

---

**FEWD**

---

# FINAL PROJECTS

# AGENDA

---



- Git Workflow Review
- Class / Project Schedule
- SCSS
- Final Project Lab Time

---

**FEWD**

---

**REVIEW**

---

**FEWD**

---

# FORM BASICS

# 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:

**ACTION (REQUIRED)**

Where to send the data (URL)

**METHOD (WILL USUALLY HAVE)**

How to send it (post or get)



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

# FORMS — METHODS

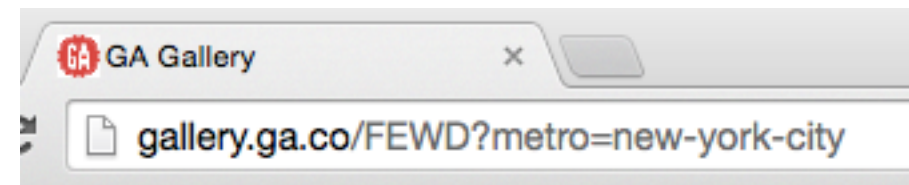
---

## 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



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

---

**FEWD**

---

# 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](#)*

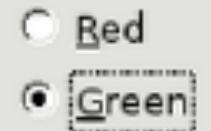
# NAME/VALUE PAIRS

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

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

NAME	VALUE
username	eboyer@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

# INPUTS — TEXT

---

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

Your Full Name

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

Your Email

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

.....

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

---

# 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)*

# SELECT AND OPTION

```
<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

- Atlanta
- Austin
- Boston
- ✓ Chicago
- Hong Kong
- London
- Los Angeles
- Melbourne
- New York City
- San Francisco
- Seattle
- Sydney
- Washington D.C.

---

# CHECKBOXES AND RADIO BUTTONS

---

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

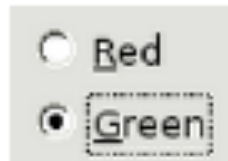
☐ Remember me

# RADIO BUTTONS

---

Radio buttons are grouped together by their name attribute

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



Red  
Green

---

# SUBMIT FORM

---

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

Continue



---

**FEWD**

---

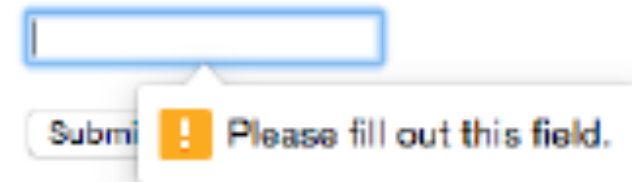
**VALIDATION**

# 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 />
```



---

# 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-parsley-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 />
```

---

**FEWD**

---

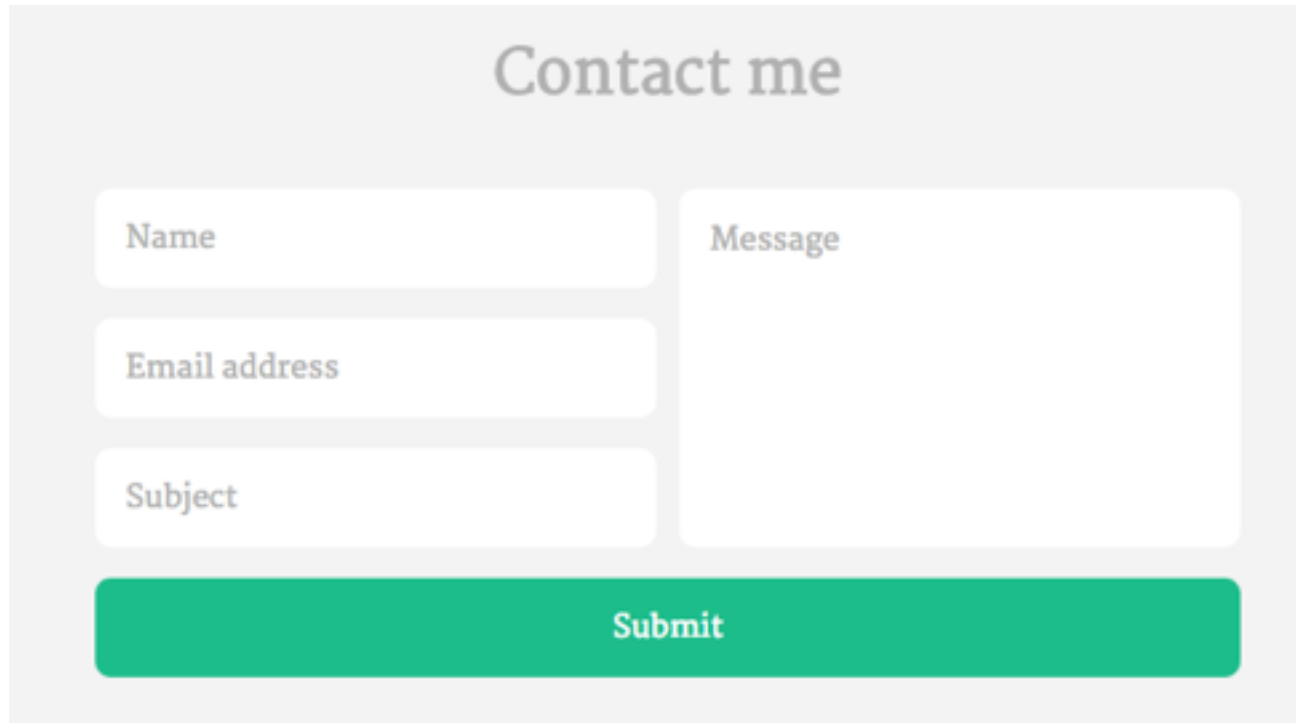
# CONTACT FORM

---

# BUILDING A FUNCTIONAL CONTACT FORM

---

1. I recommend [FormSpree](#) for contact forms that send you an email without having to use PHP.
2. Your site will need to be hosted on a server in order to test emails.



A contact form titled "Contact me" is displayed on a light gray background. The form consists of four input fields and a submit button. On the left side, there are three stacked text input fields labeled "Name", "Email address", and "Subject". To the right of these is a larger text area labeled "Message". At the bottom of the form is a wide, rounded green button with the word "Submit" in white text.



Sass

# SASS

---

## Variables

```
$primaryColor: #FD8A71;  
$secondaryColor: #5DAD7F;  
  
.Button{  
  background: $primaryColor;  
}  
  
a{ color: $primaryColor; }
```



# SASS

---

## Math

```
$containerWidth: 100% - 20px;  
  
// Parent container is 980px  
  
.Boundary{  
  width: $containerWidth;  
  // Will compile to 960px  
}
```

# SASS

---

## Math

```
$containerWidth: 100%;  
  
.Third{  
  width: $containerWidth / 3;  
  // Will compile to 33.3333%  
}
```

# SASS

---

## Nesting

```
.MainNavigation{}

.MainNavigation > nav{
  overflow: hidden;
}

.MainNavigation > nav > ul{
  list-style: none;
}

.MainNavigation > nav > ul > li{
  margin: 0 10px 0 0;
}
```

```
.MainNavigation{
  > nav{
    overflow: hidden;
    > ul{
      list-style: none;
      > li{
        margin: 0 10px 0 0;
      }
    }
  }
}
```

---

# HOW TO USE SASS

---



---

**FEWD**

---

# **HOMEWORK**

**CSS AND RESPONSIVENESS DONE THIS WEEK**

---

**FEWD**

---

# EXIT TICKETS