Varun Sharma

Full guide to

JavaScript

REGEX

Varun Sharma

RegEx ...?

A JavaScript Regular Expression (RegEx) is a tool for **matching patterns in strings**. It allows you to search, extract, and **manipulate text efficiently**.

Essentially, it is a sequence of characters defining a search pattern, commonly used for tasks like **validating input**, parsing data, or finding and **replacing text** in strings.

Let's discover some of its key features >>

Varun Sharma

Patterns

index.js

```
/* There's two ways in which we can define regular
   expression statements, known as patterns: */

//Double forward slashes
  const regexExample1 = /pattern/;

//Using RegExp constructor
  const regexExample2 = new RegExp("pattern");
```

Swipe >>

Varun Sharma

Matches

index.js

```
/* The .test() method allows us to check if a
   string matches the defined pattern */

const string = “Follow CodeWithEmil”;

const regex_1 = /follow/;

const regex_2 = /CodeWithEmil/;


console.log(regex_1.test(string));

// Returns “false”: RegEx are case-sensitive

console.log(regex_2.test(string));

// Returns “true”
```

Swipe >>

Varun Sharma

Classes

index.js

```
/* This technique, similar to arrays, allows us to
   match any element from a pattern to a text */

// Matching vowels
const regex = /[aeiou]/;
console.log(regex.test("hello")); //True (has vowels)
console.log(regex.test("wrld")); //False (no vowels)

//Matching any digit
const regex = /[0-9]/;
console.log(regex.test("hello")); //False (no digits)
console.log(regex.test("123")); //True (has digits)
```

Swipe >>

Varun Sharma

Flags

index.js

```
/* Appended after the closing slash, flags modify
   how the regular expression behaves */

const text = “Hello world! New world I see...”;
const replaceRegex = /world/g;  //global flag used

console.log(
  text.replace(replaceRegex, “Test”);
); // Output: Hello Test! New Test I see...

/* The global flag modifies every instance of
   matching text instead of only the first one */
```

Swipe >>

Varun Sharma

index.js

```
/* Use the “i” flag for case-insensitive matching */

const text = “I love Tacos, Tacos, Tacos”;
const replaceRegex = /tacos/i;  //i flag used

console.log(
  replaceRegex.test(text);
); // Output: true
```

Swipe >>

Varun Sharma

Quantifiers

index.js

```
/* Specifies how many times a character should
   appear in the input string */

const regex = /a{2,4}/;

console.log(regex.test("a")); //False
console.log(regex.test("aa")); //True
console.log(regex.test("aaa")); //True
console.log(regex.test("aaaa")); //True
console.log(regex.test("aaaaa")); //False

/* This program checks whether the number of
   times "a" appears is between 2 and 4 */
```

Swipe >>

Varun Sharma

index.js

```
/* More types of quantifiers */

// "*" matches 0 or more of the preceding character
  console.log(/ho*//.test("hoo")); //True
  console.log(/ho*//.test("h")); //True

// "+" matches 1 or more of the preceding character
  console.log(/ho+/.test("hoo")); //True
  console.log(/ho+/.test("h")); //False

// "?" matches 0 or 1 of the preceding character
  console.log(/colou?r/.test("colour")); //True
  console.log(/colou?r/.test("color")); //True
```

Swipe >>

Varun Sharma

Anchors

index.js

```
/* Anchors are used to match a pattern with a
   particular position within the input string */

// ^ searches for the beginning of a string
const regex = /^hello/;
console.log(regex.test("hello world")); //True
console.log(regex.test("world hello")); //False

// $ searches for the end of a string
const regex = /hello$/;
console.log(regex.test("hello world")); //False
console.log(regex.test("world hello")); //True
```

Swipe >>

Varun Sharma

Groups

index.js

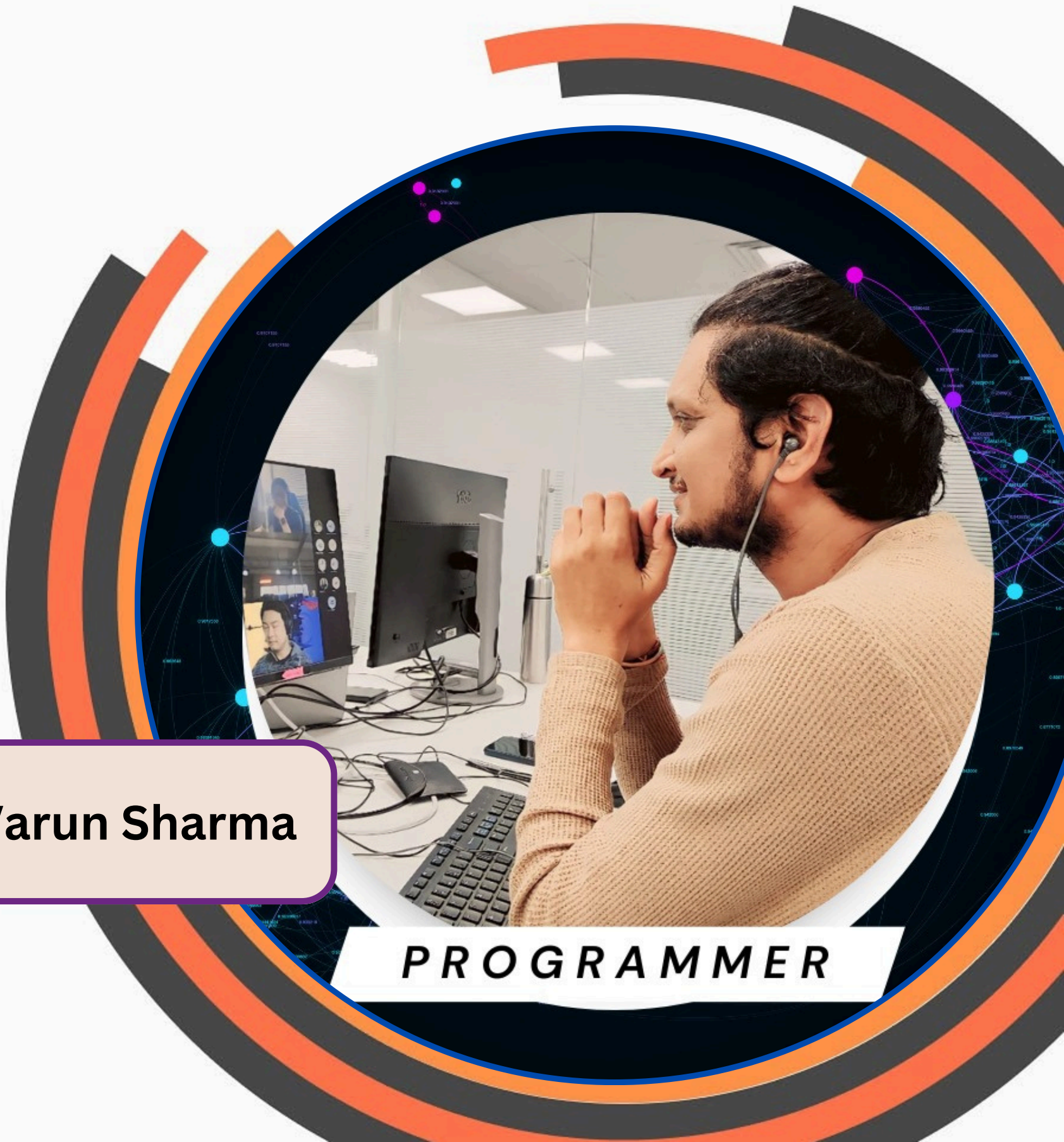
```
/* By grouping things, you can match one of several
   options or apply rules (like repetition) to the
   whole group. */
```

```
const regex = /(hello|world)/;

console.log(regex.test("hello world")); //True
console.log(regex.test("My world is pretty")); //True
console.log(regex.test("Hola, hello, salut")); //True
console.log(regex.test("Follow for more!")); //False

/* This program verifies whether the string passed
   contains or not EITHER "hello" or "world" */
```

Swipe >>

Thanks for reading!

Don’t forget to leave your ❤ !



Varun Sharma