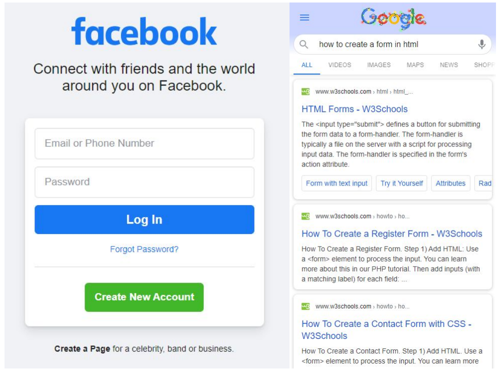
Web Forms

HTML5

What are Web Forms?

When a web page is expected to accept input from the user, the traditional way to do this is via a form in the web page. If you frequent the internet, you've likely come across all sorts of forms in your travels—some of the most common including:

- sign up
- sign in
- enter a post
- upload a photo
- fill out your bio and profile
- enter a search



Connect with friends and the world around you on Facebook.

Create a Page for a celebrity, band or business.

www.w3schools.com › html › html_...

HTML Forms - W3Schools

The <input type="submit"> defines a button for submitting the form data to a form-handler. The form-handler is typically a file on the server with a script for processing input data. The form-handler is specified in the form's action attribute.

www.w3schools.com › howto › ho...

How To Create a Register Form - W3Schools

How To Create a Register Form. Step 1) Add HTML: Use a <form> element to process the input. You can learn more about this in our PHP tutorial. Then add inputs (with a matching label) for each field: ...

www.w3schools.com › howto › ho...

How To Create a Contact Form with CSS - W3Schools

How To Create a Contact Form. Step 1) Add HTML. Use a <form> element to process the input. You can learn more

Any standard web form is composed of a form element and its contents. Before we dive into the elements that belong inside of a form, we should consider some important attributes that the form element is afforded to give us necessary flexibility.

Action Attribute

A form’s action attribute tells the form where to deliver its data to go when you submit.

Much like an image or anchor can use relative or absolute paths, the action attribute can utilize either as well.

This becomes incredibly useful once you begin developing a back-end for your website, as it is the back-end that can read, understand, and store submitted data. Because an action can use an absolute path, you can also build a form that submits to a different website or web server.

If you do not fill in a value for the action attribute, the form will submit to the current page. Upon submit you’ll see the page reload.

Method Attribute

Along with where to send, via the action attribute, we can also let the form know how to send the data.

This is where the method attribute comes in. There are two common values we can enter for this attribute:

1. GET
   This type of request will append the form data to the URL. This is incredibly useful if you'd like for users to be able to share a link with their submission. A great example would be a Google search—the search term, after submitting, appears in the address bar so that you may send the same search to a friend.

2. POST
   This type of request will append the form data to the HTTP request itself—meaning as a user, you won't be able to easily see or share it. This offers a bit more privacy and won't allow users to share their form responses easily.

   POST requests are often used for forms involving one-time submission, or sensitive information. Some examples include payment forms, sign ins, and sign ups.

Always take a moment to decide which method makes the most sense for the form you're building!

Inputs

The default input element is of the text type. You can change the type via the type attribute. It is important to note that there are a wide array of types at your disposal.

When building web forms it is important to consider the strengths, weaknesses, and overall function of each type of input and which one would help collect data in an efficient and user-friendly way.

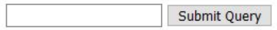
Keep a list of input types handy and become familiar with the options available to you.

An essential type, aside from text, is submit.

An input with the submit type will display as a button, capable of submitting the form on click.

A form with a single text input might look like so…

```
<form action=”#” method=”GET”>
   <input type=”text” name=”my-text”>
   <input type=”submit”>
</form>
```



Note that a descriptive name attribute should always be added. The name will assist in labelling the data when submitting the form.

Labels

The label element allows for us to properly caption a form input field.

Labels are incredibly important in the modern web:

- If you click or tap on a label associated with a form input field, that field will become focused (highlighted so that you may freely type or manipulate the value of the field.)
- Accessibility tools—like screen readers—rely on labels to let the user know what sort of information a form input field requires.

When adding any label, it is best to associate it with a form input field.

There are two ways of associating a label with a form input field:

1. The input you'd like to associate can simply be placed inside of its corresponding label tags. This is considered implicit.

   ```
   <label>Enter your name:
     <input type="text" name="username">
   </label>
   ```

2. A "for" attribute can be added to the label. Its value must match the "id" value for the input. This is considered explicit.

   ```
   <label for="your-name">Enter your name:</label>
   <input id="your-name" type="text" name="username">
   ```

Common Field Controls (Input Types)

An input element's type attribute determines which form control should display in its place. There is an array of choices available, so it is good to familiarize yourself with the most common and popular of the lot:

- Text
- Password
- Hidden
- Checkbox
- Radio Button (radio)
- Button (submit; reset; button)
- Image Button (image)
- E-mail Address (email)
- Phone Numbers (tel)
- Search
- URL
- Numeric (number)
- Slider
- Date and/or Time
- Colour Picker (color)
- File

Other Field Controls

There are a few form fields that, surprisingly, do not make use of the input element. They, instead, have their own elements that determine their consistent field control output and behaviours.

Make note of the following elements that are also used inside of a web form context:

- Large Text Field (textarea)
- Dropdown (select and option)
- Autocomplete / List Match (datalist and option)
- Progress and Metre Bars (progress and meter)

Adding Structure to Forms

If you find a form may end up seeming rather disorganized it can be divided into sections to bring some order to it.

In forms, these sections are called fieldsets. Each fieldset can be labeled with a legend element.

Working with fieldsets and legends are covered in detail on the MDN website.

This type of labelling is often pertinent for the sake of grouping checkboxes or other tightly-knit element groupings.

Learning More About Forms

There is plenty more to read and learn about web forms. Try here to get started!