



HTML Forms



Forms

Forms allow users to **enter data**. This data is often sent to a **web server** to be **processed** and **stored**, or to **update** the **interface** (e.g. showing a profile page after logging in).

Forms are made up of different **widgets**. These could be:

- Input fields (text, password, email, etc)
- Checkboxes
- Radio buttons (aka “toggles”)
- Sliders
- Date pickers

... or many more!

👉 Creating a Form

Get in touch!

Name

Email

Message

Send!

Creating a form

Open the `forms-starter` directory in VS Code and open `index.html` in a Web Browser. In the body you'll see the following:

```
<body>
  <header>
    <h1>Get in touch!</h1>
  </header>
  <form action="" method="">
    <!-- your code here -->
  </form>
</body>
```

We'll look at the `action` and `method` attributes soon, but first, let's create our form!

👉 Creating a form

Let's start with the name input. There are two elements here, the label, and the input so we'll wrap them in a div to group them:

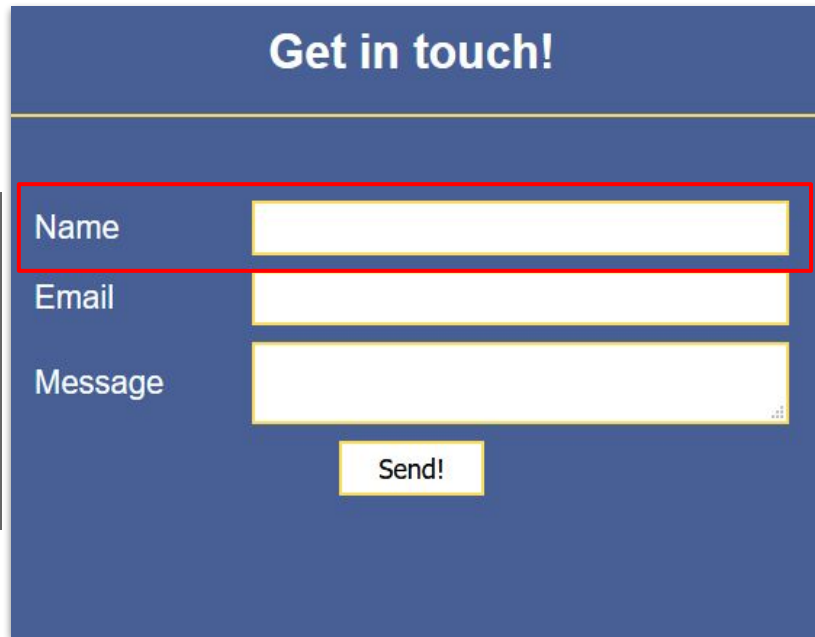
```
<form action="" method="">
  <div class="form-item">
    </div>
</form>
```

The image shows a contact form titled "Get in touch!" on a dark blue background. The form contains three input fields: "Name", "Email", and "Message". The "Name" field is highlighted with a red border. Below the input fields is a "Send!" button. The form is styled with a dark blue background and white text for labels and the button.

👉 Creating a form

Then we'll add the two elements:

```
<form action="" method="">
  <div class="form-item">
    <label for="name">Name</label>
    <input type="text" id="name" name="name">
  </div>
</form>
```



Get in touch!

Name

Email

Message

Send!

✓ Refresh your browser and see what changed!

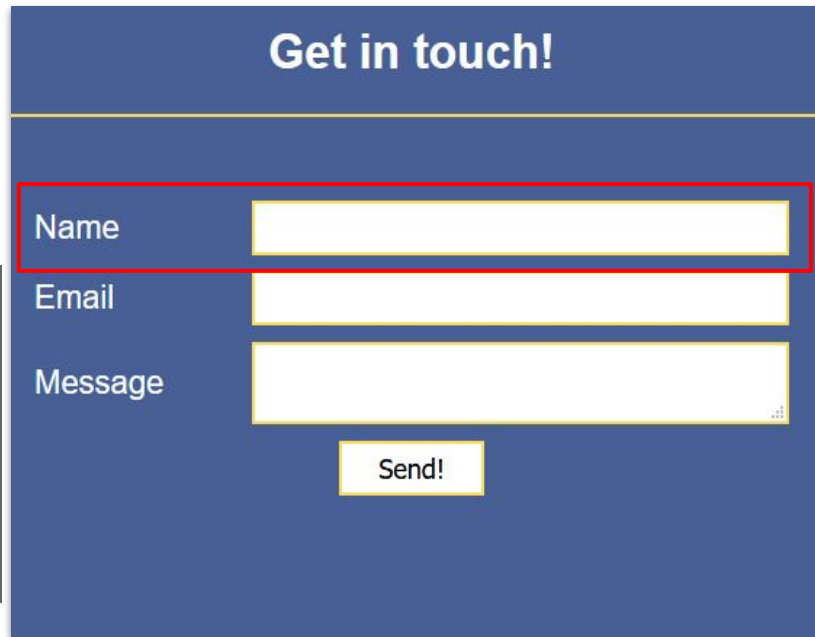
👉 Creating a form

Did you notice the new attributes `for` and `name`?

```
<form action="" method="">
  <div class="form-item">
    <label for="name">Name</label>
    <input type="text" id="name" name="name">
  </div>
</form>
```

`for` matches the widget's `id`.

The data from each widget is paired with the `name` when it is sent to the server.



Get in touch!

Name

Email

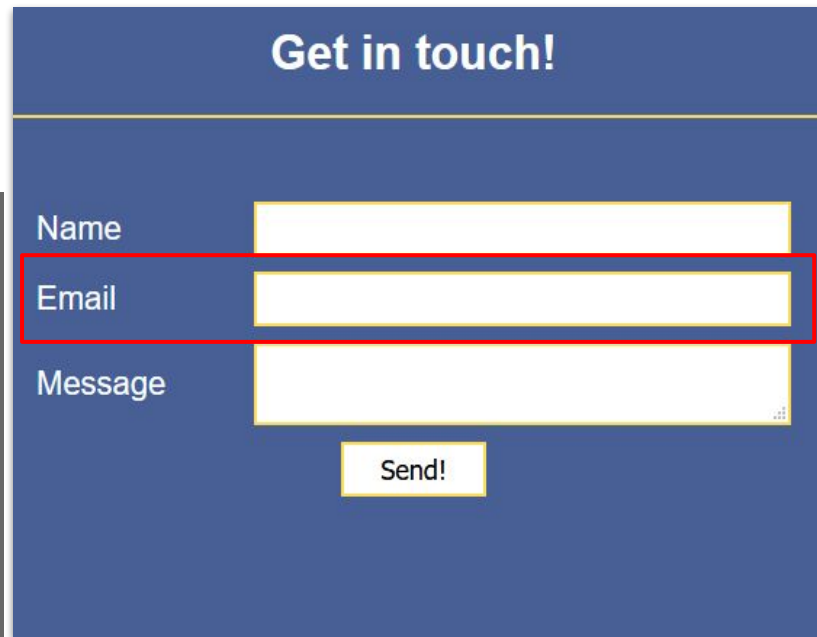
Message

Send!

👉 Creating a form

Next we'll add the email widget:

```
<form action="" method="">
  <div class="form-item">
    <label for="name">Name</label>
    <input type="text" id="name" name="name">
  </div>
  <div class="form-item">
    <label for="mail">Email</label>
    <input type="email" id="mail" name="email">
  </div>
</form>
```



The image shows a web form titled "Get in touch!" on a blue background. It contains three input fields: "Name", "Email", and "Message". The "Email" field is highlighted with a red rectangular border. Below the fields is a "Send!" button.


Notice this time set the input **type** to **email**.

✅ Refresh your browser and see what changed!

👉 Creating a form

We'll use a textarea widget for the message:

```
<form action="" method="">
  <div class="form-item">
    <label for="name">Name</label>
    <input type="text" id="name" name="name">
  </div>
  <div class="form-item">
    <label for="mail">Email</label>
    <input type="email" id="mail" name="email">
  </div>
  <div class="form-item">
    <label for="message">Message</label>
    <textarea id="message" name="message"></textarea>
  </div>
</form>
```



The image shows a web form with a dark blue header containing the text "Get in touch!". Below the header, there are three input fields on a lighter blue background. The first field is labeled "Name" and is a text input. The second field is labeled "Email" and is an email input. The third field is labeled "Message" and is a text area; this field and its label are highlighted with a red rectangular border. Below these fields is a yellow button with the text "Send!".



Refresh your browser and see what changed!

👉 Creating a form

Finally, we'll add the submit button:

```
<form action="" method="">
  <div class="form-item">
    <label for="name">Name</label>
    <input type="text" id="name" name="name">
  </div>
  <div class="form-item">
    <label for="mail">Email</label>
    <input type="email" id="mail" name="email">
  </div>
  <div class="form-item">
    <label for="message">Message</label>
    <textarea id="message" name="message"></textarea>
  </div>
  <div class="form-item">
    <button type="submit">Send!</button>
  </div>
</form>
```

A screenshot of a web form titled "Get in touch!" in white text on a dark blue background. The form contains three input fields: "Name" (a text input), "Email" (an email input), and "Message" (a text area). Below these fields is a "Send!" button, which is highlighted with a red rectangular border. The entire form is set against a dark blue background with a thin white horizontal line separating the title from the input fields.

✓ Refresh your browser and see what changed!

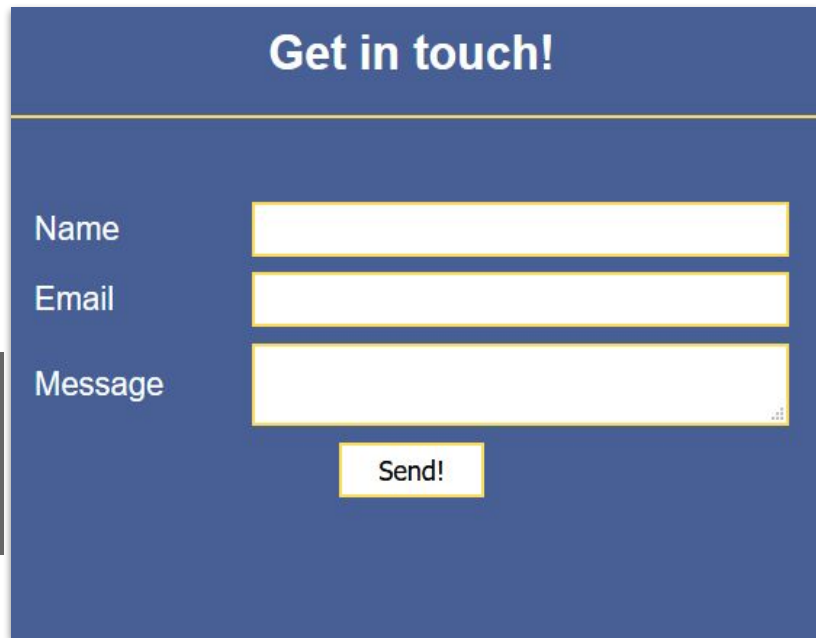
👉 Creating a form

Let's revisit the `action` and `method` attributes of the form.

`method` defines which HTTP method to use when the form is submitted. We'll be using POST to send the data from the form to the server:

```
<form action="" method="post">  
  <!-- your form here -->  
</form>
```

`action` defines the URL when the form's data will be sent.



The image shows a contact form with a dark blue header and background. The header contains the text "Get in touch!" in white. Below the header, there are three input fields with yellow borders, each preceded by a label: "Name", "Email", and "Message". At the bottom right of the form area is a yellow button with the text "Send!" in black.

👉 Making our form functional

At the moment our contact forms don't do anything. What we want them to do is to send us an email with the form content when someone clicks send.

We're going to use some third party software to do this for us.

Head over to <https://formsfree.io/> and sign up for an account.

The email address you use to sign up is where messages from your contact form will be sent to.



👉 Adding Functionality to the Contact Form

Formspree will give you your very own URL to submit your form to.

Point your form at that URL, and make sure it is using "POST" for the method.

```
<form action="https://formspree.io/your-formspree-code" method="post">  
  <!-- your form here -->  
</form>
```

✅ Does the form work?... 🤔

Testing the form

Since we have are opening the file directly from our computer's file system, it won't have the `https://` in the url, so the browser won't treat the file as a normal web page. This means we can't do fancy things like send forms just yet.

Fortunately, we can use Formspree's testing area to test our forms:

<https://test.formspree.io/>

Once the site is deployed, we will be able to send messages through the form!