

WDD 231

[Home](#)[W1](#)[W2](#)[W3](#)[W4](#)[W5](#)[W6](#)[W7](#)

Advanced HTML Forms

Overview

The focus of this learning activity is to capture and display HTML **form** data on a confirmation page using URL query parameters. **URLSearchParams** is a built-in JavaScript object that provides utility methods to easily access, manipulate, and iterate over the parameters in the URL query string.

You have already used URL query parameters many times and may not have known it as a user. For example, when you use **google.com** to search for something, you enter your search term (such as "Javascript Help") into the form input text field and select the **Google Search**. The resulting URL on the next page will look like this: **<https://www.google.com/search?q=JavaScript+Help>**. The **q** is the **name** of the parameter and **JavaScript+Help** is the **value** of the parameter.

Prepare

In previous learning activities, you have learned about the purpose of HTML forms and when to use the different form elements such as various **input** types (text, password, checkbox, date, etc.) to collect information and interact with the user.

The key HTML form concepts include:

- HTML forms are used to collect user input and submit it to a server for processing.
- There are over 30 form elements related to form input, layout, and behavior including text fields, labels, radio buttons, checkboxes, dropdowns, and buttons.
- Form data can be sent using different methods (**GET** or **POST**) depending on the requirements.
- Form elements can be grouped using the **<fieldset>** and **<legend>** elements.
- Form validation can be done using HTML attributes (e.g., **required**, **pattern**) or JavaScript.

In this learning activity, you will learn how to use JavaScript to capture the information entered into the form and display it on a confirmation page using the

URLSearchParams object to parse the URL and extract the query string parameters.

The key concepts are:

- JavaScript can be used to manipulate form data and handle form submissions dynamically.
- **URLSearchParams** is a built-in JavaScript object that provides utility methods for working with query string parameters.

URLSearchParams

The **URLSearchParams** interface provides utility methods to work with the query string of a URL. It allows you to easily access, manipulate, and iterate over the parameters in the query string.

For example, if you have a URL like this: **https://example.com/?name=John&age=30**

You can use **URLSearchParams** to extract the parameters:

```
const params = new URLSearchParams(window.location.search);  
const name = params.get('name'); // "John"  
const age = params.get('age'); // "30"
```

In this example, **window.location.search** returns the query string part of the URL (e.g., **?name=John&age=30**). The **URLSearchParams** constructor takes this string and creates an object that allows you to access the individual parameters using the **.get()** method.

Additionally, you can use methods like **.has()**, **.set()**, and **.delete()** to check for the existence of parameters, update their values, or remove them from the query string.

For example, you can check if a parameter exists:

```
if (params.has('name')) {  
  console.log('Name parameter exists!');  
}
```

You can also update a parameter's value:

```
params.set('age', 31); // Update age to 31
```

And you can remove a parameter:

```
params.delete('name'); // Remove name parameter
```

Finally, you can convert the `URLSearchParams` object back to a query string using the `.toString()` method:

```
const queryString = params.toString(); // "age=31"
```

This allows you to easily manipulate the query string and update the URL without having to manually construct it.

Activity Instructions

In this activity, you will work with a form that has several different form elements. You will then create a page to display the form information entered by the user. You will use the `URLSearchParams` object to parse the URL and extract the query string parameters and display them on the confirmation page.

These activities are designed so that you can follow along and learn. Pause the video instruction as needed and code your own version.

Step 1: Download Starting Files

Download the [starting file](#) to your computer's desktop

Step 2: Overview of Form Methods

Watch and follow along: [Form Data Part 1](#)

Project Overview

Step 3: Capturing Form Data

Watch and follow along: [Form Data Part 2](#)

A demonstration capturing information from `URLSearchParams`

Test and Share

Test your updates and share your work with your peers on Microsoft Teams.

Optional Resources

Review [Learn Forms](#) – MDN

Read: [form methods](#) for GET and POST

Read: [URLSearchParams](#)

Back

Copyright © Brigham Young University-Idaho | All rights reserved