# 2.4 Mandatory Pairing: Forms

Submit Assignment

**Due** Monday by 10am **Points** 20 **Submitting** a website url

If your browser is a translator delivering messages to you that were sent from distant computers, HTML forms are your way to respond. Forms allow your browser to ask you for information, then pass that information along to your banking website, Facebook, Instagram, or whoever originally requested the information.

When you update your status on social media, log in to a website, or send an email, you are filling out a form that your browser will submit, via an HTTP request, to another computer.

## Release 0: Submit a Form

TRY THE THING

**httpbin** (https://httpbin.org) allows you to fill out an example HTML form, then view the details of the message that a computer would receive if it were processing your form.

1. Fill out the httpbin example **form** (https://httpbin.org/forms/post) with some example data. Use data that actually makes sense, or the next step might be difficult.
2. Submit the form, then spend some time going through the odd-looking result. Anytime you fill out a form, your browser translates the information to something along these lines. It's meant to be read by computers, not humans, which is why it looks a bit strange.

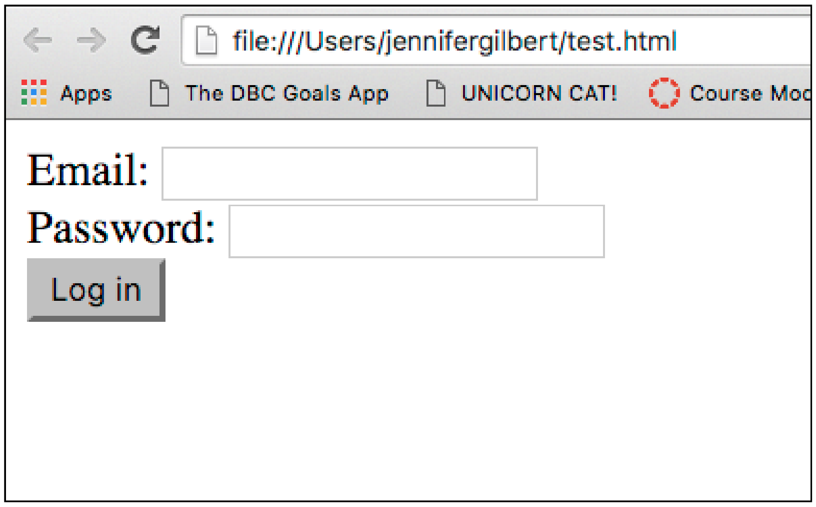
## Release 1: Identify Form Tags

Many different HTML elements are designed for data entry. In this release, you'll explore and discuss many of them with your pair.

Here's the code for a simple form:

```
<form action="login" method="post">
  <div>
    <label>Email:</label>
    <input type="email" name="email">
  </div>
  <div>
    <label>Password:</label>
    <input type="password" name="password">
  </div>
  <button type="submit">Log in</button>
</form>
```

And here's how that form would render in the browser (enlarged for easy viewing):



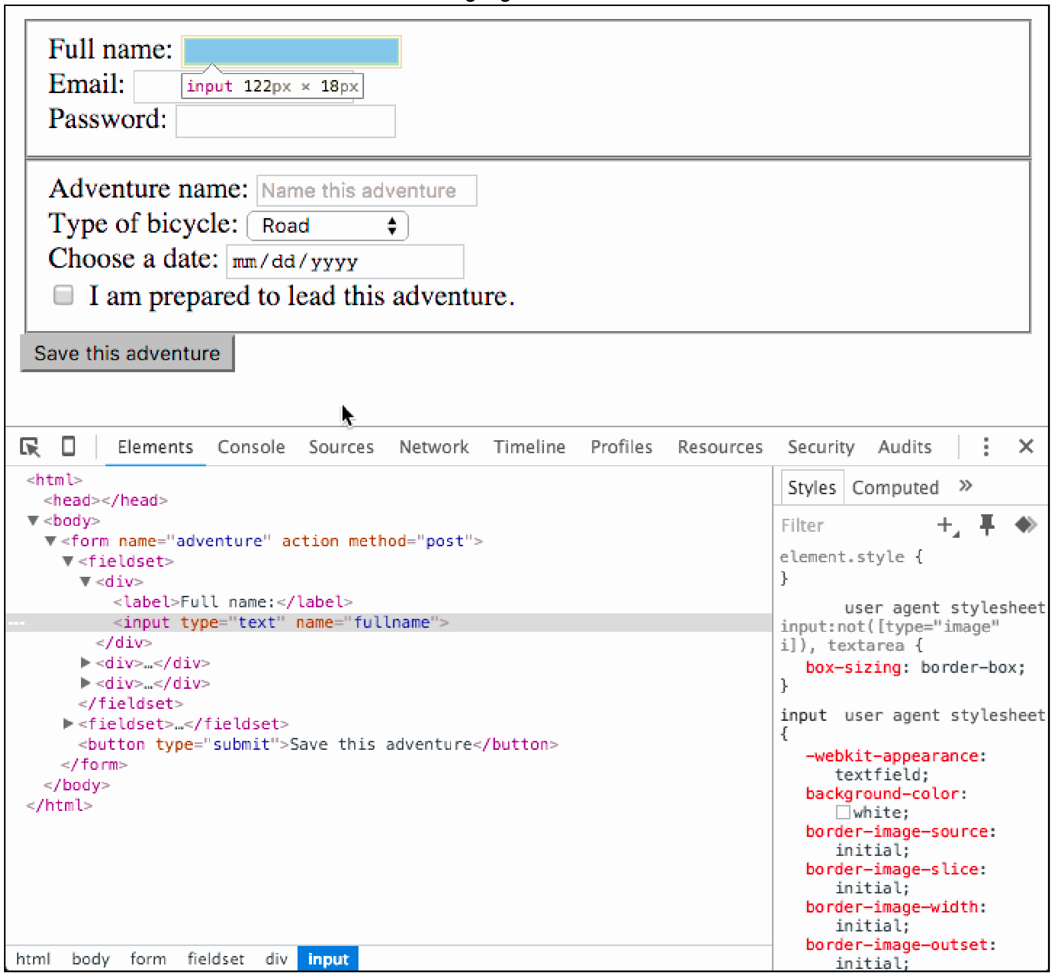
Every form is wrapped in a `form` tag. The `action` attribute of a form tag identifies which page the browser should load after the form is submitted. For now, you can leave the `method` attribute of a form tag as "POST," as that's the most common one and HTTP methods are beyond the scope of this challenge.

Notice that the `input` tag can have different types. For instance, the "password" type will obscure the letters being typed in, turning them into dots or asterisks for privacy.

## DO THE THING

The best way to learn about how forms work is to simply play with one, so we've provided a fun example in `phase-0-tracks/html/forms`.

1. Open `example-form.html` in your browser. No need to fill it out yet; for now, it doesn't do anything when you submit, but we'll fix that in a moment.
2. In Chrome, right-click on one of the inputs and choose **Inspect**. You should get a panel that allows you to explore the page's source HTML, as in the GIF below. Notice how the browser highlights various elements in blue as the mouse arrow rolls over them.



```
<html>
  <head></head>
  ▼<body>
    ▼<form name="adventure" action method="post">
      ▼<fieldset>
        ▼<div>
            <label>Full name:</label>
            <input type="text" name="fullname">
          </div>
        ▶<div>…</div>
        ▶<div>…</div>
        </fieldset>
      ▶<fieldset>…</fieldset>
        <button type="submit">Save this adventure</button>
      </form>
    </body>
</html>
```

```
element.style {
}
      user agent stylesheet
input:not([type="image"
i]), textarea {
  box-sizing: border-box;
}
input  user agent stylesheet
{
  -webkit-appearance:
     textfield;
  background-color:
     □white;
  border-image-source:
     initial;
  border-image-slice:
     initial;
  border-image-width:
     initial;
  border-image-outset:
     initial;
```

3. In the `forms` folder, create a file called `input_elements.md`. Explore the form and make a quick list of the different input elements and form tags you come across, and their purpose.
4. Let's give the form an action, so it will submit. You can submit to httpbin's form tester and see what data our form is successfully submitting to httpbin's server. Update the form's `action` attribute to `https://httpbin.org/post`.
5. Fill the form in with meaningful data and submit the form to verify that you're successfully sending the data.
6. When you look at the httpbin data report, you will see that all of the data you filled in has a label. There's the "bike" field, the "email" field, and so on. Where do those labels come from? Look at the HTML to figure it out, and update some of those labels and resubmit the form to test your theory.

## Release 2: Build a Form

We don't know how to build a server yet, so unless we submit to httpbin, we won't be able to use any data we collect in our forms. However, we can still get comfortable with the HTML and CSS needed to create a form.

DO THE THING

1. Create a new file in the `forms` folder. Call it `practice-form.html`.
2. Add the base HTML layout elements to the practice form.
3. Create a simple `success.html` page in the same folder with the base HTML layout elements and some text like "Thanks for submitting your form."
4. Add a form to the practice page. The content of your form can be anything you like, but it must contain the following:
   1. An `action="success.html"` attribute in the form tag that will display the `success.html` page when the form is submitted.
   2. A `select` with at least three choices.
   3. A `textarea` input, which you may have to look up in the **form documentation** (https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms) to use.
   4. Three other inputs of varying types.
   5. Labels and `name` attributes for all of the form inputs.
   6. A submit button that triggers the form action and displays the `success.html` page.
5. Validate your HTML code with an **HTML validator** (https://validator.w3.org/#validate_by_input).
6. Commit your code, push up your changes, and submit the GitHub URL of `practice-form.html` to complete this challenge.

## Release 3: Give Feedback to Your Pair

DO THE THING

Take a moment to give each other actionable, specific, and kind feedback on your pairing session. An example of actionable, specific, and kind feedback might be, "Sometimes we went faster than I was comfortable with because you were on a roll. It might help to pause here and there to make sure your pair is still with you. But you were really encouraging, and I appreciated your positive attitude."

**Some Rubric (4)**

| Criteria | Ratings | | | Pts |
|---|---|---|---|---|
| create a submit input element<br>threshold: 3.0 pts | Exceeds Expectations<br>5.0 pts | Meets Expectations<br>3.0 pts | Does Not Meet Expectations<br>0.0 pts | 5.0 pts |
| create a form with several inputs, including select and textarea | Exceeds Expectations<br>5.0 pts | Meets Expectations<br>3.0 pts | Does Not Meet Expectations<br>0.0 pts | 5.0 pts |
| set the action of a form to another html page | Exceeds Expectations<br>5.0 pts | Meets Expectations<br>3.0 pts | Does Not Meet Expectations<br>0.0 pts | 5.0 pts |
| demonstrate good Git workflow | Exceeds Expectations<br>5.0 pts | Meets Expectations<br>3.0 pts | Does Not Meet Expectations<br>0.0 pts | 5.0 pts |
| | | | | Total Points: 20.0 |