

A JavaScript redirect uses the programming language JavaScript (JS) to send users from one URL to another.

You can use a JS redirect to redirect users to a confirmation page after they submit a form or to provide fallback redirection if a page gets deleted, for example.

But they are not generally not recommended for SEO because search engines can have difficulty crawling and indexing sites that rely on them.

In this article, we'll discuss creating a redirect with JS and suggest alternatives.

How to Redirect with JavaScript

There are three main ways to redirect to another URL with [JavaScript](#):

1. **`window.location.href`**
2. **`location.assign()`**
3. **`location.replace()`**

Set a New `window.location.href` Property

You can use the **`window.location.href`** property in JavaScript to get or set the URL of the current webpage.

The use of "`window.location.href`" for redirection simulates the action of clicking on a hyperlink. It adds a new entry to the browser's session history. This allows users to use the "back" button to return to the previous page.

To redirect a user to a different URL, you can assign a new URL to this property.

The syntax is:

```
window.location.href = 'https://exampleURL.com/';
```

When tied to a user interaction-triggered change, such as a button click, this would redirect the user to “https://exampleURL.com/”

How?

You need to add the code inside of **<script>** tags inside of the **<head>** of your webpage and add an “**event handler**” to a button.

Here’s how you could do that:

```
<html>
<head>
<script>
function myFunction() {
window.location.href = "https://exampleURL.com/";
}
</script>
</head>
<body>
<h2>Redirect to a Webpage</h2>
<p>The location.href method redirects the user to a new webpage:</p>
<button onclick="myFunction()">Redirect</button>
</body>
</html>
```

JavaScript Validation API

Constraint Validation DOM Methods

Property	Description
checkValidity()	Returns true if an input element contains valid data.
setCustomValidity()	Sets the validationMessage property of an input element.

If an input field contains invalid data, display a message:

The checkValidity() Method

```
<input id="id1" type="number" min="100" max="300" required>
<button onclick="myFunction()">OK</button>

<p id="demo"></p>

<script>
function myFunction() {
  const inpObj = document.getElementById("id1");
  if (!inpObj.checkValidity()) {
    document.getElementById("demo").innerHTML = inpObj.validationMessage;
  }
}
</script>
```

