

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
<div><div></div>What is JavaScript</div>
<div><div></div>Getting started with JS</div>
<div><div></div>Variables</div>
<div><div></div>Data types</div>
<div><div></div>ASI</div>
<div><div></div>Strict mode</div>
<div><div></div>Iteration</div>
<div><div></div>Conditionals with Truthy / Falsey</div>
<div><div></div>Objects, Arrays + JSON</div>
<div><div></div>Structuring JS Code</div>
<div><div></div>Destructuring</div>
<div><div></div>Scope</div>
<div><div></div>Functions, function expressions and arrow functions</div>
<div><div></div>The ECMAScript 6 Specification</div>
<div><div></div>OOP in JavaScript</div>
<div><div></div>Best Practices</div>
<div><div></div>Closures</div>
<div><div></div>Callbacks and Promises</div>
<div><div></div>Cookies</div>
<div><div></div>Hoisting</div>
<div><div></div>Prototypes</div>
<div><div></div>Query Parameters</div>
<div><div></div>Higher Order Functions</div>

# Query Parameters

## Contents

- [Overview](#)
- [Tutorial](#)
- [Exercises](#)

## Overview

Query Parameters refer to URL-based parameters which are appended to the end of a URL address. In JavaScript, these carry significant meaning.

## Tutorial

The **URL Search Parameters** `URLSearchParams` JavaScript interface defines utility methods to work with the query string of a URL:

```
https://test.com/?name=Chris
https://test.com/hello?name=Chris
```

In this case, we have a single query parameter named `name`, with the value `Chris`.

You can have multiple parameters like this:

```
https://test.com/hello?name=Chris&age=20
```

The parameters passed as a query string are normally used server-side, to generate a proper response.

To access the value of a query inside the browser, using JavaScript, we have a special API called `URLSearchParams`, supported by all modern browsers:

```
const params = new URLSearchParams(window.location.search);
```

(note: don't pass the full URL as a parameter to `URLSearchParams()`, but only the query string part of the URL, which is accessed using `window.Location.search`.)

In case of the above example: `window.location.search` is equal to the string `?name=Chris`.

Now that you have the `params` object you can query it.

Check if a parameter was passed:

```
params.has('test');
```

Get the value of a parameter:

```
params.get('test');
```

Iterate over all the parameters using `for...of`:

```
const params = new URLSearchParams(window.location.search)
for (const param of params) {
  console.log(param)
}
```

A parameter can have more than one value; in this case, we pass the same parameter name multiple times like this:

