Project Tutorial 2a

Introduction to Web Technologies

Dr. Yuzuko Nakamura

Dept. of Computer Science

Y.Nakamura@ucl.ac.uk

Goals

- For you to understand everything happening in the starter code.
  - But not yet begin to change it.
- Explain basics of web application development.

HTML

- Hypertext Markup Language.
  - The format in which webpages are written and displayed by web browsers.
- Originally defined by Tim Berners-Lee and now maintained and developed by the World Wide Web Consortium (W3C).
  - http://www.w3.org/
- Uses tags to ‘mark up’ the content of web pages to specify the content structure.
  - e.g. <p>This is a paragraph</p>
  - Same concept as XML (the extensible markup language), which is used to mark up data for any user-defined purpose.

Basic HTML page structure

```
<!DOCTYPE html>
<html>
<head>
   <title>Page Title</title>
</head>
<body>
  <h1>Header</h1>
  <p>A paragraph with a <a
href="http://www.cs.ucl.ac.uk">Link</a></p>
</body>
</html>
```

Basic HTML page structure

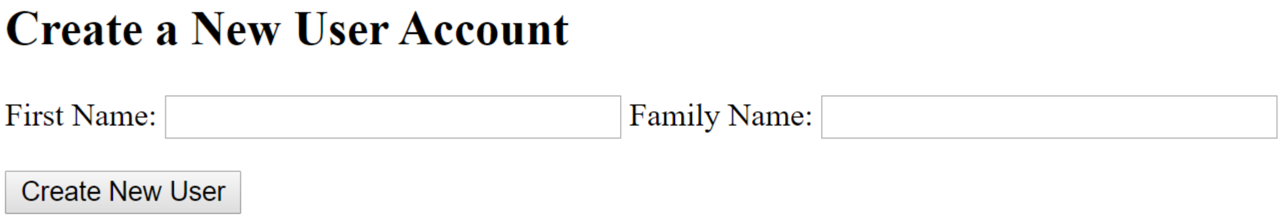
```
<!DOCTYPE html>
<html>
<head>
   <title>Page Title</title>
</head>
<body>
  <h1>Header</h1>
  <p>A paragraph with a <a
href="http://www.cs.ucl.ac.uk">Link</a></p>
</body>
</html>
```

HTTP(S)

- Hypertext Transfer Protocol.
- Request-Response protocol:
  - Browser sends an HTTP request to a server and gets back an HTTP response.
  - Stateless: each request independent from the previous/next.
  - Protocol specifies format of these messages.
- HTTPS is the secure version.
  - Uses Transport Layer Security (TLS) and Secure Sockets Layer (SSL) for a secure connection & secure communication.

HTML forms

- Specifies elements like text fields and buttons for inputting data



Create a New User Account

First Name: Family Name:

Form example

Form surrounded by <form> tags

Various input types exist (button, checkbox, radio, etc.)

```
<h2>Create a New User Account</h2>

<form method='post' name='newUserForm'>
  <label>First Name:</label>
  <input type='text' name='firstName' size='30'>
  <label>Family Name:</label>
  <input type='text' name='familyName' size='30'>

  <p><input type='submit' value='Create New
User'></p>
</form>
```

"Submit" input has special function

GET vs. POST methods

- Two different ways of sending information from one page to another.
- GET: puts data in URL. Ex:
  - browse.php?viewid=20&sort=true
    - This sets "viewid" and "sort" variables.
  - Publicly visible (it's in the URL).
- POST: includes data in HTTP request to specified page.
  - Data only visible in HTTP communication.
  - When you click the back/refresh button and the site asks if you want to resend the data, it was using POST.

CSS

- Cascading Style Sheets.
- Used to specify the presentation (look and format) of an HTML document (web page).
  - Allows separation of presentation from content (good practice).
- Specifies a language for describing styles by setting attributes of HTML elements (color, size, margins, etc.).

CSS example

```
h1 {
  font-family: Georgia, "Times New Roman", Times, serif;
  font-size: 30px;
  color: #00ff66;
}

p {
  font-weight: bold;
  line-height: 1.3em;
  border-style: solid;
}
```

Three ways of including CSS styling

- Within HTML tags. Ex:
  `<h1 style="font-size: 30px; color: #0f6">`
- In HTML `<head>`, in between `<style> </style>` tags.
- Loaded from external file. In `<head>`:
  ```
  <link rel="stylesheet" href="css/my
  stylesheet.css">
  ```

CSS classes and ids

- In addition to styling HTML tags (e.g. <p>), can also style a subset of page elements using **class** and **id** (unique).

In HTML file:

```
<div class="myclass">...</div>
<div id="special">...</div>
```

In CSS:

```
.myclass {
}

#special {
}
```

Bootstrap

- Bootstrap is a CSS library with many predefined styles.
  - Automatically handles resizing for different devices / screen sizes.
- Optionally makes use of Jquery JavaScript functions.
- Use predefined styles by adding `class="[bootstrap class]"` to your HTML tags.

Getting started with Bootstrap

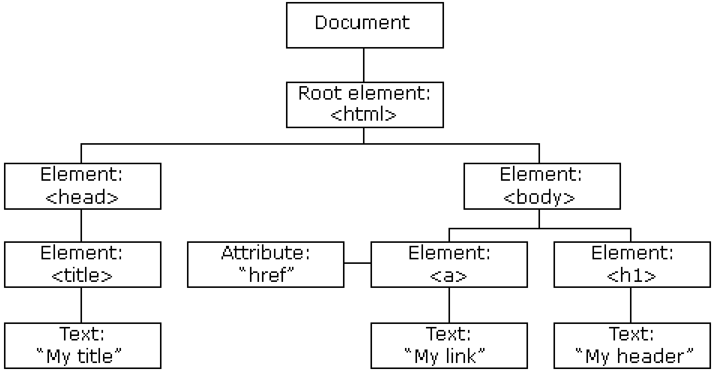
- W3Schools tutorials: https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp
- Documentation: https://getbootstrap.com/docs/4.5/getting-started/introduction/

JavaScript

- A client-side scripting language (run by user’s web browser, not the website server).
  - Possible for user to turn off JavaScript features.
- A powerful dynamic object-oriented language.
  - Not Java! C-style syntax and similar to Python in some ways.
- Good for interactivity: can perform actions in response to events (clicking a link, button, typing text, etc.).
- JavaScript libraries useful for simplifying programming.
  - JQuery, Node.js, etc.

The Document Object Model (DOM)

- The browser parses HTML to create an object-based tree structure called the DOM.
- JavaScript can access the DOM to manipulate its structure and content.
  - Dynamically modify the web page.
    - E.g., hide/show sections, move elements around, create new sections



Document

Root element: <html>

Element: <head>

Element: <title>

Text: "My title"

Element: <body>

Attribute: "href"

Element: <a>

Text: "My link"

Element: <h1>

Text: "My header"

Image from w3schools.com

PHP

- PHP: Hypertext Processor (originally: Personal Home Page).
- An imperative, object-oriented programming or scripting language geared toward creating dynamic websites.
  - C/Java style syntax; procedural and OO programming.
  - Controversial language design but high utility.
  - Extensive range of libraries (see www.php.net).
- A server-side language.
  - Code is run on the server. No PHP code is delivered to or run on the browser.

Mixed PHP/HTML structure

- PHP code enclosed in non-HTML <?php … ?> tags.
- Can have files containing only PHP code, or HTML interspersed with PHP code.
- Save any file containing PHP code as .php.
- These files must be requested and processed through your web server running PHP.

HTML + PHP webpage example

```
<html>
  <head>
    <title>Example Document</title>
  </head>
  <body>
    <h1>A Title</h1>
    <?php
      $date = date("l, F d, Y h:i", time());
      echo "<p>Today is $date</p>";
    ?>
  </body>
</html>
```

PHP syntax basics

- Semicolons are required at ends of statements.
- Variables are prefixed with '$'
  - No type declaration. Ex: $myvar = 10;
- See http://www.php.net/manual/en/ for the full PHP manual.

Functions in PHP

- Code can be structured by declaring functions (methods/procedures):

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

print factorial(5);
```

- Similar to Python

PHP and forms

```
<h2>Create a New User Account</h2>

<form method='post' action='createNewUser.php'
name='newUserForm’>
  <label>First Name:</label>
  <input type='text' name='firstName' size='30'>
  <label>Family Name:</label>
  <input type='text' name='familyName' size='30'>

  <p><input type='submit' value='Create New
User'></p>
</form>
```

Accessing form data

- The file createNewUser.php could contain:

```
<?php
   var_dump($_POST); echo "<br />";
   $user = array();
   $user["firstName"] = $_POST["firstName"];
   $user["familyName"] = $_POST["familyName"];
   var_dump($user);
?>
```

- var_dump is a function provided to “dump” a variable value.
- The form data is accessible in the ‘super global’ array $_POST (or $_GET).

The web server

- A server is a continually-running process that waits for messages to be received on a network port.
- WAMP uses the Apache server (open source - www.apache.org).
  - By default: uses port 80 in production, 8080 for development.
- Routes message contents to a web container running on the server.
- Web container handles the message and returns a response.
- Web server returns response back to caller.

Web application

- Examples:
  - Java EE Platform (Servlet container),
  - .Net (C#),
  - PHP,
  - Grails (Groovy),
  - Rails (Ruby),
  - Django (Python).
- Interfaces with database and serves up dynamic (changing) HTML pages.

Summary

- Pages will have static HTML, CSS, JavaScript content.
- PHP will be used to create dynamic content and connect to the MySQL database.
- Forms are needed to collect information/actions from user.