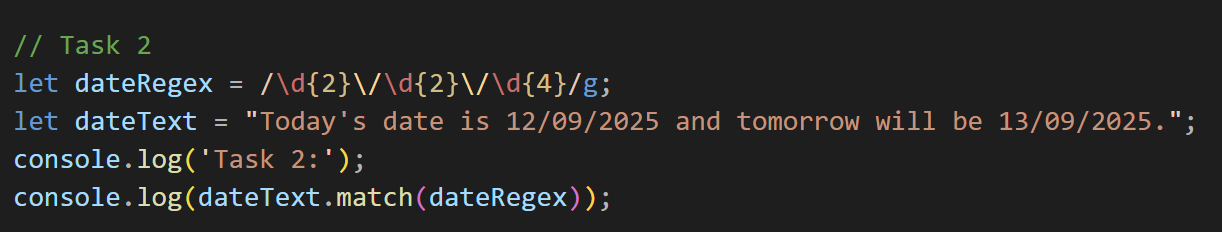
let mobileRegex = /^[6-9]\d{9}$/;

**Task 2: Extract Dates from Text**

● Input: "Today's date is 12/09/2025 and tomorrow will be 13/09/2025."

● Output: ["12/09/2025", "13/09/2025"]

**Hint:**

let dateRegex = /\d{2}\/\d{2}\/\d{4}/g; 

**Task 3: Password Strength Checker**

Rules:

● Minimum 8 characters

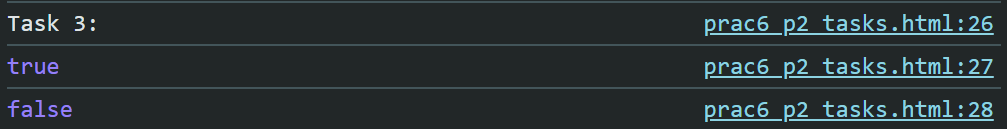
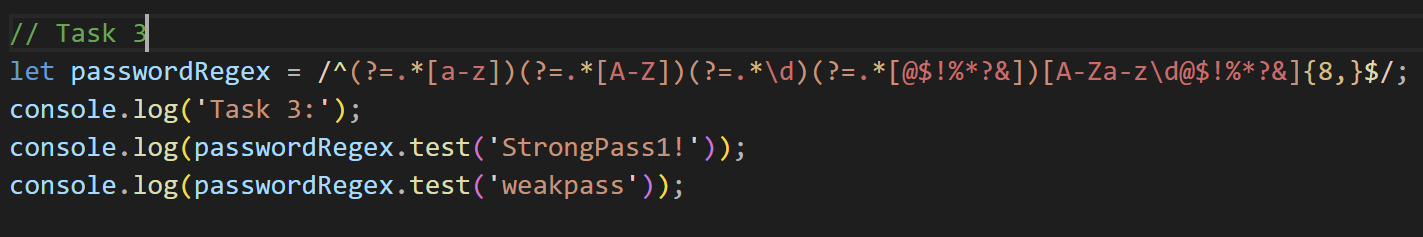
● At least 1 uppercase, 1 lowercase, 1 number, and 1 special character

**Regex:**

let passwordRegex = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{8,}$/;

**Examples:**

● "StrongPass1!" → Valid

● "weakpass" → Invalid 

**Task 4: Find Duplicate Words**

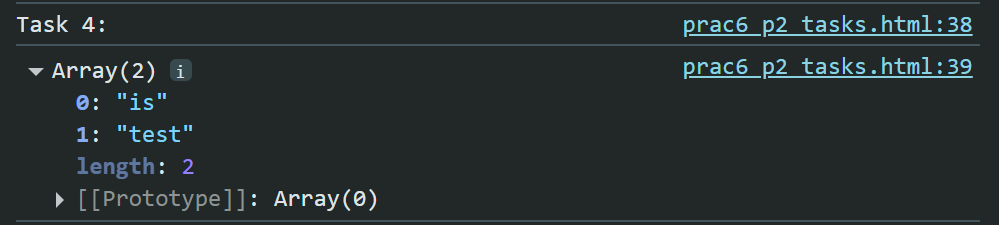
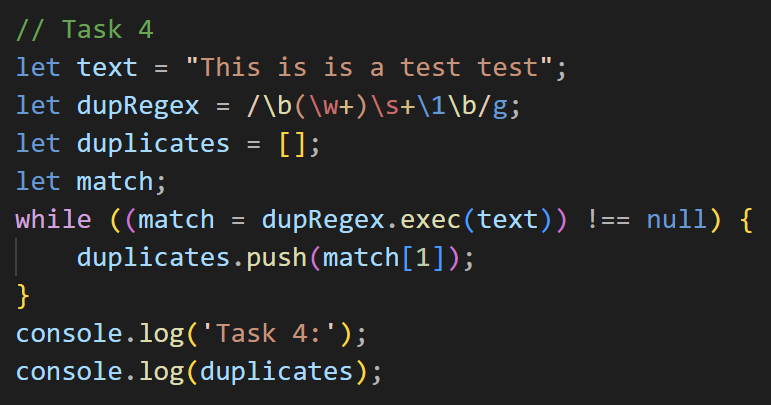
● Input: "This is is a test test"

● Output: ["is", "test"]

**Hint:**

let text = "This is is a test test";

let dupRegex = /\b(\w+)\s+\1\b/g;

console.log(text.match(dupRegex)); // ["is", "test"] 

**Task 5: Extract Hashtags from a Tweet**

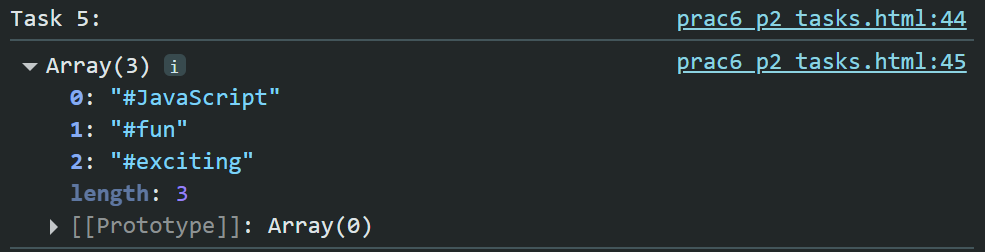
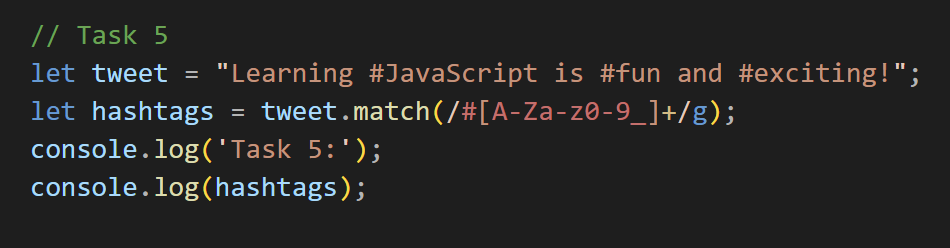
● Input: "Learning #JavaScript is #fun and #exciting!"

● Output: ["#JavaScript", "#fun", "#exciting"]

**Hint:**

let tweet = "Learning #JavaScript is #fun and #exciting!";

let hashtags = tweet.match(/#[A-Za-z0-9\_]+/g);

console.log(hashtags); 

**Practice Problems**

| **Problem** | **Expected Input** | **Expected Output** |
| --- | --- | --- |
| Validate PIN | "440001" | Valid |
| Validate URL | "https://google.com" | Valid |
| Extract Numbers | "Order 123 costs $45" | ["123", "45"] |
| Extract Emails | "Contact us at help@abc.com or sales@xyz.org" | ["help@abc.com",  "sales@xyz.org"] |



