



JavaScript Regex match()

Summary: in this tutorial, you'll learn about the JavaScript String `match()` method to match a string against a regular expression.

To understand how the `match()` method works and how to use it effectively, you should have the basic knowledge of [regular expression](#).

Introduction to the JavaScript match() method

The String `match()` method matches a [string](#) against a [regular expression](#):

```
str.match(regex);
```

If the `regex` is not a regular expression, the `match()` will convert it to a regular expression using the `RegExp()` constructor. The `match()` returns an [array](#) depending on whether the regular expression uses the global flag (`g`) flag or not:

- If the `regex` uses the `g` flag, then `match()` method returns an array containing all the matches. The result does not contain the capturing groups.
- If the `regex` doesn't use the `g` flag, the `match()` method will return the first match and its related capturing groups. The result of the match is the same result as `RegExp.exec()` with additional properties. See the example below for the details.

The `match()` returns `null` if it does not find any matches.

JavaScript Regex match() method

Let's take some examples of using the `match()` method.

1) Using the JavaScript regex match() method with the expression that has the global flag

The following example shows how to use the `match()` method with the global flag (`g`). It returns an array of matches:

```
let str = "Price: $5-$10";
let result = str.match(/\$\d+/g);
console.log(result);
```

Output:

```
["$5", "$10"]
```

In this example, the `match()` searches for any number that follows the `$` sign.

2) Using the JavaScript regex match() method with the expression that has the global flag

The following example illustrates how to use the `match()` method with a regular expression that doesn't have a global flag. It returns an array of the first match with additional properties.

```
let str = "Price: $5-$10";
let result = str.match(/\$\d+/);
console.log(result);
```

Output:

```
▼ ["$5", index: 7, input: "Price: $5-$10", groups: undefined] ⓘ
  0: "$5"
  groups: undefined
  index: 7
  input: "Price: $5-$10"
  length: 1
  ► __proto__: Array(0)
```

The additional properties are:

- `index` : is the index at which the match was found.
- `input` : a copy of the search string.
- `groups` : is the object of named capturing groups whose keys and values are the names and the capturing groups respectively. In this example, it is `undefined` because we did to define any named capturing groups.

3) Using the JavaScript Regex `match()` method with the named capturing group

The following shows how to use the `match()` method with a named capturing group. It captures the `"yellow"` into a group named `"color"` :

```
let str = 'I like yellow color palette!';

let re = /(?!<color>yellow) color/;
let result = str.match(re);

console.log(result);
```

Output:

```
▼ (2) ["yellow color", "yellow", index: 7,
  0: "yellow color"
  1: "yellow"
  ▼ groups:
    color: "yellow"
  index: 7
  input: "I like yellow color palette"
  length: 2
```

In this tutorial, you have learned how to use the JavaScript String `match()` method to match a string against a regular expression.