Forms

After completing this section, you will:

- [ ] Be able to use forms in your HTML5 documents.
- [ ] Know a variety of input types and their applications.
- [ ] Know how to use checkboxes and radio buttons.
- [ ] Be able to add dropdowns in your forms.

Introduction

HTML5 forms are an essential component of most web pages. Often, developers and designers spend a great deal of time working on the home page of their websites. However, they might be better off if they spent that time working on the forms. The form is the point at which users “convert”-- where they decide whether or not to provide their personal information, make a purchase, sign up or move on to another site. Creating forms that users want to fill out is a critical skill that could be a course in and of itself!

Form Tag

As usual, we’ll be starting with our basic document structure.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Forms</title>
  </head>
  <body>

  </body>
</html>
```

We’ll add a `form` tag to declare our form in the document. There are two important attributes for the form tag: `action` and `method`.

```
  <body>
    <form action="#" method="post">

    </form>
  </body>
```

The `action` attribute is used to specify where to send the form data once the form is submitted. The action target is typically a back-end script written in a server-side language addressable via a URL. However, since we're not covering back-end languages in this course, we add a `#` sign.

The `method` attribute is related to the request type of the form. The most common two are `get` and `post` requests. Forms with the `get` method append the data submitted through them to the URL of the `action` attribute. When the form passes to the server, the URL is visible in the browser's address bar.

> The Get Method places the form data in a query string that looks like this: `index.html?userName=Lassoff&age=44&weight=250`. You can see data visible in the URL and, as you can imagine, this is far from a secure method of transmitting data. You would never use the Get method with data like social security or credit card numbers.

The `post` method is used more frequently. With `post`, the form values are sent to the processor as part of the HTTP header and not visible in the browser.

Text Input

Most forms request the user the user enter their name before anything else. We'll add an `input` tag inside our form and set its `type` attribute to `text`. The `input` tag and `text` type display a field inside the browser that accepts `text` input inside it.

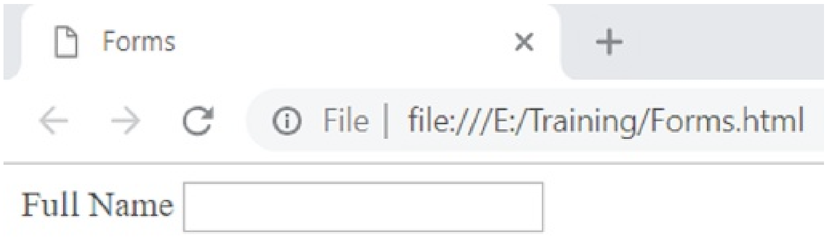
We then add two more attributes to the `input` tag; `id` and `name`.

We need the `id` attribute so that we can access what the user types in the form field using Javascript code. On the other hand, the `name` attribute is used to label the value of its `input` tag when it's sent to a back-end form processor. That way, we can differentiate the values of different pieces of information in the form.

We should also add a `label` tag to describe what the purpose of `input` field is. We can connect the `label` and `input` tags further by using a `for` attribute within the `label` tag and assigning the value of the `input`'s `id` to the `for` attribute.

The `label` tag and its `for` attribute are even more important for improving our form's accessibility to those with disabilities.

```
<form action="#" method="post">
  <label for="userName">Full Name</label>
  <input type="text" id="userName" name="userName" />
</form>
```



You'll notice that with the label form properly configured. You can click on the label or the form field to activate the form, making it more accessible to those with small motor disabilities.

HTML5 Text Inputs

Let's add another field for the user to enter their email address.

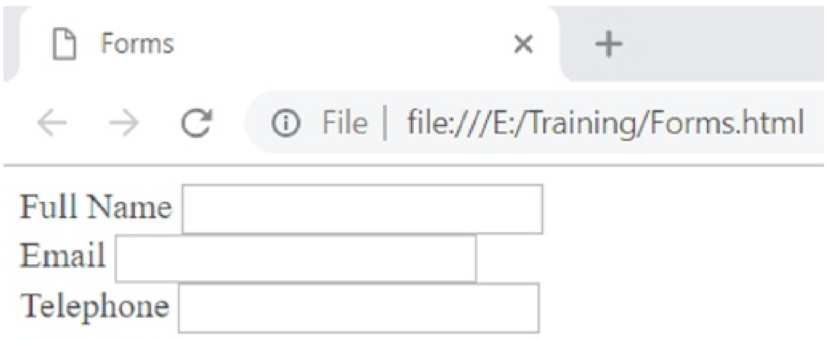
```
    <form action="#" method="post">
      <label for="userName">Full Name</label>
      <input type="text" id="userName" name="userName" /><br/>
      <label for="email">Email</label>
      <input type="email" id="email" name="email" />
    </form>
```

You'll notice that the value of the `type` attribute in the `input` tag is `email`. If you test on your laptop or desktop computer, you'll see no difference in the text field.

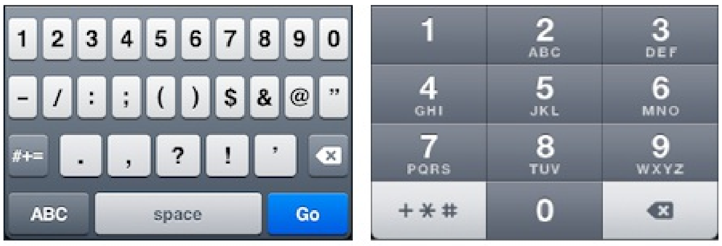
Using `email` affects the keyboard that shows up when the user types in an `input` field. Instead of the usual keyboard, the mobile email optimized keyboard appears, which has an `@` sign in the lower-right corner and allows the user to type email addresses faster.

Another piece of information we can request in our form is the telephone number.

```
    <form action="#" method="post">
      <label for="userName">Full Name</label>
      <input type="text" id="userName" name="userName" /><br/>
      <label for="email">Email</label>
      <input type="email" id="email" name="email" /><br/>
      <label for="phone">Telephone</label>
      <input type="tel" id="phone" name="phone" />
    </form>
```



This time we use `tel` for the value of the type attribute which provides a keyboard optimized for phone numbers on mobile devices. The phone number keyboard appears on the right in the image below.



Numeric keyboard (iPhone) Phone / Number pad (iPhone)

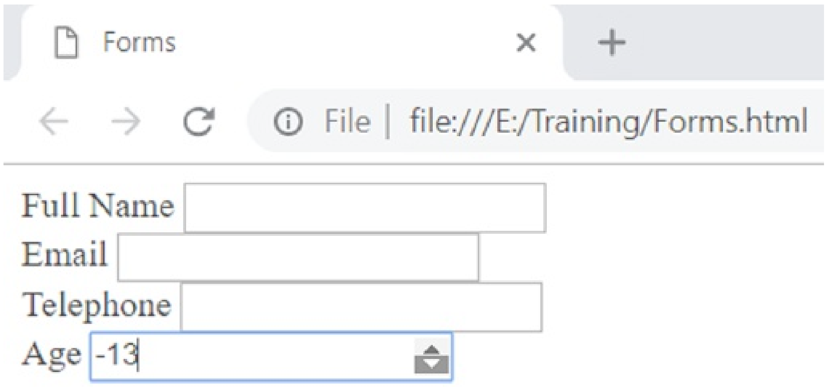
On the left is the keyboard optimized for numerical inputs.

Numerical Inputs

Let's ask for the user's age next.

```
<form>
  ...
  <label for="age">Age</label>
  <input type="number" id="age" name="age" />
</form>
```

The `number` type `input` gives our field an up and down arrow that we can use to increase or decrease the input value. By default, you can type any number in the input. When requesting age, a problem arises when the user decreases the number below zero, and it goes into negative territory.



Full Name
Email
Telephone
Age -13

Since people can't have a negative age, we can add the attributes `min` and `max` to the `input` tag. We'll assign the `min` attribute with the value of `0` and `max` value of 150.

```
<form>
  ...
  <label for="age">Age</label>
  <input type="number" id="age" name="age" min="0" max="150" />
</form>
```

Now, when we try to decrease our age below zero using the down arrow, the number won't change.

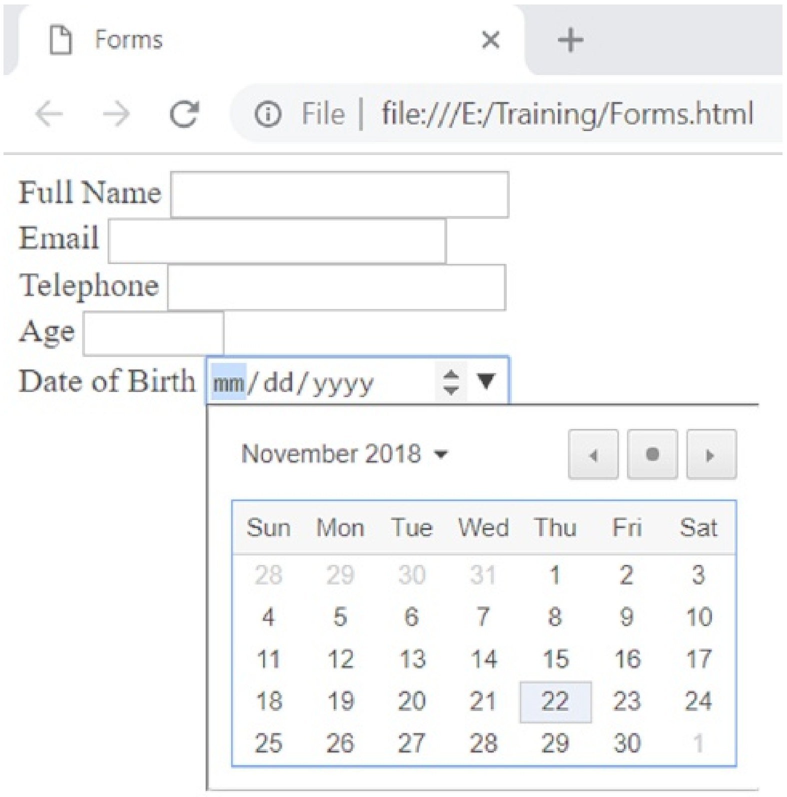
However, a user can still type in whatever number they want into the field.

> Client side limitations like this do not make a form more secure. Just by typing a value someone can introduce incorrect data into our form. Remember that for security comes from a combination of client-side deterrents (such as the `max` and `min` attributes and server-side security.

While the `min` and `max` attributes' solution isn't perfect, it still improves your form a little bit.

Let's continue and add a `date` field into our form. The `input` type will be `date` which will present the user with a calendar from which they can choose a date.

```
<form>
  ...
  <label for="dob">Date of Birth</label>
  <input type="date" id="dob" name="dob" />
</form>
```



> The calendar may appear slightly different on different browsers.

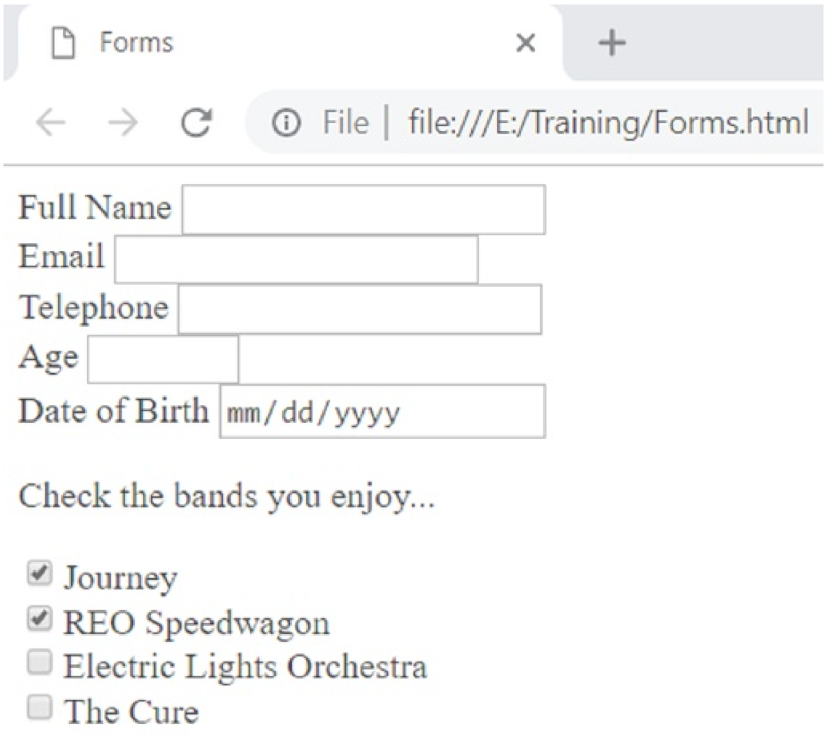
There are several other `input` types we can use, but these are the most common.

Checkboxes

Checkboxes are useful because of their simplicity and ease of use. They are useful for when you want the user to select one or more items from a group. Let’s present the user with a few band names from which they can choose their favorite(s).

```
    <form>
      ...
      <p>Check the bands you enjoy...</p>
      <input type="checkbox" name="bands" value="journey">Journey<br />
      <input type="checkbox" name="bands" value="reo">REO Speedwagon<br />
      <input type="checkbox" name="bands" value="elo">Electric Lights Orchestra<br />
      <input type="checkbox" name="bands" value="cure">The Cure<br />
     </form>
```

From the user interface in the browser, users can select one or more of those bands.



Full Name
Email
Telephone
Age
Date of Birth mm/dd/yyyy

Check the bands you enjoy...

✓ Journey
✓ REO Speedwagon
☐ Electric Lights Orchestra
☐ The Cure

Note that the name attribute in the code contains the same value for each of the checkboxes in the set. It is the same to ensure that the form processes correctly and that the set of checkboxes work as a team.

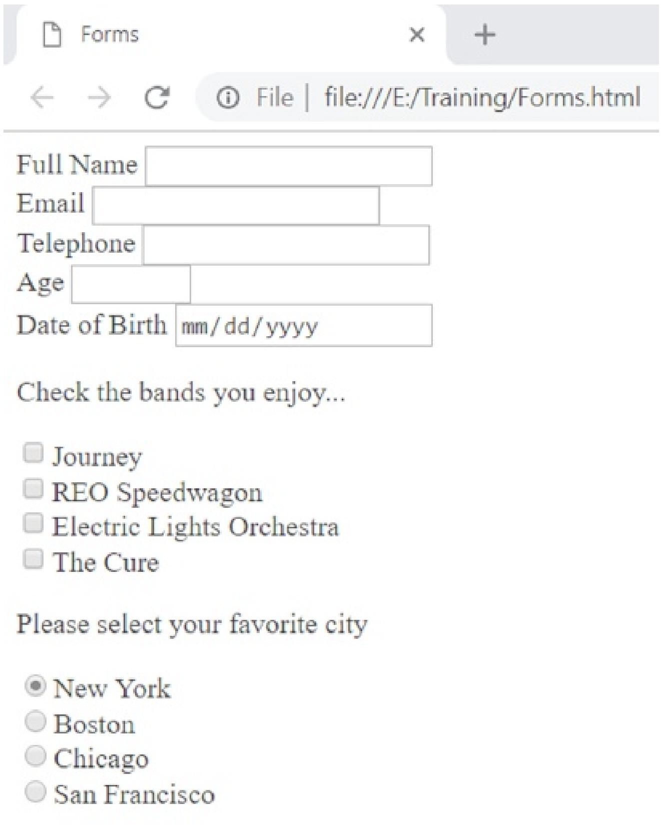
Radio Buttons

Radio buttons are designed for selecting only one option from a group, but the rest of the HTML code is very similar to the checkbox.

Let’s ask the user select his/her favorite US city.

```
<form>
  ...
  <p>Please select your favorite city</p>
  <input type="radio" name="city" value="NY">New York<br/>
  <input type="radio" name="city" value="Boston">Boston<br/>
  <input type="radio" name="city" value="Chicago">Chicago<br/>
  <input type="radio" name="city" value="SF">San Francisco<br/>
</form>
```

Again the `name` attribute has the same value. This is even more important with radio buttons since only one of the buttons can be selected from the set at any one time.



Full Name
Email
Telephone
Age
Date of Birth mm/dd/yyyy

Check the bands you enjoy...

☐ Journey
☐ REO Speedwagon
☐ Electric Lights Orchestra
☐ The Cure

Please select your favorite city

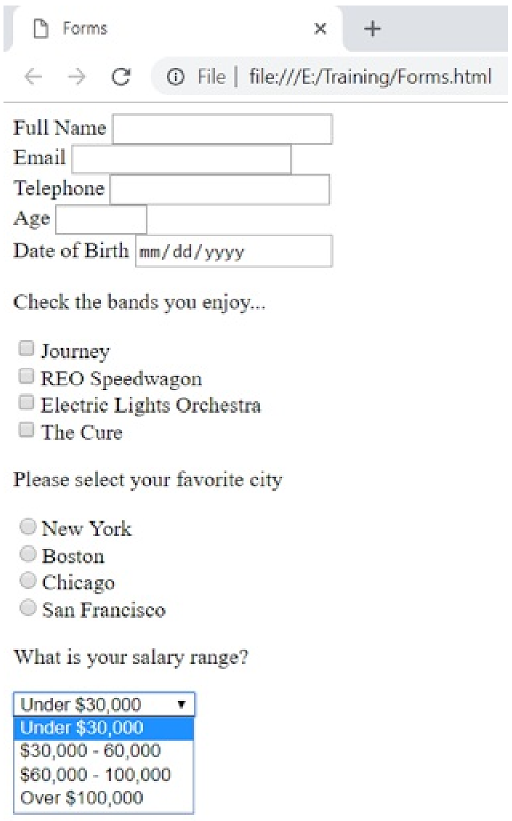
● New York
○ Boston
○ Chicago
○ San Francisco

Drop Downs

The last form item we'll learn about is the `select` tag. The `select` tag allows us to create drop-down menus. You'll commonly see these when having to select your country or state on a form.

In our form, we'll ask the user to select from a range of salaries.

```
<form>
  ...
  <p>What is your salary range?</p>
  <select>
    <option value="1">Under $30,000</option>
    <option value="2">$30,000 - 60,000</option>
    <option value="3">$60,000 - 100,000</option>
    <option value="4">Over $100,000</option>
  </select>
</form>
```



Full Name
Email
Telephone
Age
Date of Birth mm/dd/yyyy

Check the bands you enjoy...

- Journey
- REO Speedwagon
- Electric Lights Orchestra
- The Cure

Please select your favorite city

- New York
- Boston
- Chicago
- San Francisco

What is your salary range?

Under $30,000
Under $30,000
$30,000 - 60,000
$60,000 - 100,000
Over $100,000

These dropdowns aren't mobile friendly. If you know that your audience will be using mostly mobile devices, you should use some other type of form field other than the `select` tag.

Do This:

Now, it's time for you to get a little practice on your own. Start a new document and name it contact-form.html. Using the code above as a guide write a form displays a field for each of the following:

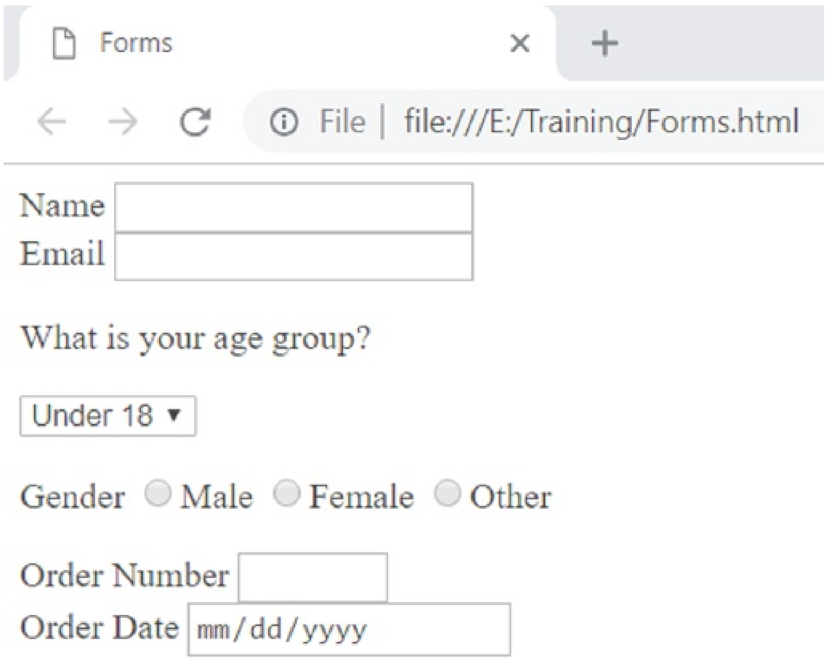
- Name
- Email
- Phone
- Website
- Message

Make sure to test it in the browser, so you know that it is working correctly before moving on.

Debug This:

There are errors in this code preventing it from displaying the form correctly. Remember to use the correct input type for each label and restrict the order number from going above `100` or below `1`.

Fix the errors, so the form displays correctly in your browser like this:



Name
Email

What is your age group?

Under 18

Gender Male Female Other

Order Number
Order Date mm/dd/yyyy

Here's the code to debug.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Forms</title>
  </head>
  <body>
    <form action="#" method="post">

      <input type="text" id="name" name="name" />
      <label>Email</label>
      <input type="email" />
      <br/>
      <p>What is your age group?</p>

        <option value="2">18-24</option>
        <option value="3">24-40</option>
        <option value="4">40+</option>

      <br>
      <p>Gender</p>
        <input type="checkbox" name="city" value="M">Male
        <input type="checkbox" name="city" value="F">Female
        <input type="checkbox" name="city" value="O">Other
        <label for="oNum">Order Number</label>
      <input type="number"/><br/>
        <label>Order Date</label>
      <input type="text" id="oDate" name="oDate" value="mm/dd/yyyy"/>
    </form>
  </body>
</html>
```

Submit This: Sign Up Form

Create an HTML5 document from scratch that is correctly formed and coded that includes at least 8 fields. All of the input types you've studied should be incorporated.

Research how to make a password field and add it in your form as well.

Remember, when submitting the work please use the following naming convention for your file: HTMLAUTHORING_LastName_SectionName.html . So if your last name is Smith and you are submitting this section your file name should be HTMLAUTHORING_Smith_forms.html .

For this course visit https://www.dropbox.com/request/RhW9kBDXtisq2Fsvg3hY to submit your assignments.