Regular Expression - Examples

Regex Used in Example 1

We have used the below regular expression pattern in example 1 to validate the email.

let regex = /^[a-z0-9]+@[a-z]+\.[a-z]{2,3}$/;

Users can follow the below explanation for the above regular expression.

* **^** - It is the start of the string.
* **[a-z0-9]+** - Any character between a to z and 0 to 9 at the start of the string.
* **@** - The string should contains ‘@’ character after some alphanumeric characters.
* **[a-z]+** - At least one character between a to z after the ‘@’ character in the string.
* **\.** – Email should contain the dot followed by some characters followed by the ‘@’ character
* **[a-z]{2,3}$** - It should contain two or three alphabetical characters at the end of the string. The ‘$’ represents the end of the string.

Examples

In the example below, when the user clicks the button, it will invoke the validateEmail() function. In the validateEmail() function, we take email input from the users using the prompt () method of JavaScript.

After that, we created the regular expression as explained in the above syntax. We have used the regex to test the user’s email input using the test() method, which returns the Boolean value based on whether the email matches the regular expression.

<html>

<body>

<h3>Using the <i> Regular expression </i> to validate email in JavaScript </h3>

<div id = "output"> </div>

<button onclick = "validateEmail()"> Validate any email </button>

<script>

var output = document.getElementById('output');

function validateEmail() {

let userEmail = prompt("Enter your email.", "you@gmail.com");

let regex = /^[a-z0-9]+@[a-z]+\.[a-z]{2,3}$/;

let result = regex.test(userEmail);

if (result) {

output.innerHTML = "The " + userEmail + " is a valid email address!";

} else {

output.innerHTML = "The " + userEmail + " is not a valid email address!";

}

}

</script>

</body>

</html>

**Regex Used in Example 2**

We have used the below regular expression pattern in example 2 to validate the email.

let regex = new RegExp(/\S+@\S+\.\S+/);

Users can follow the below explanation for the above regular expression.

* **\S+** - It represents any alphanumeric word.
* **\.** – It represents the dot character.

Basically, the above pattern matches the word@word.word kind of email address.

Example 2

In the example below, we have created the email input using HTML. Users can enter any email in the input field. After entering the email into the input, users need to click the ‘validate input email’ button, which will invoke the submitEmail() function.

In the submitEmail() function, We used the above regular expression with the test() method to check the user’s input email string.

In the output, users can enter various emails and observe the output.

<html>

<head>

<style>

div {

font-size: 1rem;

color: red;

margin: 0.1rem 1rem;

}

</style>

</head>

<body>

<h2>Using the <i> Regular expression </i> to validate email in JavaScript </h2>

<div id = "output"> </div>

<input type = "email" id = "emailInput" placeholder = "abc@gmail.com">

<br><br>

<button onclick = "submitEmail()"> Validate input email </button>

<script>

var output = document.getElementById('output');

function submitEmail() {

let userEmail = document.getElementById('emailInput').value;

let regex = new RegExp(/\S+@\S+\.\S+/);

let isValid = regex.test(userEmail);

if (isValid) {

output.innerHTML = "The " + userEmail + " is a valid email address!";

} else {

output.innerHTML = "The " + userEmail + " is not a valid email address!";

}

}

</script>

</body>

</html>







