HTML Forms Lab

- Due No Due Date
- Points 1
- Submitting a website url

[(https://github.com/learn-co-curriculum/phase-1-html-forms-lab)](https://github.com/learn-co-curriculum/phase-1-html-forms-lab) [(https://github.com/learn-co-curriculum/phase-1-html-forms-lab/issues/new)](https://github.com/learn-co-curriculum/phase-1-html-forms-lab/issues/new)

Learning Goals

- Apply what we know of HTML forms to build a functional contact form

Introduction

Business of all types and sizes want to provide multiple ways for their potential customers to contact them. One of the most common ways to do this is on a website is to provide a *contact form*.

In this lab, we will practice what we've learned about HTML forms by creating a contact form similar to those we see on business websites.

Reinforcing Our Understanding of HTML Forms

To see and test your form in action, run `httpserver` or open `index.html` in a new browser tab. To test your work, run `npm test` .

Some basic HTML has been provided for this lab, along with HTML comments on what is needed. Follow the provided comments and test messages to see what is needed when creating the form. At completion, your form should have:

- A *required* text input for a full name with a placeholder, "Enter Name"
- A *required* email type input for an email address with a placeholder, "Enter Email"
- A *tel* type input for a telephone number with a placeholder, "Enter Telephone (optional)"
- A text area for a user to include a message with a placeholder, "Enter Message"
- Labels for each text input
- A checkbox with text of your choosing
- A submit button

The `required` Attribute

The `required` attribute prevents the browser from submitting a form. All required inputs must be filled in properly, preventing an unneeded page reload. This functionality is baked into HTML5 for us automatically. All that is needed is to include the word `required` as an attribute, without setting it to any value.

When an input is required, the `type` attribute (e.g., `text` vs `email`) will determine what the browser checks. In the case of an email, for instance, the input must match the format of an email address, including the "@".

Conclusion

Forms are an essential way for users to submit data to a website. Some forms, like search bars, only contain a single input field, but we can make them as complex as we need.

We are only focused on building out the front end side of this form. When a form is submitted, it is up to the *backend* of a website to decide how to handle and store form data when submitted.