

Regular Expressions in JavaScript

Regular Expressions (regex) are patterns used to match character combinations in strings. In JavaScript, regular expressions provide a powerful way to search, validate, and manipulate text by defining patterns that can match specific string sequences.

Syntax of Regular Expressions:

A regular expression is usually written between two forward slashes (/), like this: `/pattern/`. The pattern can be any sequence of characters and special symbols that defines what you're looking for in the string.

For example:

- `/abc/` matches the exact sequence "abc".
- `/d/` matches any digit.

Basic Components of Regular Expressions

1. Literals:

- Normal characters are treated as literals (e.g., `/hello/` will match the exact string "hello").
- Case sensitivity: `/hello/` will match "hello" but not "Hello" or "HELLO".

2. Metacharacters: These characters have special meaning:

- `.` (dot): Matches any character except newlines.
 - Example: `/h.llo/` will match "hello", "hxlllo", etc.
- `^`: Anchors the regex to the start of the string.
 - Example: `^hello/` matches "hello" only if it appears at the beginning.
- `$`: Anchors the regex to the end of the string.
 - Example: `/world$/` matches "world" only if it appears at the end.
- `[]`: Defines a character class, matching any one of the characters inside the brackets.
 - Example: `/[aeiou]/` matches any vowel (a, e, i, o, u).
- `|`: Logical OR; matches either the pattern on the left or the right.
 - Example: `/cat|dog/` matches either "cat" or "dog".
- `()` (parentheses): Grouping constructs to apply operators to a sub-pattern.
 - Example: `/(abc)+/` matches one or more occurrences of "abc".

3. Quantifiers:

- `*`: Matches 0 or more of the preceding element.
 - Example: `/a*/` matches "aaa", "aa", or an empty string.
- `+`: Matches 1 or more of the preceding element.
 - Example: `/a+/` matches "a", "aa", "aaa", etc.
- `?`: Matches 0 or 1 of the preceding element.
 - Example: `/a?/` matches an empty string or "a".
- `{n}`: Matches exactly n occurrences of the preceding element.
 - Example: `/a{3}/` matches exactly "aaa".
- `{n,}`: Matches n or more occurrences.
 - Example: `/a{2,}/` matches "aa", "aaa", "aaaa", etc.
- `{n,m}`: Matches between n and m occurrences.
 - Example: `/a{2,4}/` matches "aa", "aaa", or "aaaa".

4. Character Classes:

- **\d:** Matches any digit (equivalent to [0-9]).
- **\D:** Matches any non-digit.
- **\w:** Matches any word character (letters, digits, and underscores, equivalent to [A-Za-z0-9_]).
- **\W:** Matches any non-word character.
- **\s:** Matches any whitespace character (spaces, tabs, line breaks).
- **\S:** Matches any non-whitespace character.

5. Escape Sequences:

- If you need to use a special character as a literal, you need to escape it with a backslash (\).
 - Example: `\.` will match a literal period (.), not any character.

Creating a Regular Expression:

You can create a regular expression in two ways in JavaScript:

1. **Using a Regular Expression Literal:** This is the most common and straightforward way to define a regular expression.

```
const regex = /hello/;
```

2. **Using the RegExp Constructor:** This method is useful if you want to create a regular expression dynamically (e.g., with variables).

```
const regex = new RegExp("hello");
```

```
const digitsRegex = /^\d+$/;
```

```
console.log(regex.test("hello")); // true
```

```
console.log(digitsRegex.test("12a45")); // false
```

JS FORM VALIDATION

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>Document</title>
</head>
<body>
  <form name="form1">
    <label for="yourname">Your Name:</label>
    <input id="yourname" type="text" name="yourname">
    <small id="error" style="color: red;"></small> <!-- Error
message container -->
    <button id="submit" type="submit">Submit</button>
  </form>
```

```

<script>
    // Select the error element
    const errorElement = document.getElementById('error');

    // Add input event listener for real-time validation
    document.getElementById('yourname').addEventListener('input',
function () {
    const name = this.value.trim(); // Get the value of the
input field

    if (name === "") {
        errorElement.textContent = "Name must be filled out";
// Display error message
    } else {
        errorElement.textContent = ""; // Clear error message
    }
    });
</script>
</body>
</html>

```

Using regex

```

document.getElementById('myForm').addEventListener('submit',
function(event) {
    event.preventDefault(); // Prevent form from submitting to allow
validation

    // Get values from the form fields
    const email = document.getElementById('email').value;
    const phone = document.getElementById('phone').value;
    let errorMessage = '';

    // Regular expression for email validation
    const emailRegex =
/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

    // Regular expression for phone number validation (Indian phone
number format)
    const phoneRegex = /^[6789]\d{9}$/;

    // Validate email

```

```

    if (!emailRegex.test(email)) {
        errorMessage += 'Please enter a valid email address.<br>';
    }

    // Validate phone number
    if (!phoneRegex.test(phone)) {
        errorMessage += 'Please enter a valid phone number (should start
with 6, 7, 8, or 9 and have 10 digits).<br>';
    }

    // Display error messages or success
    const errorElement = document.getElementById('error-message');
    if (errorMessage) {
        errorElement.innerHTML = errorMessage;
    } else {
        errorElement.innerHTML = 'Form submitted successfully!';
        // You can submit the form here if you want after the validation
        // this.submit();
    }
});

```

const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$/;

Form Validation

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form name="form1">
        <label for="yourname">Your Name:</label>
        <input id="yourname" type="text" name="yourname">
        <small id="error" style="color: red;"><!-- Error
message container -->
        <button id="submit" type="submit">Submit</button>
    </form>
</script>

```

```

        // Select the error element
        const errorElement = document.getElementById('error');

        // Add input event listener for real-time validation
        document.getElementById('yourname').addEventListener('input',
function () {
            const name = this.value.trim(); // Get the value of the
input field

            if (name === "") {
                errorElement.textContent = "Name must be filled out";
// Display error message
            } else {
                errorElement.textContent = ""; // Clear error message
            }
        });
    </script>
</body>
</html>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form name="form1">
        <label for="yourname">Your Name:</label>
        <input id="yourname" type="text" name="yourname">
        <small id="nameError" style="color: red;"></small> <!-- Name
error message container -->
        <br><br>

        <label for="email">Email:</label>
        <input id="email" type="text" name="email">
        <small id="emailError" style="color: red;"></small> <!-- Email
error message container -->
        <br><br>
    </form>

```

```

        <button id="submit" type="submit">Submit</button>
    </form>

    <script>
        // Select error elements for name and email
        const nameErrorElement = document.getElementById('nameError');
        const emailErrorElement =
document.getElementById('emailError');

        // Add input event listener for name field
        document.getElementById('yourname').addEventListener('input',
function () {
            const name = this.value.trim(); // Get the value of the
input field

            if (name === "") {
                nameErrorElement.textContent = "Name must be filled
out"; // Display error message
            } else {
                nameErrorElement.textContent = ""; // Clear error
message
            }
        });

        // Add input event listener for email field
        document.getElementById('email').addEventListener('input',
function () {
            const email = this.value.trim(); // Get the value of the
email input field
            const emailRegex =
/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/; // Email regex
pattern

            if (!emailRegex.test(email)) {
                emailErrorElement.textContent = "Please enter a valid
email address"; // Display error message
            } else {
                emailErrorElement.textContent = ""; // Clear error
message
            }
        });
    </script>
</body>

```

```
</html>
```