LET'S GET EVERYTHING SET UP!

1. In Schoology, go to: **Courses(in the top menu) > FEWD CHI 1: Section 1**
2. Then go to the **Class Materials** folder — it's the pink one!
3. Navigate to the **Week 8 (It's the yellow folder) > Lesson 15 folder**
4. There you'll find all the materials for today's class
5. Download starter_code_lesson_15.zip
6. Move it from your Downloads folder to your Desktop
7. Double-click on starter_code_lesson_15.zip to unzip it
8. After you've unzipped, delete the original .zip to avoid confusion and make sure you don't unzip it again later!!!

FINAL PROJECT

GENERAL ASSEMBLY

ADVANCED CSS POSITIONING & FORM BASICS

Sarah Holden

LEARNING OBJECTIVES

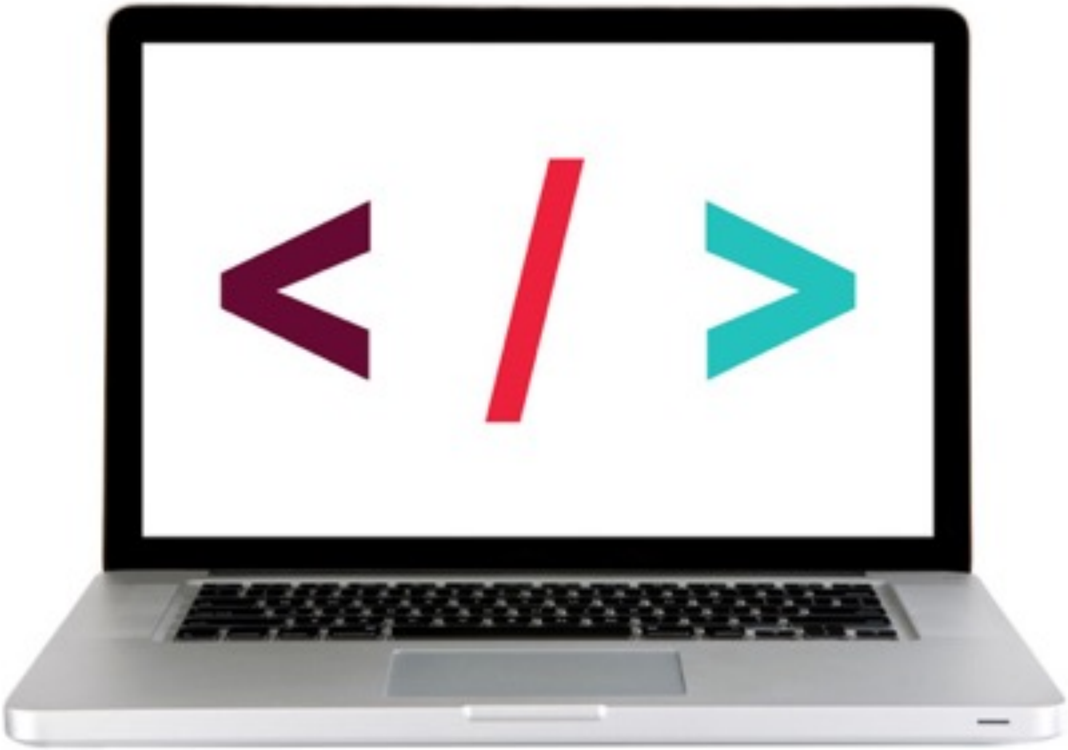
- Identify and differentiate between different CSS positioning techniques
- Be able to differentiate the different types of inputs and why/where we would use each.

AGENDA

- Advanced CSS Positioning
- Forms and Inputs
- Lab
- Form Validation

ADVANCED CSS POSITIONING

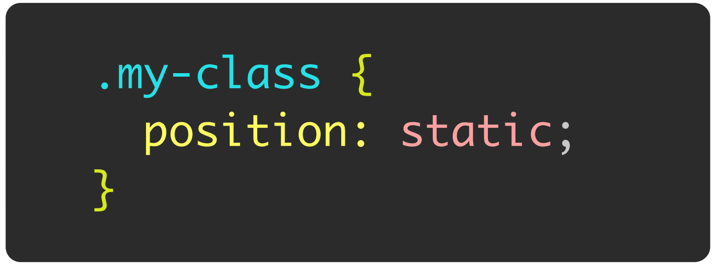
LET'S TAKE A CLOSER LOOK



[Positioning 101]

STATIC POSITIONING

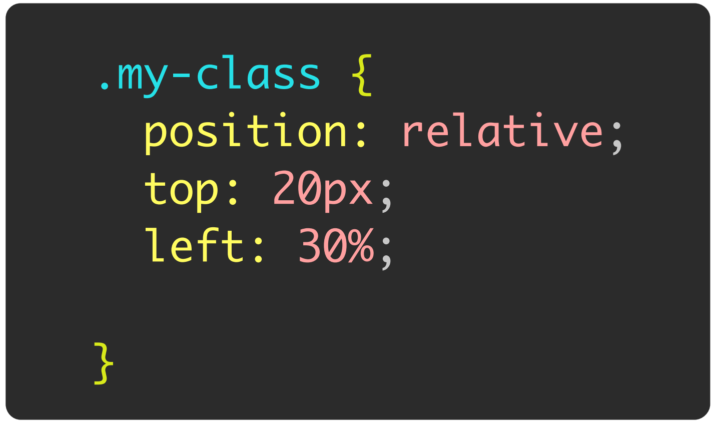
- This is the normal flow of the document, the **default**
- Elements render in order, as they appear in the document flow.



```
.my-class {
  position: static;
}
```

RELATIVE POSITIONING

- Relative positioning moves an element *relative to where it would have been in normal flow*.
- For example, "left: 20px" adds 20px to an element's **left** position
- Creates a *coordinate system* for child elements.

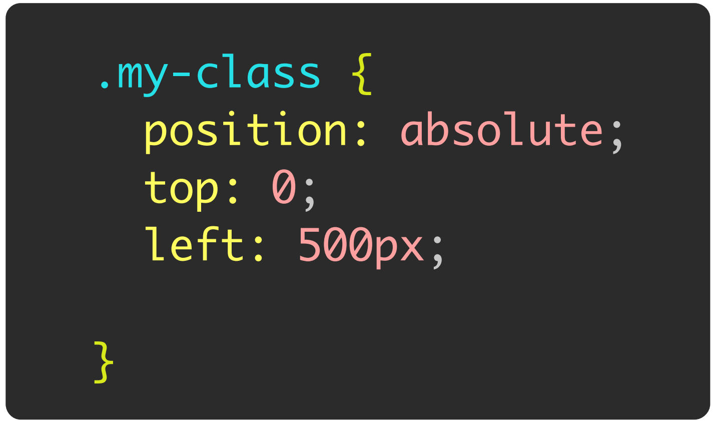


```
.my-class {
  position: relative;
  top: 20px;
  left: 30%;

}
```

ABSOLUTE POSITIONING

- When the *position* property is given a value of *absolute*, an element is taken out of the normal flow of the document.
- This element no longer affects the position of other elements on the page (they act like it's not there).
- You can add the *right*, *top*, *left* and *bottom* properties to specify where the element should appear relative to its first positioned (not static) ancestor element

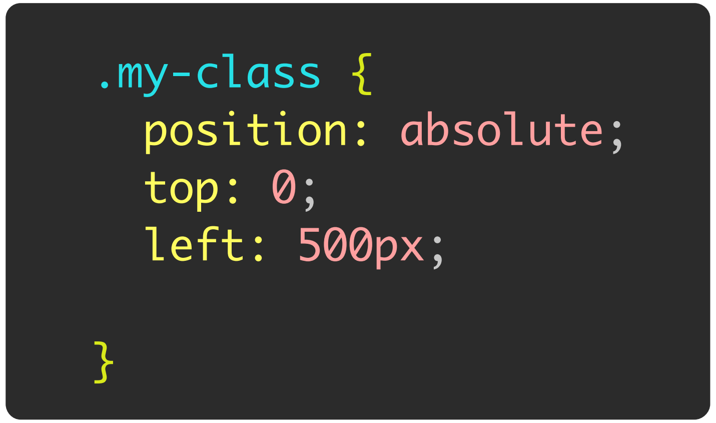


```css
.my-class {
  position: absolute;
  top: 0;
  left: 500px;

}
```

FIXED POSITIONING

- When the *position* property is given a value of *fixed*, the element is positioned in relation to *the browser window*
- When the user scrolls down the page, it stays in the same place.
- You can add the *right*, *top*, *left* and *bottom* properties to specify where the element should appear in relation to the browser window.

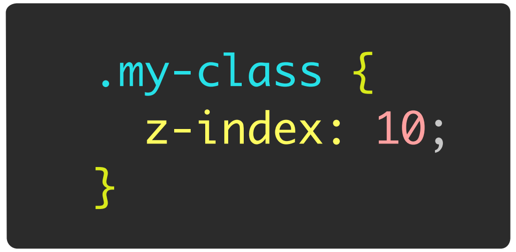


```
.my-class {
  position: absolute;
  top: 0;
  left: 500px;

}
```

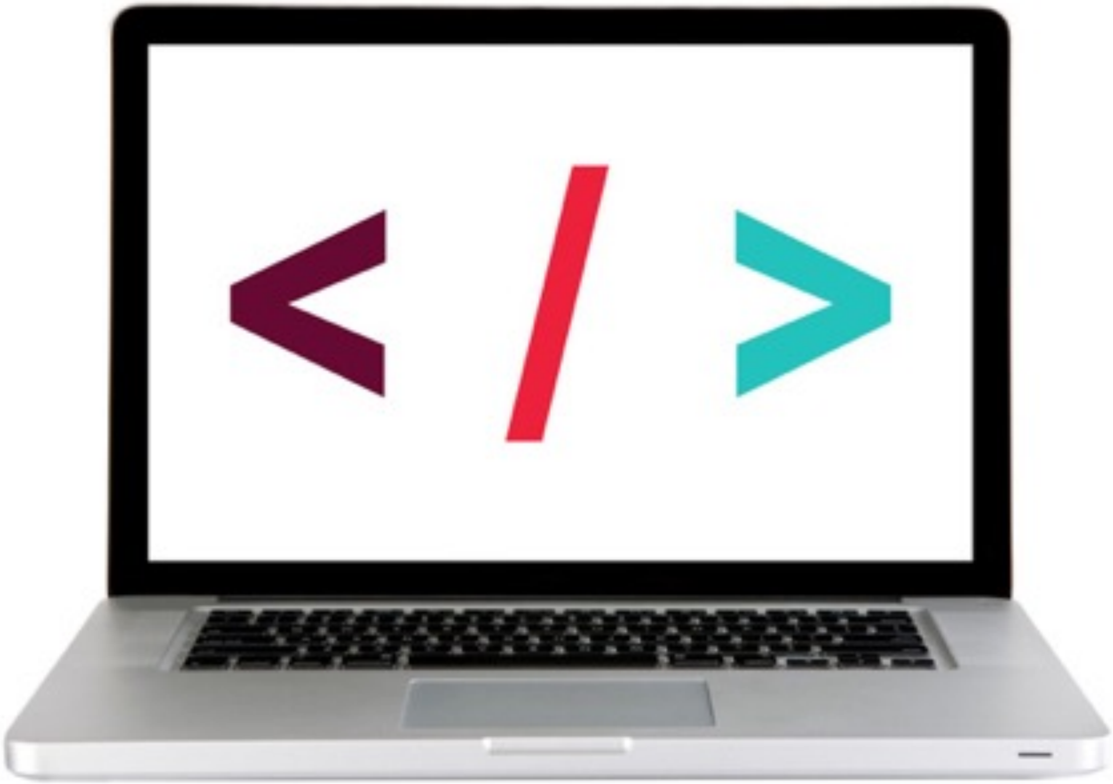
OVERLAPPING ELEMENTS — Z-INDEX

- When using relative, fixed or absolute positioning, elements can overlap.
- When elements overlap, the elements that appear later in the HTML code sit on top of those that appear earlier in the page.
- If you want to control which elements are layered on top of each other, you can use the z-index property.
- This property takes a number — the higher the number the closer that element is to the front.
- Similar to 'bring to front' and 'send to back' in programs like *Adobe Illustrator*.



```
.my-class {
  z-index: 10;
}
```

LET'S TAKE A CLOSER LOOK



Positioning Fun Code Along

WANT TO LEARN MORE?

Resources for more info/examples:

- Textbook (CSS & HTML): Pages 363 - 369
- A List Apart: CSS Positioning 101

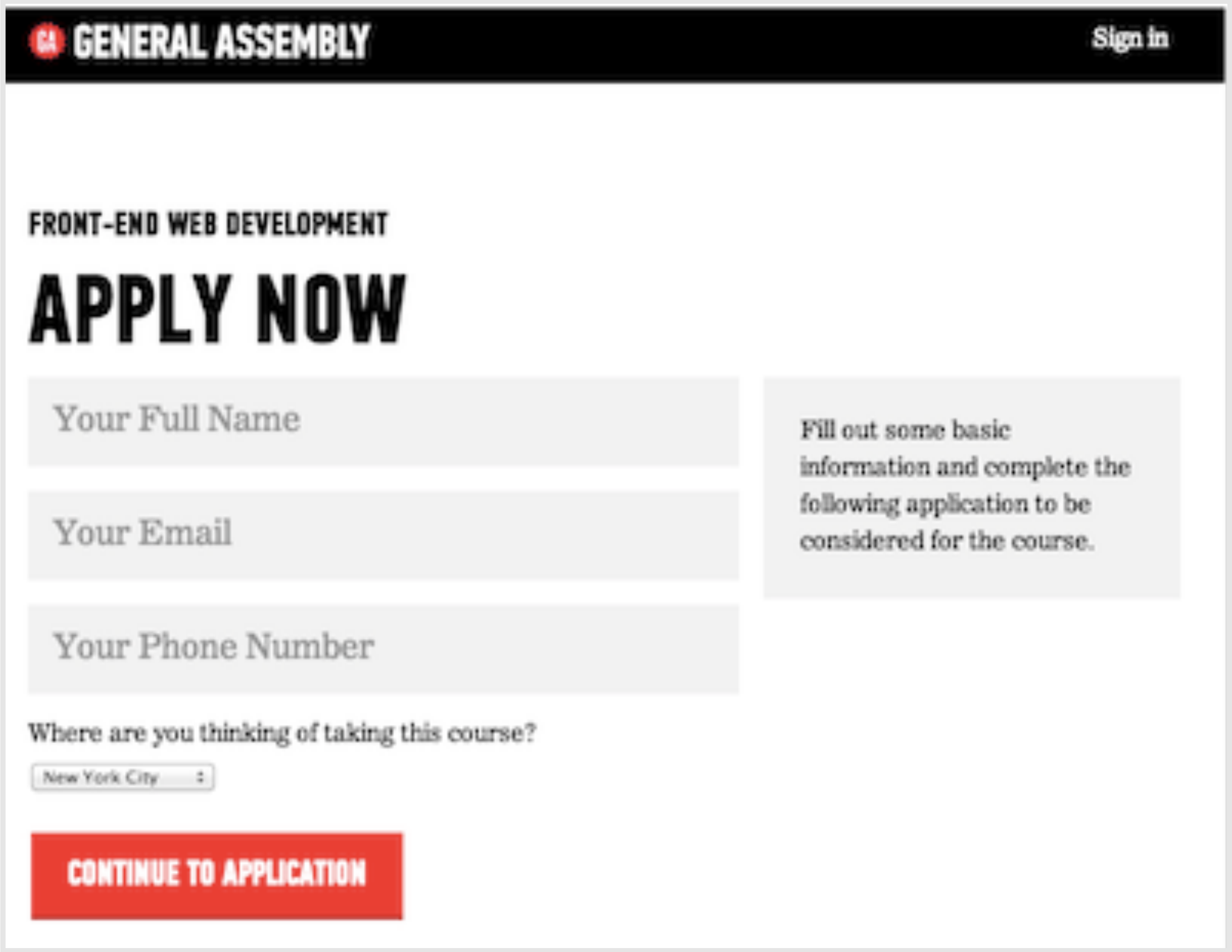
FORM BASICS

FORMS AND INPUTS

FORM BASICS

FORMS

How we get data from users



GENERAL ASSEMBLY

Sign in

FRONT-END WEB DEVELOPMENT

APPLY NOW

Your Full Name

Your Email

Your Phone Number

Fill out some basic information and complete the following application to be considered for the course.

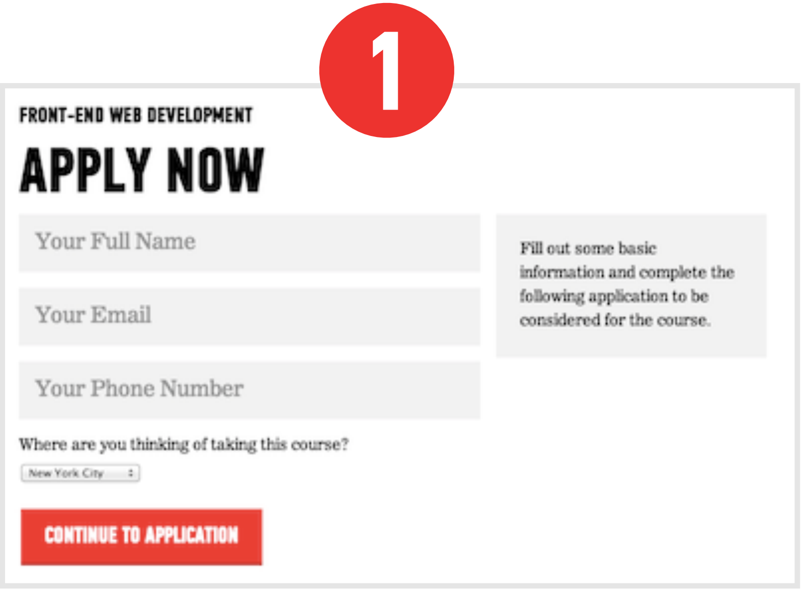
Where are you thinking of taking this course?

New York City

CONTINUE TO APPLICATION

FORMS

1. The user fills out the form and presses the submit button



FRONT-END WEB DEVELOPMENT

APPLY NOW

Your Full Name

Your Email

Your Phone Number

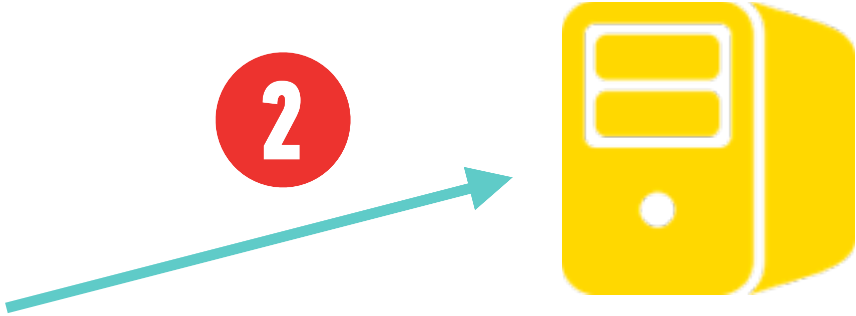
Fill out some basic information and complete the following application to be considered for the course.

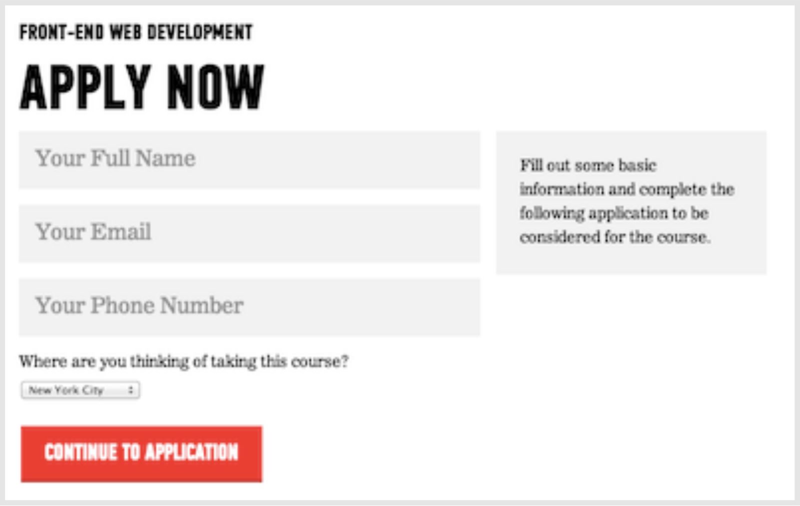
Where are you thinking of taking this course?

New York City

FORMS

2. The **name** of each form field is sent to the server along with the **value** the user entered or selected





FRONT-END WEB DEVELOPMENT

APPLY NOW

Your Full Name

Your Email

Your Phone Number

Fill out some basic information and complete the following application to be considered for the course.

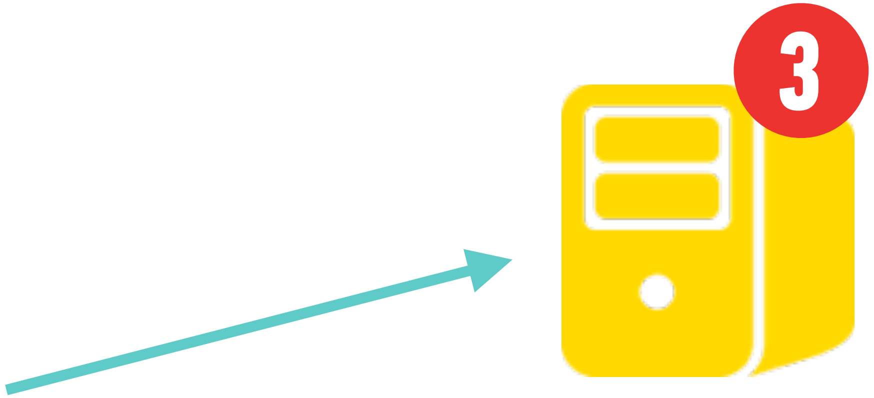
Where are you thinking of taking this course?

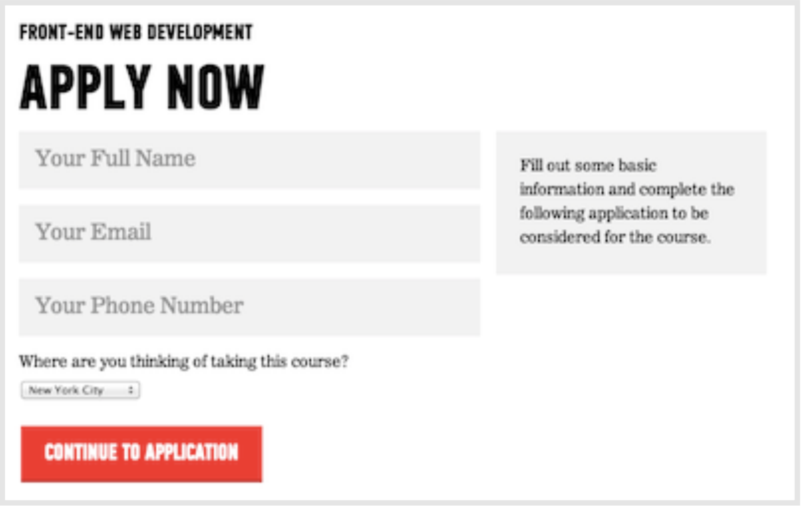
New York City

CONTINUE TO APPLICATION

FORMS

3. The server processes the data using a language such as PHP, C# or Java. It may also store the information in a database





FRONT-END WEB DEVELOPMENT

APPLY NOW

Your Full Name

Your Email

Your Phone Number

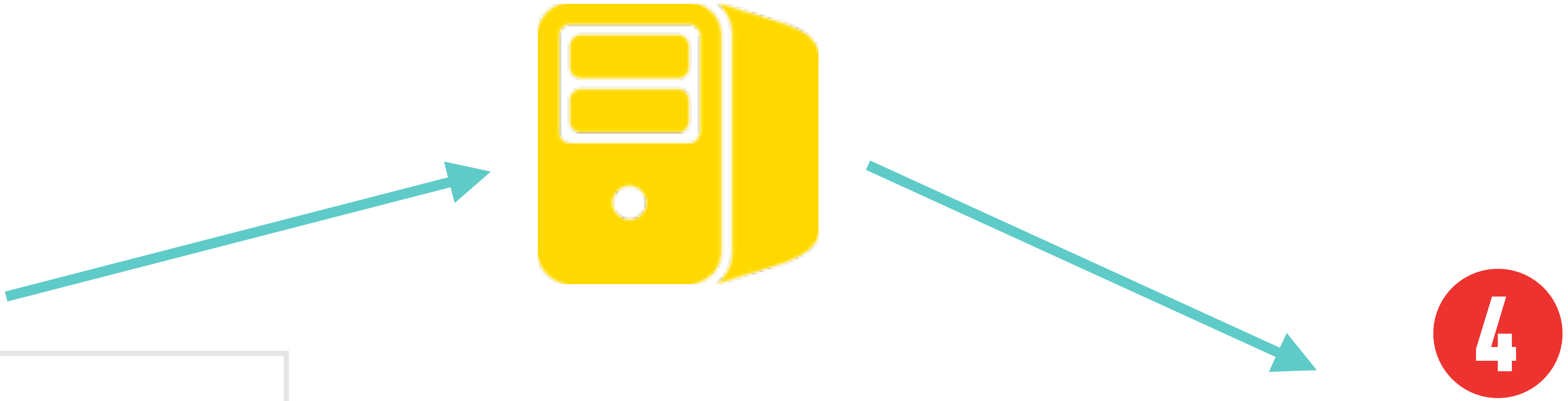
Fill out some basic information and complete the following application to be considered for the course.

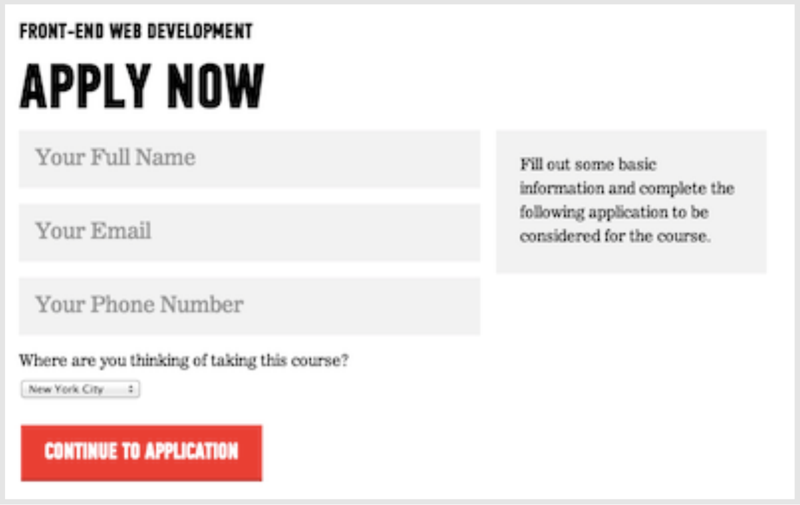
Where are you thinking of taking this course?

New York City

FORMS

4. The server creates a new page to send back to the browser based on the information received.




FRONT-END WEB DEVELOPMENT

APPLY NOW

Your Full Name

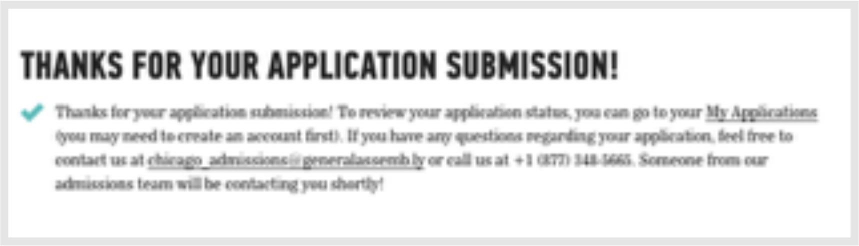
Your Email

Your Phone Number

Fill out some basic information and complete the following application to be considered for the course.

Where are you thinking of taking this course?

New York City


THANKS FOR YOUR APPLICATION SUBMISSION!

✓ Thanks for your application submission! To review your application status, you can go to your My Applications (you may need to create an account first). If you have any questions regarding your application, feel free to contact us at chicago_admissions@generalassemb.ly or call us at +1 (877) 348-5665. Someone from our admissions team will be contacting you shortly!

FORMS

Form controls live inside the <form element>

```
<form action="http://www.example.com/login.php" method="post">
  <!--Data collection elements go here-->
</form>
```

FORMS

Form attributes:

```
<form action="http://www.example.com/login.php" method="post">
  <!--Data collection elements go here-->
</form>
```

FORMS

Form attributes:

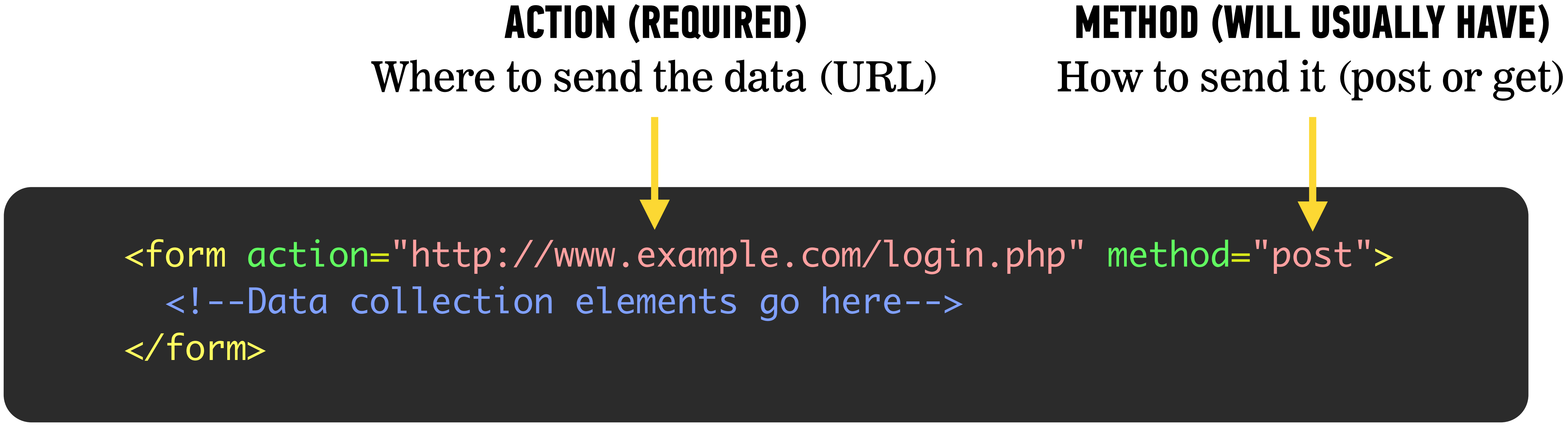
ACTION (REQUIRED)

Where to send the data (URL)

```
<form action="http://www.example.com/login.php" method="post">
  <!--Data collection elements go here-->
</form>
```

FORMS

Form attributes:



ACTION (REQUIRED)

Where to send the data (URL)

METHOD (WILL USUALLY HAVE)

How to send it (post or get)

```html
<form action="http://www.example.com/login.php" method="post">
  <!--Data collection elements go here-->
</form>
```

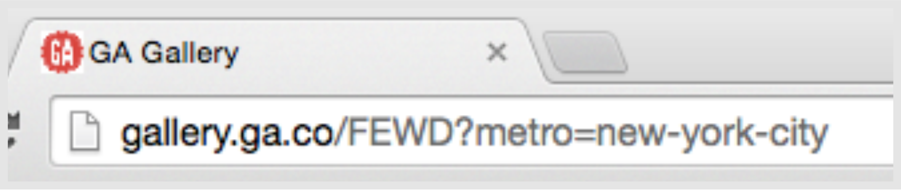
FORMS — METHODS

POST

- Data is not shown in URL
- Can contain sensitive data
- No size limitations
- Adds information to, or deletes info from a database

GET

- Short forms (such as search fields)
- Appended to URL in name/value pairs
- Never use for sensitive info!!!
- Useful for form submissions when user wants to bookmark results



gallery.ga.co/FEWD?metro=new-york-city

```
<form action="http://www.example.com/login.php" method="post">
  <!--Data collection elements go here-->
</form>
```

FORM BASICS

GETTING INFORMATION FROM USER

GETTING INFO — INPUTS

Place any inputs between <form></form> tags

Attributes:

- **Type** — text, submit, password, email, checkbox, button, radio, file, etc.
- **Name, value** — The name attribute is sent to the user along with the value the user selects.
- **Placeholder** — For text inputs - hint for what user should enter in field

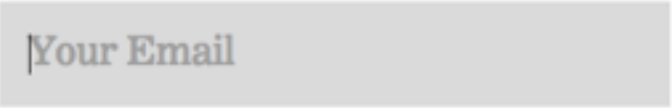
Note: For a complete spec see MDN

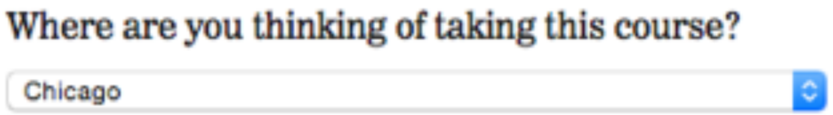
FORM

ADD TEXT

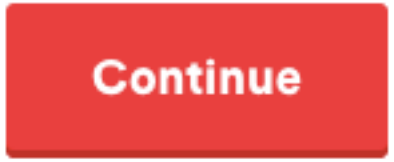
MAKE CHOICES

SUBMIT




Where are you thinking of taking this course?

Chicago



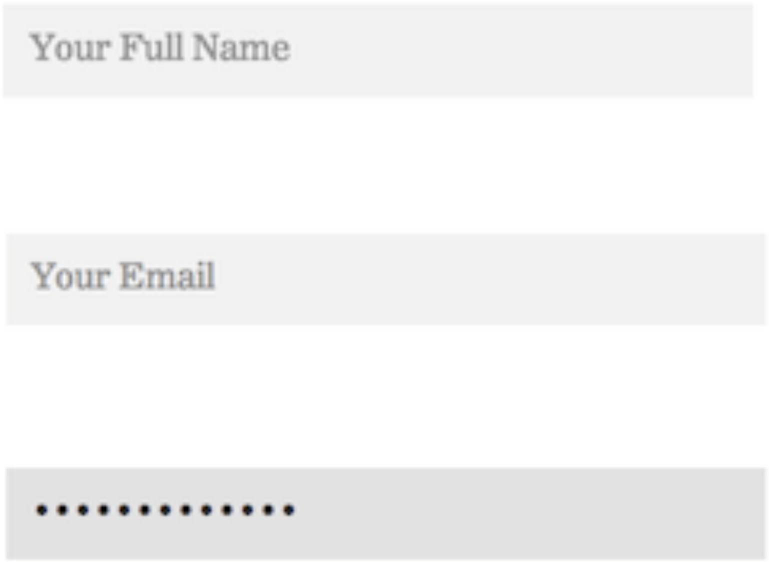
INPUTS — TEXT

ADD TEXT

```
<input type="text" name="username">
```

```
<input type="email" name="email">
```

```
<input type="password" name="password">
```



Your Full Name

Your Email

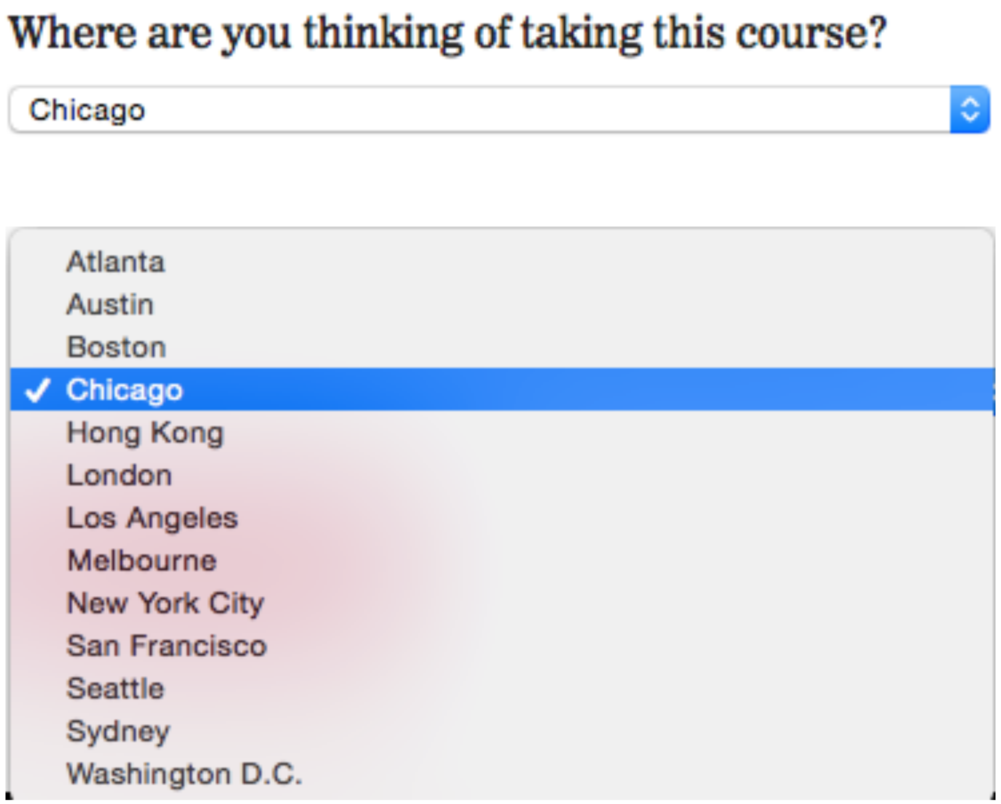
••••••••••••••

*Can also carry a maxlength attribute to limit the number of characters the user may enter

SELECT AND OPTION

MAKE CHOICES

```
<select name="referral">
  <option value="friend">Friend</option>
  <option value="instructor">Instructor</option>
  <option value="online">Online</option>
</select>
```



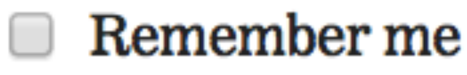
Where are you thinking of taking this course?

Chicago

CHECKBOXES AND RADIO BUTTONS

MAKE CHOICES

```
<input type="checkbox" name="store_credentials">
```



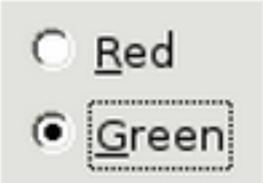
□ Remember me

RADIO BUTTONS

MAKE CHOICES

Radio buttons are grouped together by their name attribute

```html
<input type="radio" name="color" value="red" label="Red">
<input type="radio" name="color" value="green" label="Green" checked="checked">
```



○ Red

● Green

LABELS

Information about the input field should be put in a <label> tag:

```
<label for="yourName">Name</label>
<input type="text" name="name" id="yourName">
```

To tie the two together:

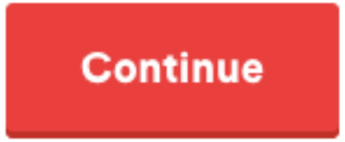
```
<label for="yourName">Name</label>
<input type="text" name="name" id="yourName">
```

Note: Clicking the label text places the focus in the input field (great for radio buttons)

SUBMIT FORM

SUBMIT

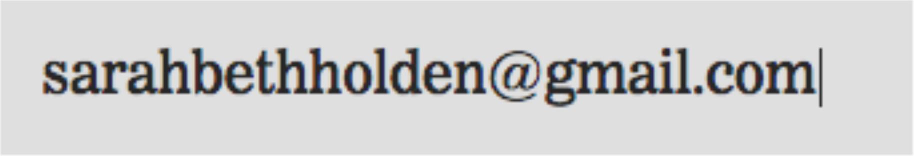
```
<input type="submit" value="Continue">
```



NAME/VALUE PAIRS

- Information is sent from the browser to the server using name/value pairs.

```
<input type="text" name="username">
```

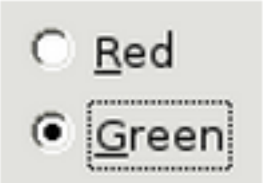


sarahbethholden@gmail.com

NAME VALUE

username=sarahbethholden@gmail.com

```
<input type="radio" name="color" value="red" label="Red">
<input type="radio" name="color" value="green" label="Green" checked="checked">
```



○ Red
● Green

NAME VALUE

color=green

LAB

ACTIVITY

EXERCISE

KEY OBJECTIVE

- Identify input types, add styles to a form

TYPE OF EXERCISE

- Individual/partner

TIMING

8 min

1. Review Screenshots for the course application form and discuss with a partner

Until 8:30

2. Write HTML for the form
3. Style the form with CSS. Focus on getting the form centered and getting the information on the right rows, and then add other styles if you have time.

* You will need to look up the optgroup and textarea elements

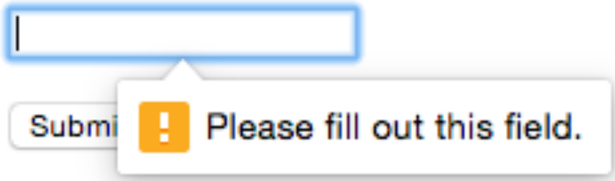
FORM BASICS

VALIDATION

VALIDATION

- You've probably seen forms on the web that give users messages if the form control has not been filled out correctly.
- Traditionally, validation has been performed using Javascript.
- HTML5 also introduced browser-based form validation.

```
<input type="text" name="fullname" required />
```



Please fill out this field.

VALIDATION

- For more substantial validation, it is highly recommended that you use a validation library, such as Parsley.
- To add parsley validation:

1. Add jQuery to your project
2. Add the parsley.js file to your project after you've included jQuery

```
<script src="js/jquery-2.1.3.min.js"></script>
<script src="js/parsley.js"></script>
```

3. Add the data-parsely-validate attribute to your form tag

```
<input data-parsley-validate/>
```

4. Add the required attribute to any fields you want to be required.

```
<input type="text" name="fullname" required />
```

LEARNING OBJECTIVES

- Identify and differentiate between different CSS positioning techniques
- Be able to differentiate the different types of inputs and why/where we would use each.

HOMEWORK

HOMEWORK

FINAL PROJECT MILESTONES:

- Milestone 3 First Draft: Build The First Draft Of Your Final Project.
  Due March 14th

FORM BASICS

EXIT TICKETS