

# Welcome

Basics of the Web + HTML

# Basics of the Web

At its core, we can generalize the web as having two sides:

Client vs Server

For Software? Hardware?

# Basics of the Web: IP Addresses

What is it?

# Basics of the Web: IP Addresses

What is it?

*Unique address for a machine on internet*

Where do we get it from?

# Basics of the Web: IP Addresses

What is it?

*Unique address for a machine on internet*

Where do we get it from?

*Your Internet Service Provider (ISP) when connecting to the internet*

# IP Protocols- IPv4 and IPv6

These protocols define how data is sent between computers over packet-switched network

## IPv4:

- 32-bit unsigned integer: 128.8.128.8
- Domain name: cs.umd.edu
- localhost: 127.0.0.1

## IPv6:

- 128 bit addresses
- Replaces IPv4
- How many possibilities do we have now?

# Basics of the Web: Web Servers

What are they?

*A computer program that delivers web pages*

What are some examples?

*Apache <http://www.apache.org/>*

*IIS Internet Information Services*

*Sun Java System Web Server*

# Basics of the Web: Web Servers pt.2

You can install and run a web server in your computer

<https://www.apachefriends.org/>

Local address: <http://localhost> or <http://127.0.0.1/>

In other words all you need is:

- IP address
- Connection to the internet
- Computer with web server software



# Basics of the Web: DNS

## **DNS** Domain Name Systems

Protocol for translating domain names to IP addresses

*Example: cs.umd.edu → 128.8.128.44*

Multiple DNS servers on internet

DNS server may need to query other DNS servers

*edu DNS server queries umd.edu server to find  
cs.umd.edu*

# Basics of the Web: URLs

URL – Uniform Resource Locator

Represents web resource

- Arbitrary files
- Web pages

Examples

- <http://www.cs.umd.edu/index.html>
- <ftp://ftp.funet.fi/pub/standards/RFC/rfc959.txt>
- <https://login.yahoo.com/>
- <file://dir/my.txt>

# Basics of the Web: URL Structure

A URL consists of:

- A Protocol
  - Http
  - Ftp
  - Https
  - File
  - ...
- IP Address
- Port (usually left empty)
- Path

For this class we will be using Chrome

Let's look at some things we can do....

# HTTP

Hypertext Transfer Protocol (HTTP) protocol that defines how user agents (e.g., browser) and web server can communicate

HTTP is a request/response protocol between clients and servers

Some methods (operations) defined as part of the protocol

- GET Use to download a resource (e.g., image, web page). Most common method used
- HEAD Returns only the header
- POST Submits data (e.g., form data) to the server

# HTTP Request Cycle

A typical HTTP request / response cycle:

- The browser requests an HTML page. The server returns an HTML file.
- The browser requests a style sheet. The server returns a CSS file.
- The browser requests an JPG image. The server returns a JPG file.
- The browser requests JavaScript code. The server returns a JS file
- The browser requests data. The server returns data (in XML or JSON).

# HTML

- The Language used to define webpages
- HTML is what the server sends to the browser (in traditional web pages/applications - more on this later)
- Renders the pages based on the tags
  - May require other things from the server to fully load

# What do our editors do?

Example editors:

- Atom
- Notepad++
- Sublime
- **Visual Code**
- Komodo Edit
- Eclipse
- JetBrains



# Creating Web Pages

HTML - HyperText Markup Language

HTML Standard

- Developed by the World Wide Web Consortium (W3C)
- <http://www.w3.org>

Document is described through a series of commands and directives present in a text file

## Creating Web Pages Pt. 2

HTML goal is to describe structure only. Presentation should be left to cascading style sheets

When interpreted by an HTML viewer those commands determine the appearance of the page

HTML documents are entirely ASCII text

Commands are explicitly inserted

# Your Bestest Friends

W3 Schools: <http://w3schools.com/>

Mozilla Web Docs: <https://developer.mozilla.org/>

# Basics

Tags dictate the webpage, here are the non-negotiable ones:

```
<!DOCTYPE html>
```

```
<html>
```

```
<!-- Other tags go here, also this is a comment -->
```

```
</html>
```

# Head Tag

Can be used in conjunction with inner tags:

- `<title> </title>`
  - It is required
  - Search engines depend on it
  - Provide a meaningful name as it is bookmarked
- `<meta>`
  - Document's metadata (e.g., keywords, description, encoding, etc.)
- `<link>`
  - Specifies relationship between document and resource
- `<style>` and `<script>`
  - To define CSS style in the document or JavaScript

# Lists

Unordered list:

```
<ul>
```

```
  <li>item1</li>
```

```
  <li>item2</li>
```

```
</ul>
```

Ordered list:

```
<ol>
```

```
  <li>item1</li>
```

```
  <li>item2</li>
```

```
</ol>
```

# Not impressed? Me neither

Attributes allow us to customize our tags to what we want:

```
<ELEM ATTR= "value">Text</ELEM>
```

```
<ol type = "A">Ordered element with letters</ol>
```

Some attributes don't need a value, if they are a boolean.

# Tables

```
<table>
```

```
  <tr> <th>Header</th> </tr>
```

```
  <tr> <td>the data</td></tr>
```

```
  <tr> <td>the data row 2</td></tr>
```

```
</table>
```



# Images

<img> tag used for image inclusion

Attributes:

- src
- width
- height
- alt

To reduce the image size without losing proportions specify either the width or height and the browser will compute the other

# Links

Links are created using the `<a>` tag

- `<a href="http://www.cnn.com">CNN Page</a>`
- Need to specify `http://`

You can also link a page with another as follows:

`<a href="page.html">`

You can create links to elements in the same document using the `id` attribute

# Validation

<http://validator.w3.org/>

# Last thing!

You have free space to host webpages using terpconnect!

[terpconnect.umd.edu](http://terpconnect.umd.edu)

# WTWAW (What To Walk Away With)

Basics of how the internet works:

- What is HTML?
- What are IP addresses?
- What are the parts to a URL?
- What is HTTP?
- Basic HTML tags

**If your git doesn't work go to office hours!**