Csci 4131 Internet Programming
Fall 2021
Lecture 4
September 20th

Instructor: Dr. Dan Challou

Logistics – Csci 4131 Lecture 4, September 20

- HW Assignment 2 out, available in week 2 modules on the Homepage of the class Canvas site (and in the assignments section). Due this coming **Friday September 24** at 11:59pm

- ***Weekly readings and exercises are in your zybook*** and additional readings, tutorials, programming homework due dates and exam dates are in item:
  - **Course Schedule: Weekly Class Readings and Tutorials, Exam Dates, and Programming Assignment Due Dates**

  in the Resources Module at the top of the Home

  Page on the class Canvas site

Currently working on the following and tutorials, but feel free to read ahead!!!

Zybook Lecture 5 preparation HW 3, +
www.w3schools.com –
 CSS Tutorial, JavaScript Tutorials:
https://www.w3schools.com/css/default.asp
http://www.w3schools.com/js/

Optional Reading, Sebesta Chapters 3,4

Questions?

Agenda

- Last Time:
  - Lists
  - Tables
  - Introduction to Forms
    - HTML input tag
  - HTTP Overview
- Today:
  - HTTP revisited
  - CSS revisited

Recall, from last lecture (and homework exercises and tutorials) HTML Forms

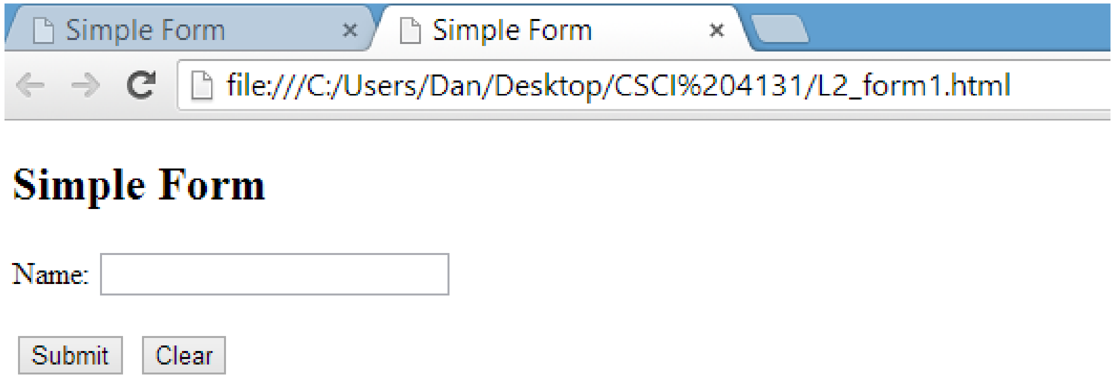
- HTML5 provides **forms** for collecting information from users
- HTML forms enable you to do some syntactic validation on the client side before sending the information collected from the user to the server side (your zybook, and http://www.w3schools.com/ for more details)
- A form typically has a collection of input fields that can be submitted for further processing

Input Types

- A form is typically used to gather input from users.
- HTML 5 provides many different types of input tags for gathering the data from the user
  - https://www.w3schools.com/tags/tag_input.asp

- When a user submits a form (usually by clicking on the Submit button – implemented by a Submit input tag), the browser gathers the data entered into the form, packages it into an HTTP request method (either a GET or POST) and send the message it off to its destination (specified in the action field)

Review Lecture 3, Exercise 1 – Add an email input type and URL input type to the Simple Form



Simple Form

Name:

Add an email input type, and a URL input type to the Simple Form above.

An answer:
L3_exercise1.html

Questions?

HW 2 Demo

- Questions?

Foundations of the World Wide Web Revisited (Components of HTTP)

Components of a Uniform Resource Locator (URL)

Consider the following URL:

- https://twin-cities.umn.edu/about-us
- PROTOCOL: **https://** indicates that the Secure HyperText Transfer Protocol (HTTPS) should be used to obtain the resource.
- Next in the URL is the server's fully qualified hostname (for example, twin-cities.umn.edu)—the name of the web-server computer on which the resource resides.
- This computer is referred to as the host, because it houses and maintains resources.
- The hostname twin-cities.umn.edu is translated into an IP (Internet Protocol) address—a numerical value that uniquely identifies the server on the Internet
- An Internet Domain Name System (DNS) server maintains a database of hostnames and their corresponding IP addresses and performs the translations automatically.

Recall: Components of a URL

Consider the following URL:

- https://twin-cities.umn.edu/about-us
  - Protocol
  - Hostname
  - The remainder of the URL (**/about-us**)

    specifies the resource's location (**/about-us**)
    and name (which is not present in our example)!!!
    If the resource name is not specified in the URL, the server uses
    the default name: **index.html**

HTTP **GET** and **POST** Requests

HTTP: GET

- https://www.bing.com/search?q=challou&form=QBLH&sp=-1&pq=challou&sc=
- **A GET request appends the data it is sending to the URL, e.g.,**
- **`www.bing.com/search?q=challou.`**

- In this case, ***search*** is the name of the routine on the server side, ***q*** is the name of a variable in
- bing's search form and ***challou*** is the search term.

- The ? in the preceding URL separates the query string from the rest of the URL in a request.

- A *name/value* pair is passed to the server with the ***name*** and the ***value*** separated by an equals sign (**=**).

- If more than one *name*/*value* pair is submitted, each pair can be separated by an ampersand (&).
- **E.g., www.bing.com/search?q=challou& …**
- ***Or a + sign***
- **E.g, www.bing.com/search?q=challou+ …**

- The HTTP server uses data passed in a query string to construct an appropriate HTTP response message

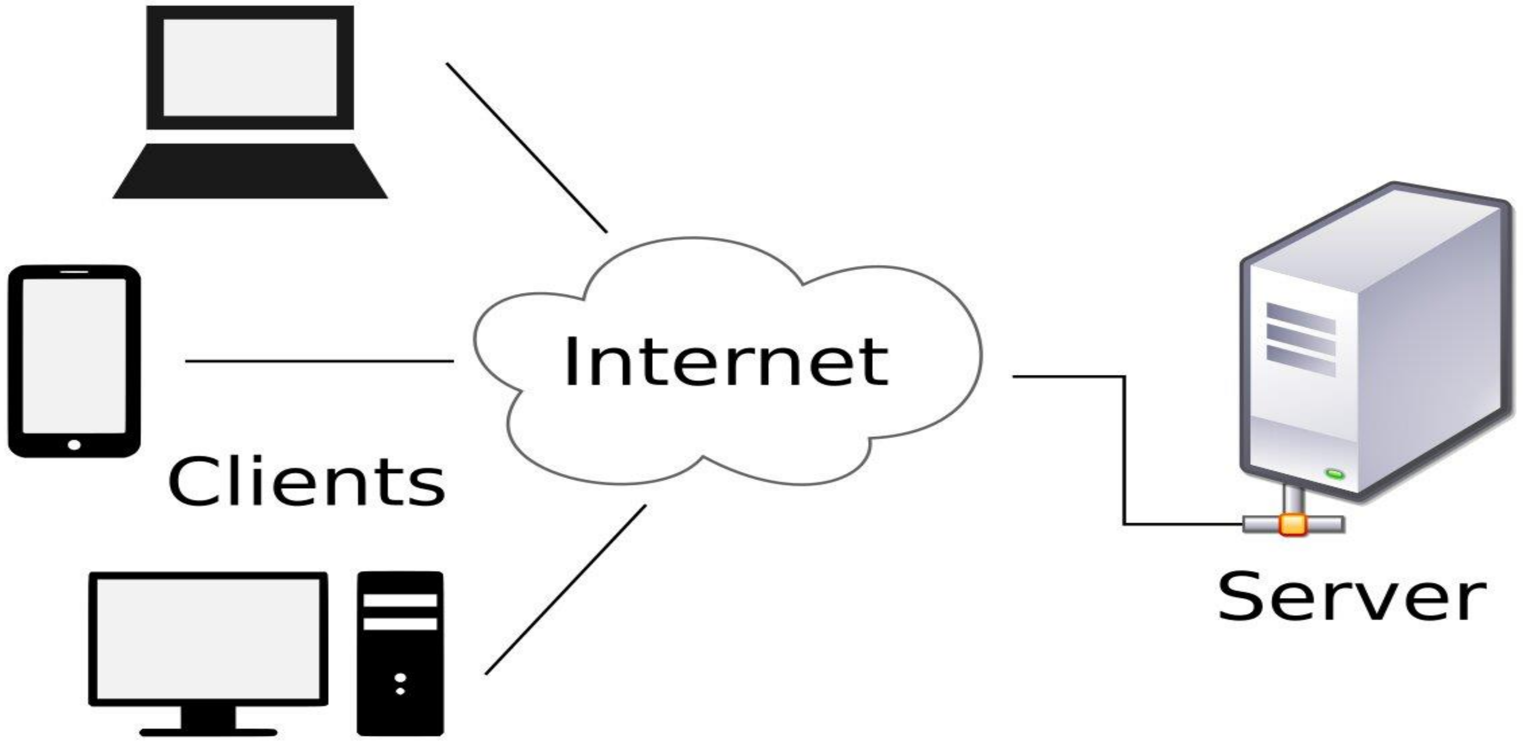
- The HTTP server then sends a response to the client.

- A get request may be initiated by submitting an HTML form whose method attribute is set to "get", or by typing the URL (possibly containing a query string) directly into the browser's address bar.

HTTP: POST

- A POST request sends form data as part of the HTTP message, not as part of the URL.
- A GET request typically limits the query string (i.e., everything to the right of the ?) to a specific number of characters, so it’s often necessary to send large amounts of information using the post method.
- The POST method is also sometimes preferred because it hides the submitted data from the user by embedding it in an HTTP message.
- If a form submits several hidden input values along with user-submitted data, the POST method might generate a URL like www.searchengine.com/search.
- The form data still reaches the server and is processed in a similar fashion to a GET request, but the user can’t see the exact information sent in the address/URL bar.

Issuing a URL from a Web-Browser is an Application of the Client / Server Model of Computing



https://en.wikipedia.org/wiki/Client%E2%80%93server_model

Questions?

On To Style

Hopefully, you have a good handle on structure (HTML) !
on to the second of the 4 components of a Web Page:

Structure (HTML)

Style (Cascading Style Sheets- (CSS))

Behavior (JavaScript – to change – e.g. - add, delete, update web page structure, style)

DOM (the data structure behind each web page)

Methods for Adding Style (CSS) to HTML 5 documents

- Inline – applied via the HTML style attribute to a particular element
- Embedded – use HTML style tag:

  <style> … </style>

  in the <head> section of the HTMLdocument.
- Define styles for particular elements, and classes that can be applied to elements in between the style start and end tag
- Separate File that is included in HTML (separates structure and content from presentation)

Use the HTML <link> tag to include a style file

CSS Inline

- The style attribute
- <p style ="font-size:32px;color:red;text-align:center"> Here is a paragraph <br> with a line break </p>

css_inline_ex.html

Embedded CSS using HTML Style Tag

https://www.w3schools.com/html/html_css.asp

Source: http://www.w3schools.com/tags/tag_style.asp

Including the CSS properties from an External Style Sheet

```
<head>
<link rel="stylesheet" type="text/css" href= "style.css" >
</head>
```

What can you style with CSS???

- Virtually any HTML element, and any property
  - Font
  - Visibility
  - Font-size
  - Color
  - Background
  - Add animation
  - Reaction to events
- And, the box that wraps around every HTML element (margin, border, padding, content)
  - https://www.w3schools.com/css/css_boxmodel.asp

Units of measure for styling (fonts, the box, and anything else you can think of)

- https://www.w3schools.com/cssref/css_units.asp

Units of Measure - Example

- measure_ex.html

CSS Rule Sets That Select by Element, Type, Id, and Class

```
/* Element Type ID and Class Selector Examples */

        /* All Elements */
        *  {margin: .5em; 1em;}

        /*Elements by Type */
        h1{ font-family: Arial, sans-serif, serif;}

        /*One Element by ID */
        #main{
                border: 2px solid red;
                padding: 1em;
                }

        /* Elements by Class */
        .blue  {color: blue;}

        .right {text-align: right;}
```

Example

```
<head>
   <meta charset = "utf-8">
   <title>Element Type ID and Class</title>

   <!-- this begins the style sheet section -->
   <link rel="stylesheet" type="text/css" href="mystyle4.css">

 </head>
 <body>
            <header>
                        <h1> This Week At The University of Minnesota</h1>
            </header>

            <section id="main">
                        <h1>Events</h1>

                        <p class="blue">Music, Sports, and lots more, just check the event calendars!
                        </p>

                        <p class="blue right">
                        There is another week of classes...
                        </p>
            </section>
 </body>
```

Elem_type_id_class.html

Can code multiple selectors in your embedded css or external style file

```
      h2 {color: green;}
/* Multiple Selectors */
      h3,h4 { color: maroon;}

/* All elements with href attributes */
      *[href] { font-size: 95%;}

/* All <a> elements with href attributes */
      a[href] {font-family: Arial, sans-serif;}
```

```
<!DOCTYPE html>
<!-- External style sheets. -->
<html>
  <head>
    <meta charset = "utf-8">
    <title>Multiple Selectors</title>
    <!-- this begins the style sheet section -->
    <link rel="stylesheet" type="text/css" href="mystyle5.css">
  </head>
  <body>
                <header>
                                <h1> This Week At The University of Minnesota</h1>
                </header>

                <section id="main">
                                <h1>Events</h1>
                                <p class="blue">Music, Sports, and lots more, just check the event calendars!
                                 </p>
                                <p class="blue right">
                                There are another 12 weeks of classes...
                                </p>
                </section>
                <h2> Here are some examples of multiple selectors, etc. </h2>
                <h3> The quick brown fox </h3>
                <h4> jumped over the lazy dog </h4>
                <a href = "http://www.google.com">Click here for Google</a>
  </body>
</html>
```

Multiple_selectors_ex.html

A CSS File for Styling the BOX

```
                /* A CSS Style file for formatting "the box" around some block elements  */


body {
                border: 3px dotted black;
                margin: 10px;
                }

section{
                border: 2px solid black;
                width: 500px;
                margin: 20px;                /* all four sides */
                padding: 10px;  /* all four sides */
                }
h1,p {
                border: 1px dashed black;
                padding: 10px;
                }
h1{
                margin: .5em 0 .25em; /* .5em top,0 right and left, .25em bottom */
                padding-left: 15px;
}
p{
                margin: 0; /* all four sides */
                padding-left: 15px;
}
```

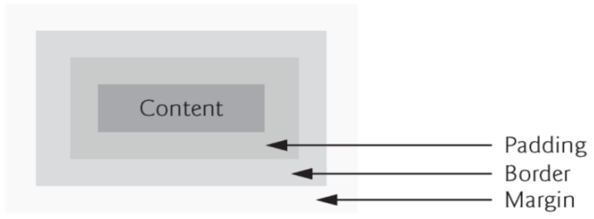


Fig. 4.13 | Box model for block-level elements.

HTML File That Uses our CSS Block Element Style File

```
<!DOCTYPE html>

<!-- Box Model Example -->
<html>
  <head>
    <meta charset = "utf-8">
    <title>Box Model Example</title>

    <!-- this begins the style sheet section -->
    <link rel="stylesheet" type="text/css" href="mystyle8.css">

  </head>
  <body>
            <section>
                        <h1>The University of Minnesota</h1>
                        <p>Educating the Leaders of Tomorrow for over 100 years.
                           There is always something happening at the U!</p>
            </section>

  </body>
</html>
```

Box example

Drawbacks of our Approach?

- Not Responsive Web Design
  - https://www.w3schools.com/html/html_responsive.asp

Lecture 4, Exercise 1:

- Update the CSS file that I just did so it uses relative styling that enables the previous example seamlessly scales to the window size

  Rel_Box_Example.html

- You can use your zybook and or w3schools, and your phone or computer **for reference**

- **You can download the files: Box_Example.html and mystyle8.css from the Week 3 module on the class Canvas site**

- One key definition 1em (a relative unit of measure) = 16px (an absolute unit of measure) in most browsers

- Note, there is one styling command that you must set to a % to make sure the relative styling works – part of your task is to find it and fix it!!!! (Hint see: https://www.w3schools.com/html/html_responsive.asp)

- **Submit Exercise via the Lecture 4, Exercise 1 link on Canvas – *just the redone css file mystyle8***
- ***Please close your computer (at least a bit) when you are done!***

Questions?

A Short Overview of the Document Object Model (DOM)

- We've seen HTML and CSS
- Before we dig into JavaScript, we should know something about the DOM
  - Why?
  - DOM scripting lets you use JavaScript to update a web page in response to user actions by changing the DOM

- The Document Object Model (DOM)is an internal representation of HTML elements in a Web Page
- As an HTML page is loaded by the web browser, the DOM for that page is created in the browser's memory

Next time

- The DOM Revisited
- JavaScript
- Event Handling