CS 338: Graphical User Interfaces

Lecture 3-2: Web Development II

Some materials adapted from
http://developer.mozilla.org
http://w3schools.com
http://getbootstrap.com
http://flask.pocoo.org

Dev Week Continued

- So far, we saw:
  - HTML
  - CSS
  - Bootstrap (the CSS part)
- This gives us a foundation for structure and style
- But with only these components, the web page is completely static — just a unchanging page
- We continue our focus on:
  - Short Review of JavaScript
  - jQuery, Bootstrap (the JavaScript part)
  - Flask (?)

JavaScript

- Dates back to 1995 (part of Netscape)
- Adopted by all major browsers today
- Interpreted
  - no compilation of code
- Weakly typed
  - code doesn't specify types of variables, arguments, etc.
- First-class functions
  - treats functions as values
  - store them, pass them as arguments, etc.

JavaScript

- Language tidbits
  - Many things look the same as Java
    - Operators

      `+, *, &&, ||, ==, …`

    - Conditionals and loops

      `if, switch, for, while, …`

JavaScript

- Language tidbits
  - Some things are a little different
    - Variable declaration

```
var a = 2;
```

    - Arrays and Objects (associative arrays)

```
var myArray = [“hello”, “there”];
alert (myArray[0]);

var myAssoc = new Array();
myAssoc[“name”] = “John”;
alert (myAssoc[“name”]);
```

JavaScript

- Language tidbits
  - A straightforward recursive function

```
function factorial(n) {
    if (n == 0) return 1;
    return n * factorial(n - 1);
}
```

JavaScript

- Language tidbits
  - A function with *closure*

```
function incrementCreator() {
    var count = 0;
    return function () {
        return ++count;
    };
}
var inc = incrementCreator();
alert(inc()); /
alert(inc()); /
alert(inc()); /
```

[from http://en.wikipedia.org/wiki/JavaScript]

JavaScript

- Adding JavaScript to a web page
  - Loading a JS file in the header:

```
<script type="text/javascript" src="URL"></script>
```

  - Embedding JS directly into the page

```
<html>
  <head … </head>
  <body>

    <script>
      document.write('Hello World!’);
    </script>

  </body>
</html>
```

JavaScript

- Handling events
  - Event handling is normally tied to a DOM object
    - The object generates the event (e.g., when clicked); the handler receives and then handles the event
  - Embedding the handler in HTML

```
<button id=“mybutton” onclick="doSomething()">
    My Button
</button>
```

  - Specifying the handler in JavaScript

```
button.onclick = ”doSomething()”;
 OR
button.onclick = doSomething;
 OR
button.onclick = function() { … };
```

JavaScript

- Handling events
  - What work is done inside the handler?
  - Retrieve and/or compute data, if needed
    - We will deal with this in the future when we discuss AJAX
  - Find DOM elements to update

```
var el = document.getElementById("myElement");
```

  - Change some property of these elements

```
el.style.color = "blue";
el.style.fontSize = "20pt";
```

We interrupt this program…

- We could talk for a long while about ways to search and manipulate the DOM with standard JavaScript.
- But the truth is, most developers don't use these methods directly, but instead, work through a higher-level framework.
  - Much like how developers could lay out web pages using only HTML and CSS, but typically use frameworks like Bootstrap.
- So let's move directly to the common case…

jQuery

- jQuery aims to help developers with “HTML document traversal and manipulation, event handling, animation, and Ajax” [jquery.com]
- Originally developed by John Resig in 2006
- “As of June 2018, jQuery is used on 73% of the top 1 million websites” [wikipedia.org]

jQuery

- Selectors
  - One of the most heavily used aspects of jQuery, the **selectors** search for objects within the DOM
  - All selectors have the form $("<*selector*>") and return a list of elements that match the selector
  - <*selector*> uses much of the syntax of CSS specifications
  - Select by… (with examples)
    - Tag: **$("p")** – matches all <p> elements
    - Id: **$("#div17 ")** – matches element with id="div17"
    - Class: **$(".menu")** – matches all elements with class="menu"

jQuery

- Selectors
  - More examples

$("*")	Selects all elements
$(this)	Selects the current HTML element
$("p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element
$("ul li:first")	Selects the first element of the first
$("ul li:first-child")	Selects the first element of every
$("[href]")	Selects all elements with an href attribute
$("a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"
$("a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"
$(":button")	Selects all <button> elements and <input> elements of type="button"
$("tr:even")	Selects all even <tr> elements
$("tr:odd")	Selects all odd <tr> elements

jQuery

- Actions
  - Once you've selected something, you can perform actions on the returned list of elements
  - Set text or html:

```
$("#test1").text("Hello world!");
$("#test2").html("<b>Hello world!</b>");
```

  - Change styling:

```
$("p").css("background-color", "yellow");
```

  - Add content:

```
$("p").append("Some appended text.");
```

jQuery

- Actions
  - Create a new div and add it to a parent div:

```
el = $('<div></div>');
$(".parent").append(el);
```

  - Hide/show an element:

```
$("div").hide();
$("div").show();
```

jQuery

- What does this code do?

```
$('<tr>').attr('id', id).append(
    $('<td>').append($('<img>').attr('src', imgSrc))
).append(
    $('<td>').text(text)
).append(
    $('<button>').addClass('btn').text('Click').click(function () {
        alert('Hello.');
    })
).appendTo('tbody');
```

jQuery

- What does this code do?

```
$('<tr>').attr('id', id).append(
    $('<td>').append($('<img>').attr('src', imgSrc))
).append(
    $('<td>').text(text)
).append(
    $('<button>').addClass('btn').text('Click').click(function () {
        alert('Hello.');
    })
).appendTo('tbody');
```

  - Creates a table row <tr>…</tr> and appends it to the body <tbody>…</tbody> of a table
    - Cell 1 has an image
    - Cell 2 has simple text
    - Cell 3 has a button that, when click, generates an alert

jQuery

- Handling events, revisited
  - We just saw the base JavaScript…

```
var el = document.getElementById("myElement");
el.style.color = "blue";
el.style.fontSize = "20pt";
```

  - Here's the same in jQuery…

```
var el = $('#myElement');
el.css('color', 'blue');
el.css('font-size', '20pt');
```

  - Better yet, here's a more compact version…

```
$('#myElement').css('color', 'blue').css('font-size', '20pt');
```

jQuery

- Handling events, revisited
  - You can use jQuery to attach an event handler to an element — normally only when the document is loaded
    - E.g., a button that changes text when clicked:

```
$(document).ready(function() {

    $(<button>).click(function() {
        $(<text>).<action>();
    });

});
```

  - Here, we're attaching a handler to both the document (DOM) and the button

jQuery

- Handling events, revisited
  - Let's look at a full example:
    - Simple with text
    - 4 buttons to change the text color or hide/show the text

```
<!DOCTYPE html>
<html>

<head>
    <title>jQuery Test</title>
    <script src="../course/lib/jquery/jquery-3.3.1.min.js"></script>
</head>

<body>

    SEE NEXT SLIDE

</body>

</html>
```

```
<div id="text-block">
    Hello!
</div>

<div>
    <button id="blue-button">Blue</button>
    <button id="black-button">Black</button>
    <button id="hide-button">Hide</button>
    <button id="show-button">Show</button>
</div>

<script>

    $(document).ready(function () {

        $('#blue-button').click(function () {
            $('#text-block').css('color', 'blue');
        });

        $('#black-button').click(function () {
            $('#text-block').css('color', 'black');
        });

        $('#hide-button').click(function () {
            $('#text-block').hide();
        });

        $('#show-button').click(function () {
            $('#text-block').show();
        });

    });

</script>
```

Example

- Let's say we wanted to add dynamic dropdown menus to The Goat Pasture
- We now have the tools to do this via CSS & JavaScript…

Example

- Additions to HTML / JS

```
<span class="menu">
    <a id="school-link" href="#" class="menu-item">Goat School</a>
    <div id="school-dropdown" class="menu-dropdown">
        <div>
            <a class="menu-item" href="#">The Goat Family</a>
        </div>
        <div>
            <a class="menu-item" href="#">What Do Goats Eat?</a>
        </div>
    </div>
</span>
<script>
    $(document).ready(function () {
        $('#school-link').click(function () {
            if ($('#school-dropdown').is(':hidden'))
                $('#school-dropdown').show();
            else
                $('#school-dropdown').hide();
        });
    });
</script>
```

Example

- Additions to HTML / JS

```
<span class="menu">
    <a id="school-link" href="#" class="menu-item">Goat School</a>
    <div id="school-dropdown" class="menu-dropdown">
        <div>
            <a class="menu-item" href="#">The Goat Family</a>
        </div>
        <div>
            <a class="menu-it                    ts Eat?</a>
        </div>
    </div>
</span>
<script>
    $(document).ready(function () {
        $('#school-link').click(function () {
            if ($('#school-dropdown').is(':hidden'))
                $('#school-dropdown').show();
            else
                $('#school-dropdown').hide();
        });
    });
</script>
```

Could use hover() as an alternative

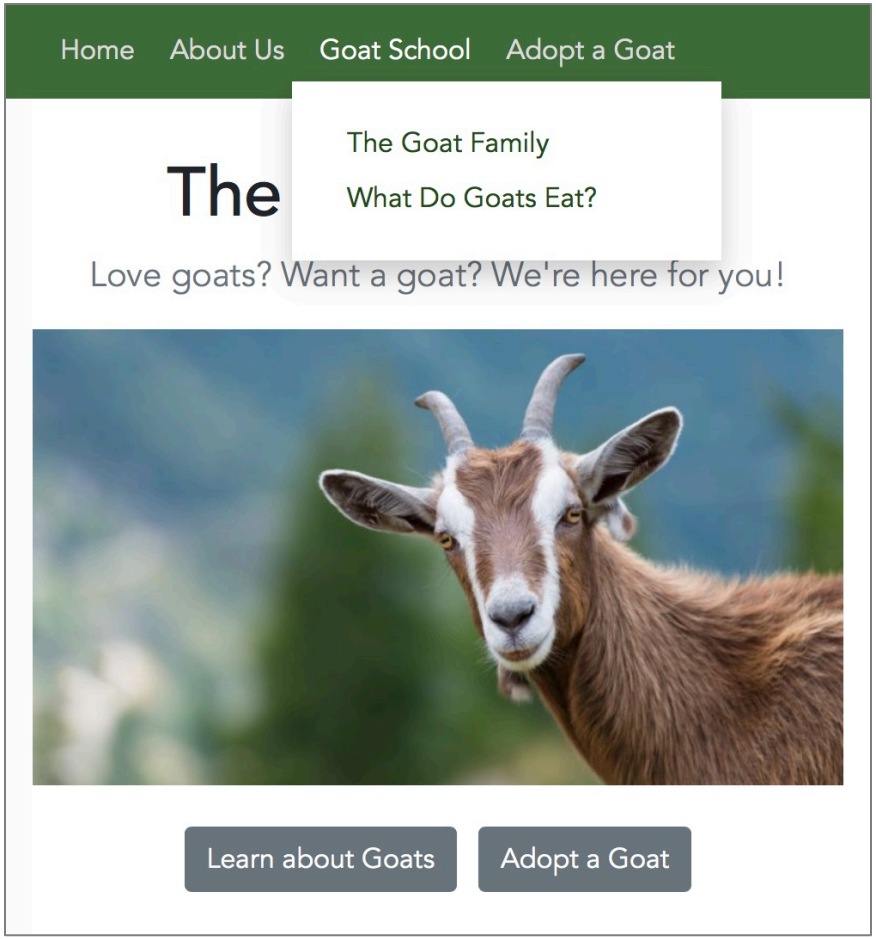
Example

- Additions to CSS

```css
.menu {
    position: relative;
    display: inline-block;
}

.menu-dropdown {
    display: none;
    position: absolute;
    top: 30px;
    background-color: white;
    min-width: 250px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    padding: 1rem;
    z-index: 1;
}

.menu-dropdown div {
    margin: .5rem 0;
}

.menu-dropdown a, .menu-dropdown a:hover {
    color: #204c19;
}
```

Example

- Resulting menu looks pretty good
- But what about…
  - Activating submenu items?
  - Clicking a different menu?
  - Clicking elsewhere on the page?
  - Different browsers?
- It gets complicated fast



Bootstrap Revisited

- We've already seen Bootstrap's functionality when it comes to styling
- Bootstrap (and packages like it) also include JavaScript functionality for common components
  - Suggested HTML/CSS/JS all work together hand-in-hand

Example

- Updated HTML with Bootstrap menus

```
<!DOCTYPE html>
<html>

<head>
    <title>The Goat Pasture - Home</title>
    <link href="../course/lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="style5.css">
    <script src="../course/lib/jquery/jquery-3.3.1.min.js"></script>
    <script src="../course/lib/popper/popper.min.js"></script>
    <script src="../course/lib/bootstrap/js/bootstrap.min.js"></script>
</head>

<body>

    <header>
        <div class="container">
            <div class="row">
                <div class="col-sm">
                    <nav class="navbar navbar-expand-md navbar-dark">
                        <a class="navbar-brand" href="/"><img src="../course/img/goat0.png" style="max-height: 40px;"></a>
                        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
                            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                            <span class="navbar-toggler-icon"></span>
                        </button>

                        <div class="collapse navbar-collapse" id="navbarSupportedContent">
                            <ul class="navbar-nav mr-auto">
                                <li class="nav-item active">
                                    <a class="nav-link" href="/">Home</a>
                                </li>
                                <li class="nav-item">
                                    <a class="nav-link" href="/about">About Us</a>
                                </li>
                                <li class="nav-item dropdown">
                                    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"
                                        data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                                        Goat School
                                    </a>
                                    <div class="dropdown-menu" aria-labelledby="navbarDropdown">
                                        <a class="dropdown-item" href="#">Action</a>
                                        <a class="dropdown-item" href="#">Another action</a>
                                        <a class="dropdown-item" href="#">Something else here</a>
                                    </div>
                                </li>
                                <li class="nav-item dropdown">
                                    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"
                                        data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                                        Adopt a Goat
                                    </a>
                                    <div class="dropdown-menu" aria-labelledby="navbarDropdown">
                                        <a class="dropdown-item" href="#">Action</a>
                                        <a class="dropdown-item" href="#">Another action</a>
                                        <a class="dropdown-item" href="#">Something else here</a>
                                    </div>
                                </li>
                            </ul>
                            <ul class="navbar-nav">
                                <li class="nav-item dropdown">
                                    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"
                                        data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                                        Account
                                    </a>
                                    <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">
                                        <a class="dropdown-item" href="#">Login</a>
                                    </div>
                                </li>
                            </ul>
                        </div>
                    </nav>
                </div>
            </div>
        </div>
    </header>

    <section class="jumbotron text-center">
        <h1 class="jumbotron-heading">The Goat Pasture</h1>
        <p class="lead text-muted">Love goats? Want a goat? We're here for you!</p>
        <p><img class="img-fluid" src="../course/img/goat1.jpg"></p>
        <p>
            <a href="/school" class="btn btn-secondary my-2 mr-2">Learn about Goats</a>
            <a href="/adopt" class="btn btn-secondary my-2">Adopt a Goat</a>
        </p>
    </section>

</body>

</html>
```

Yikes!
That's some deep HTML.

But just follow the examples on the Bootstrap web site and you'll be fine…

and reap the benefits…

Example

- (Includes Bootstrap and Popper JS)

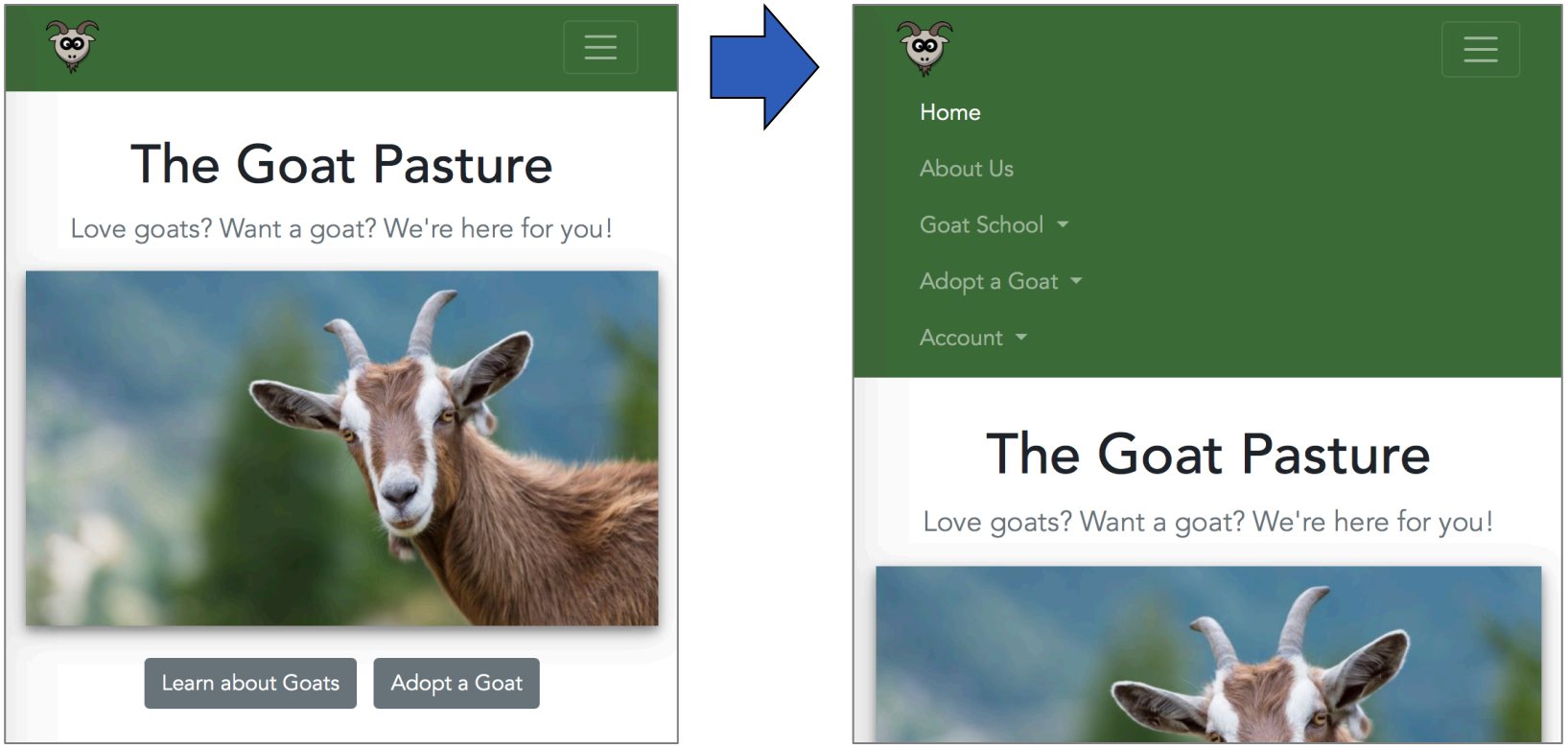
```
<!DOCTYPE html>
<html>

<head>
    <title>The Goat Pasture - Home</title>
    <link href="../course/lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="style5.css">
    <script src="../course/lib/jquery/jquery-3.3.1.min.js"></script>
    <script src="../course/lib/popper/popper.min.js"></script>
    <script src="../course/lib/bootstrap/js/bootstrap.min.js"></script>
</head>

<body>
   ...
```

Example

- Easy brand icon, submenus, right-justification, …

Example

- Plus responsive menu layouts at small screen sizes



Remember Our Stack?

- User Interface: **HTML, CSS, JavaScript, jQuery**
  - The essential components of web development
- Styling/Interaction: **Bootstrap**
  - Concepts transfer well to others (e.g., Pure)
- Web Framework: **Flask**
  - Simpler than many alternatives, but still very powerful
  - Concepts transfer well to others (e.g., Express, Django)
- Database: **SQLite**
  - Simpler than many alternatives, but still very powerful
  - Concepts transfer well to others (e.g., MySQL, etc.)
- Platform: **sandbox.cci.drexel.edu**

Remember Our Stack?

- User Interface: **HTML, CSS, JavaScript, jQue**
  - The essential components of web development
- Styling/Interaction: **Bootstrap**
  - Concepts transfer well to others (e.g., Pure)
- Web Framework: **Flask**
  - Simpler than many alternatives, but still very po
  - Concepts transfer well to others (e.g., Express, D
- Database: **SQLite**
  - Simpler than many alternatives, but still very po
  - Concepts transfer well to others (e.g., MySQL, et
- Platform: **sandbox.cci.drexel.edu**

Done for now, will revisit later (e.g., AJAX)

Cover only essentials, later

Will wait until later in the course

That's it for now!

- That's our very quick tour of a more-or-less complete stack for web development
- There are many, many, many more pieces to each of these components — we've only touched the surface
- We'll continue to flesh out more pieces this term, in the context of larger issues
  - E.g., Flask, AJAX, database access, graphics, possibly user authentication and cookies