IS 2560: Web Tech & Standards Course Introduction

Graduate Program Information Science and Technology

School of Information Sciences

University of Pittsburgh

Alawya Alawami Ph.D.

Outline

- Course overview
- Logistics/Policies
- Overview/logistics
- WWW Overview
- HTTP Protocol
- Intro to HTML and CSS

About me

- Alawya Alawami PhD.
  - IT development consultant at Neurobehavioral medicine consultants
  - Visiting Lecturer at the University of Pittsburgh
  - Major: Information Science, Computer Science
- Research Interest
  - Web mining
  - Text mining
  - NLP

Tell Me about you

- Name
- Let us know what you like to go by
- Program
- Major
- Experience
- Interests/hobbies
- Career goals?

Course Overview

General info

- Web Technology and Standards
  - IS 2560
  - Monday 06:00-09:00 Room IS 404
- TA: TBD
- Instructor: Alawya Alawami
  - Class Website: CourseWeb
  - Personal Email: aaa221@pitt.edu
  - Office Hours: By Appointment/ one hour before class.

Course Expectation

- Prerequisites
  - Basic understanding of HTML
  - Ability to write code in C, C++, C# and/or Java
  - Some awareness of operating systems
- Preparation
  - Reading and thinking
  - Thinking and **“doing”** – beyond what I have “done”
- Engagement
  - Experimentation
  - Planning

Course Resources

- The textbook – it is to be read and digested prior to class
  - See Syllabus
  - **Weekly reading assignments**
- The lectures – they will go beyond and around the readings and assignments
- The web and manuals – there is more around than you will ever digest
- Your own experimentation and discussion with other students

Imperatives for 2560

- By the end of this course, you will be able to
- Work in a team to implement high-quality web sites that serve dynamic content to users in in various forms.
- Describe both the major technologies and design approach you used to implement your sites.
- Learn about common web application architecture and the technologies behind them
- Develop major component of web sites forms, scripts, shopping cart, etc.
- Comparing and contrasting your approaches with other options
- Understand and follow trends in the development of web technologies and standards
- Learn about other web technologies and solve problems on your own.

Policies

- The course will consist of 1 hour and 30 min
- Open discussion Students are expected to come to class prepared to discuss the topic assigned for the day.
- We will work on a hands-on problem related to the topic discussed.
- Bring your laptop to class to work on these problems/tutorials.
- You will be given a reading list each week. You are responsible for doing you reading before the class and prepare for the topic. This will also prepare you for the in-class tutorial.
- Absence: If for some reason, you missed the class, you need to work on the problem on your own and submit it by the deadline (No exceptions).
- In addition, students are expected to be doing their own exploratory reading on related subjects throughout the term.
- Knowledge and skills you take away from a course comes from **Your Effort**.
- **Exams may be added** to the requirements or substituted for selected requirements if in the instructor's opinion students are not staying abreast of course readings.
- **Plagiarism will not be tolerated (Cite, Cite,Cite).**
  - **For projects, attribute code to the original author and indicate clearly your contribution.**
  - **If the instructor suspects students are submitting non-original material, a grade of 0 will be submitted for the assignment and a failing grade for the course may be assigned**

Assignment Submission

- Through courseweb
- **Late assignments will be subject to a 10% penalty for each business day.**
- Attendance is mandatory and the grade for attendance will be counted toward your in-class activity. If you missed a class, you are required to work on the problem individually and submit it before the deadline.
- Submission: Each submission should
  - The assignment number or name, e.g. Project 2 or chat App
  - Source code
  - A link to your code in github
  - A live demo of your app uploaded to your pitt server, jsFiddle, cloud9 (the front-end link), plunker, codepen, AWS, your own hosting, as a github page, find a way to have a live demo.
  - (for inclass-activity and final project)Team member and their Pitt ID and their role, how did each member contribute to the project/activity (database designer, developer, interface designer etc.)
  - Personal Growth Report. What did you learn, where did you struggle, any great a-ha moments? Number of hours logged on this project/assignment
  - Any other feedback. How can we improve this assignment for future students.

Final Project (tentative)

- **Final project (more detailed later):**
- Completer web site that will integrate:
  - Data storage (MYSQL or MONGODB)
  - Multiple users with different interfaces (Admin view vs User view)
  - MVC structure
  - Admin capabilities
  - Location awareness
  - Session management capabilities
  - RESTful Web service
  - Ex: review site, social network site, blog, chat app

Communication

- All students are welcome to ask question any time during the lecture.
- During the class activities, I will be present to guide you through any difficulties you may have.
- Feel free to send emails or set up an appointment.
- I usually respond to emails within 24-48 hours, if you haven't received a response during that time, please resend your email. **Please make sure you add the course number to the email subject (INFSCI2560)**.

Grading

In-Class Activities	**30**
Final Project	30
Assignments	40

In Class Activities Grading				
Grade	**Symbol**	**Criteria**	**Minimum**	**Maximum**
Star plus	*+	exceeds expectations	**95**	**100**
Star	*	meets expectations	**80**	**94**
Star minus	*-	acceptable but does not completely meet all expectations	**70**	**80**
zero	0		**0**	**0**

A	100-90
B	80-89
C	60-79
F	0-59

Web Tech

- The core web technologies are three:
  - HTML
  - HTTP
  - URL
- These technologies depend more generally on internet networking technologies
  - NCP (~1970)
  - IP/TCP (1974)
  - DNS (1983)
- Web technologies are moving, generally driven by two technology developments
  - XML
  - CORBA/Web Services

Role of Standards

- Standards are often misunderstood
- There are many different kinds of standards
- Few people know how they actually come about
- IT standards are unique in many ways
- There are good and bad standards
- There are premature and delayed standards
- Standards are used by government and industry in a variety of ways

Credit

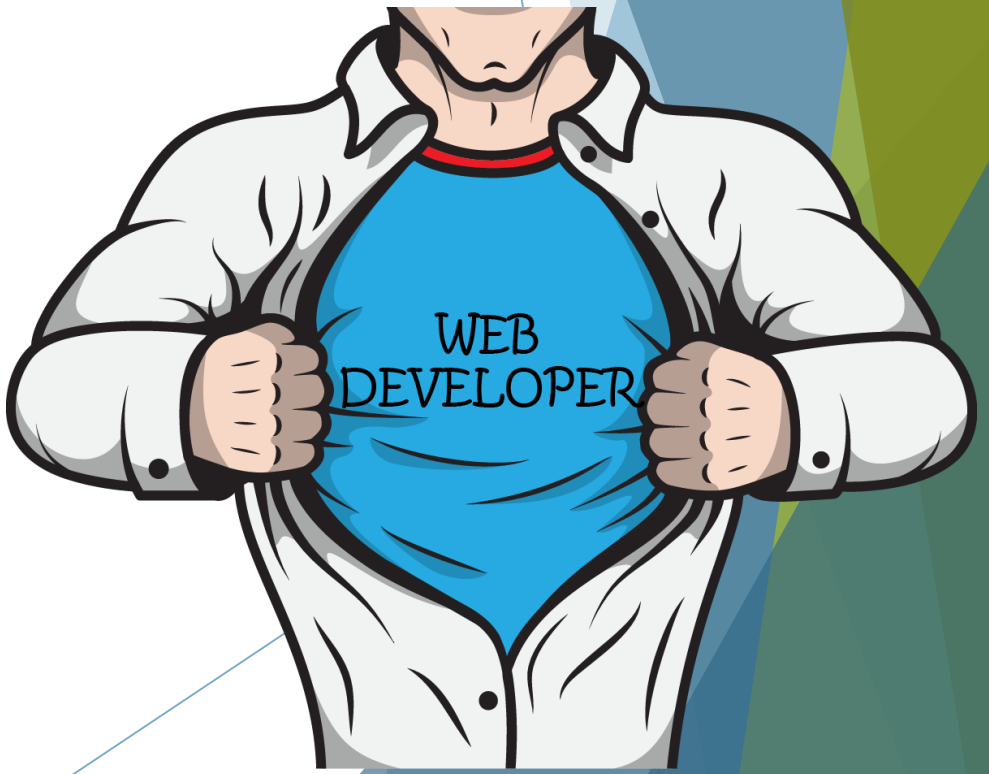
- Course Material Adapted in Part from Content created by:
  - Michael B Spring, University of Pittsburgh
  - David J Malan, Harvard University

Tools

- Browser (Chrome or Mozilla developer version)
- Node.js
- Angular JS
- MongoDB
- Bootstrap
- Netbeans (or Eclipse whatever you feel more comfortable with) with J2EE
- Notepad++ (or any text editor you prefer)
- FTP software (WinSCP or FileZilla)
- MYSQL
- You will need space in your laptop or PC (Free up some space, so all these software won't slow it down)

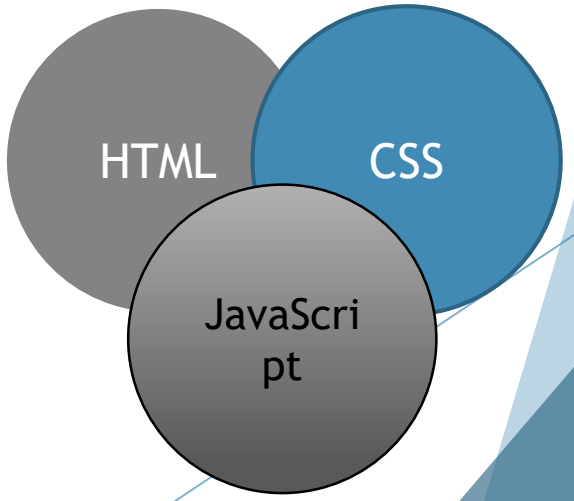
Web Developers

Front-End Developer / Designer
Back-End Developer / Web Programmer
Full Stack Developer



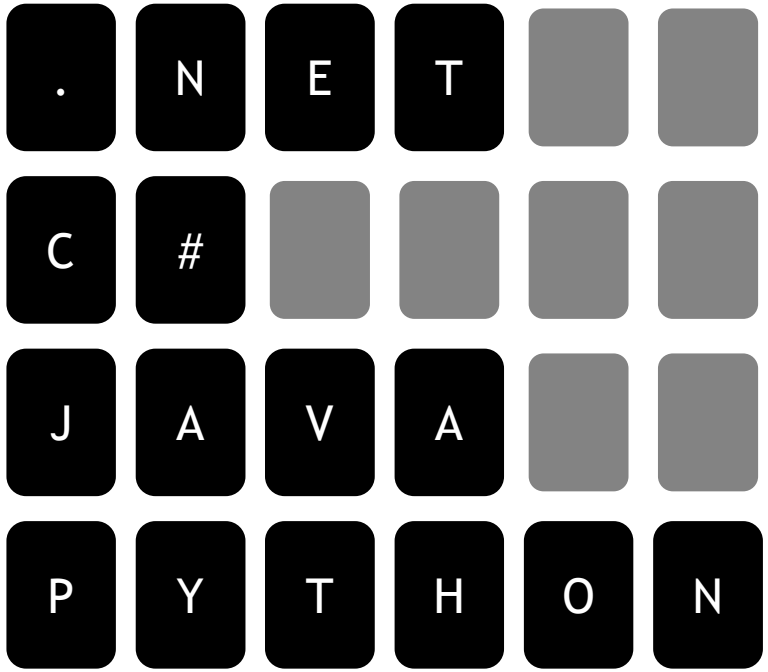
Front-End Developer

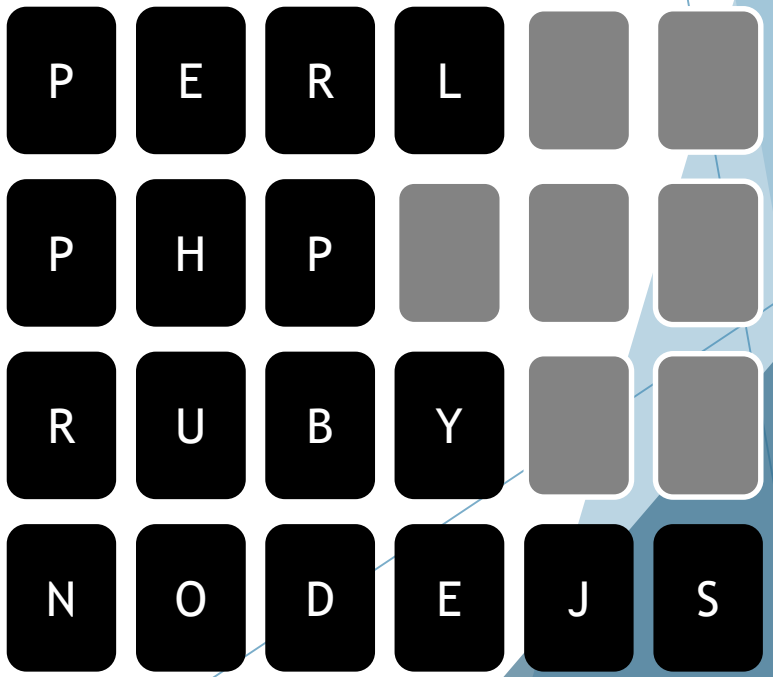
- Focuses on look and feel of a website
- Uses HTML, CSS, and JavaScript
- Is competent in color scheming, graphic design, and information flow
- Creates a great user experience
- Is right-brained: Strong intuition, creativity, & imagination



Back-End Developer

- Creates the inner workings of a website
- Is competent in programming languages (PHP, .NET, Python, C, Ruby)
- Is left-brained: logic, linear thinking, technical
- Hands-on coding experience is required





Full Stack Developer

- Comfortable working with both back-end and front-end technologies.
- Work with
  - Databases
  - PHP/Python/.Net/Ruby/Java
  - HTML
  - CSS
  - JavaScript
  - *and everything in between*

What is a Full Stack?

- **System Administration**
  - Shell Scripting (Linux)
  - Cloud Computing (Amazon, Rackspace)
  - Search Engine Integration (ElasticSearch, Sphinx)
  - Caching
  - Monitoring

What is a Full Stack?

- **Web Development Tools**
  - Version Control (Git, Mercurial)
  - Virtualization (VirtualBox, Vagrant, Docker)

What is a Full Stack?

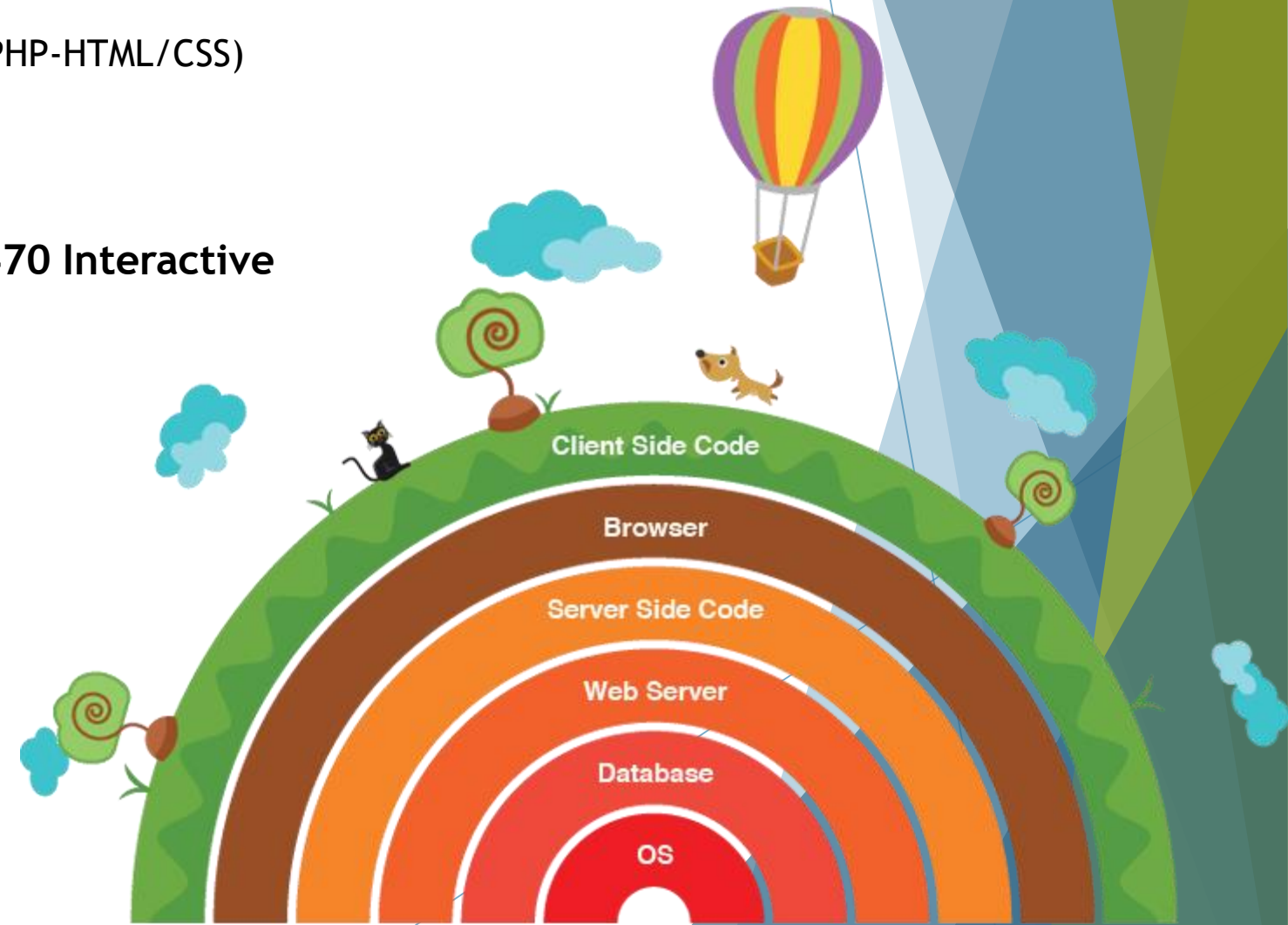
- **Back-end Technologies**
  - Web Servers (Apache, Nginx)
  - Programming Languages (PHP, NodeJS, Ruby)
  - Databases (MySQL, PostgreSQL, MongoDB, Redis) – General (SQL, JSON, XML)

What is a Full Stack?

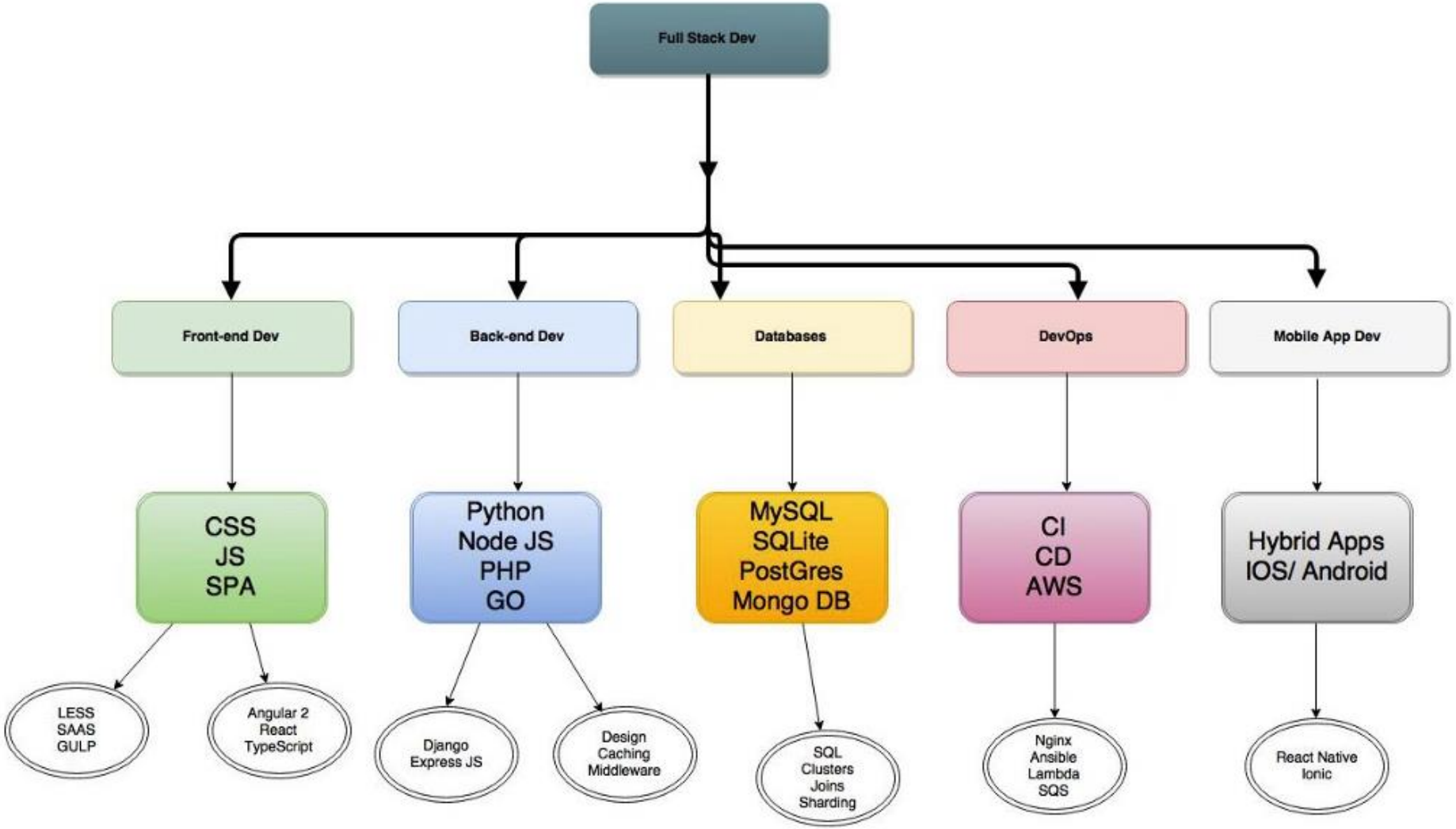
- **Front-end Technologies**
  - HTML / HTML5 Semantic Web
  - CSS / CSS3: LESS, SASS, Media Queries
  - JavaScript: JQuery, AngularJS, …
  - Browser Compatibility
  - Responsive Design
  - AJAX, JSON, XML, WebSocket

What is a Full Stack?

- **Design**
  - Convert design to front-end code (PHP-HTML/CSS)
  - User Interface (UI)
  - User Experience (UX)
- **UI and UX are covered in INFSCI2470 Interactive System Design**



Big picture


Full Stack Dev
Front-end Dev
Back-end Dev
Databases
DevOps
Mobile App Dev
CSS
JS
SPA
Python
Node JS
PHP
GO
MySQL
SQLite
PostGres
Mongo DB
CI
CD
AWS
Hybrid Apps
IOS/ Android
LESS
SAAS
GULP
Angular 2
React
TypeScript
Django
Express JS
Design
Caching
Middleware
SQL
Clusters
Joins
Sharding
Nginx
Ansible
Lambda
SQS
React Native
Ionic

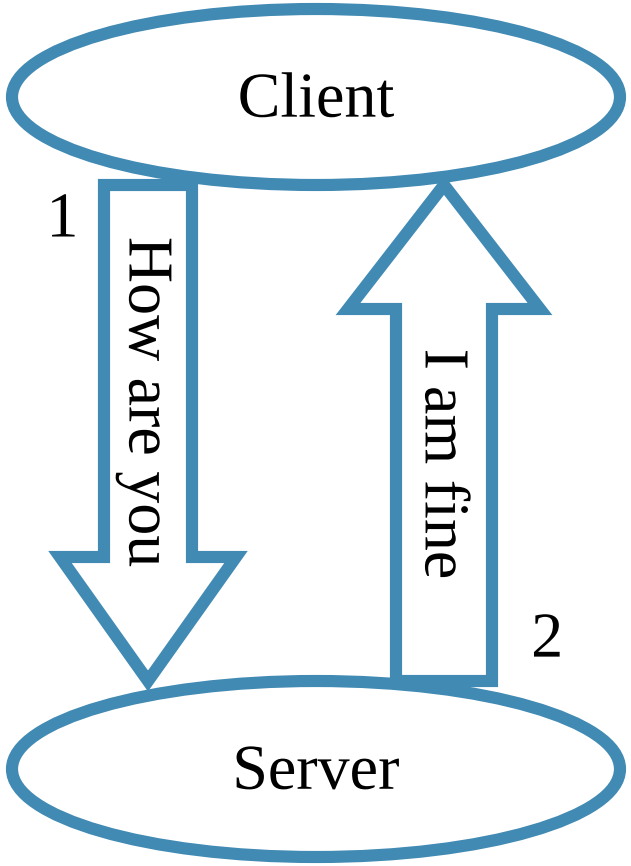
IS 2560: WWW Tecnologies

Graduate Program Information Science and Technology

School of Information Sciences

University of Pittsburgh

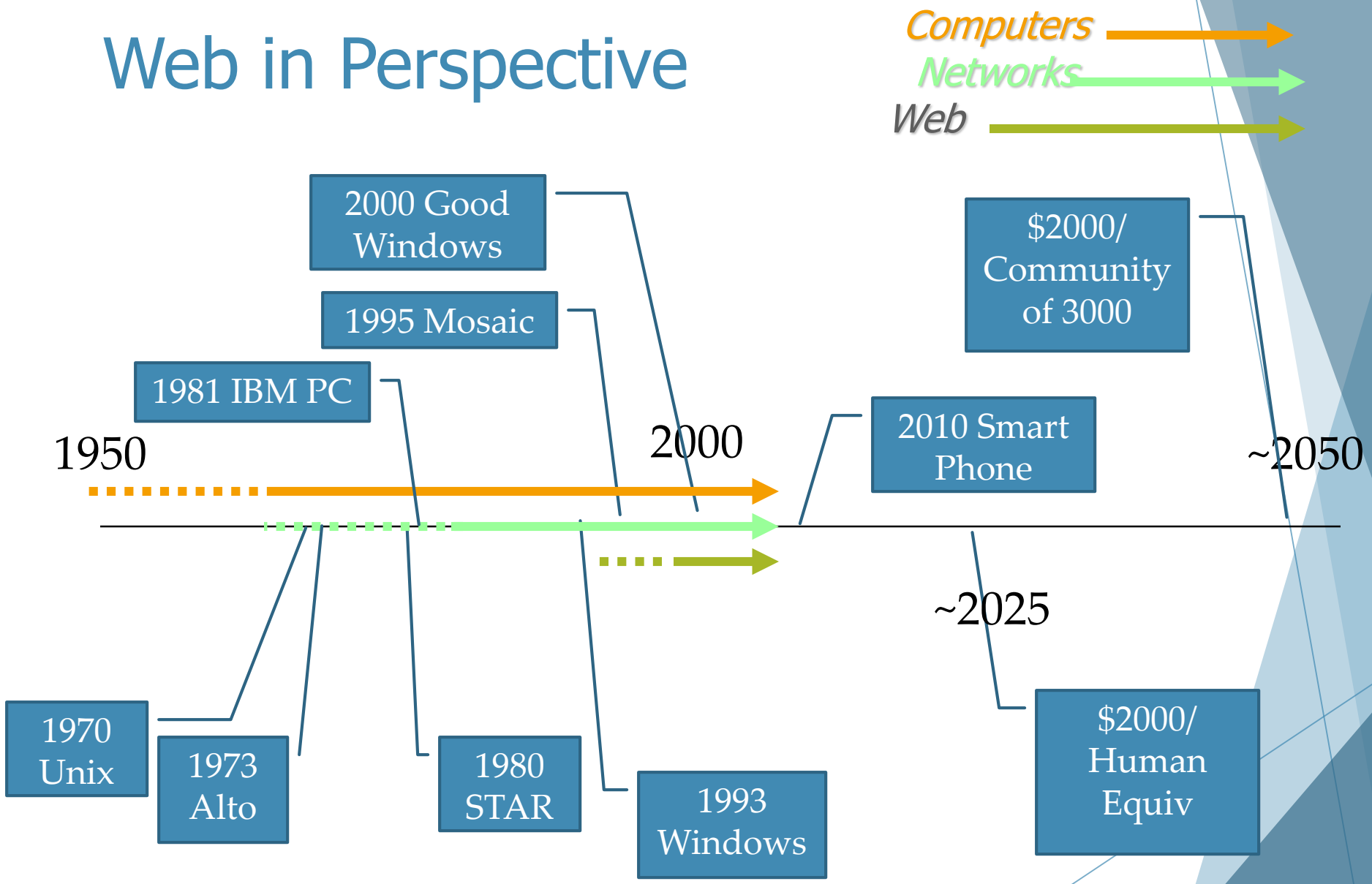
It’s Very Simple



Overview

- The Internet vs the World Wide Web
- What makes the WWW
  - HTML and XML
  - Identifiers
  - The Protocol
    - Requests and Responses

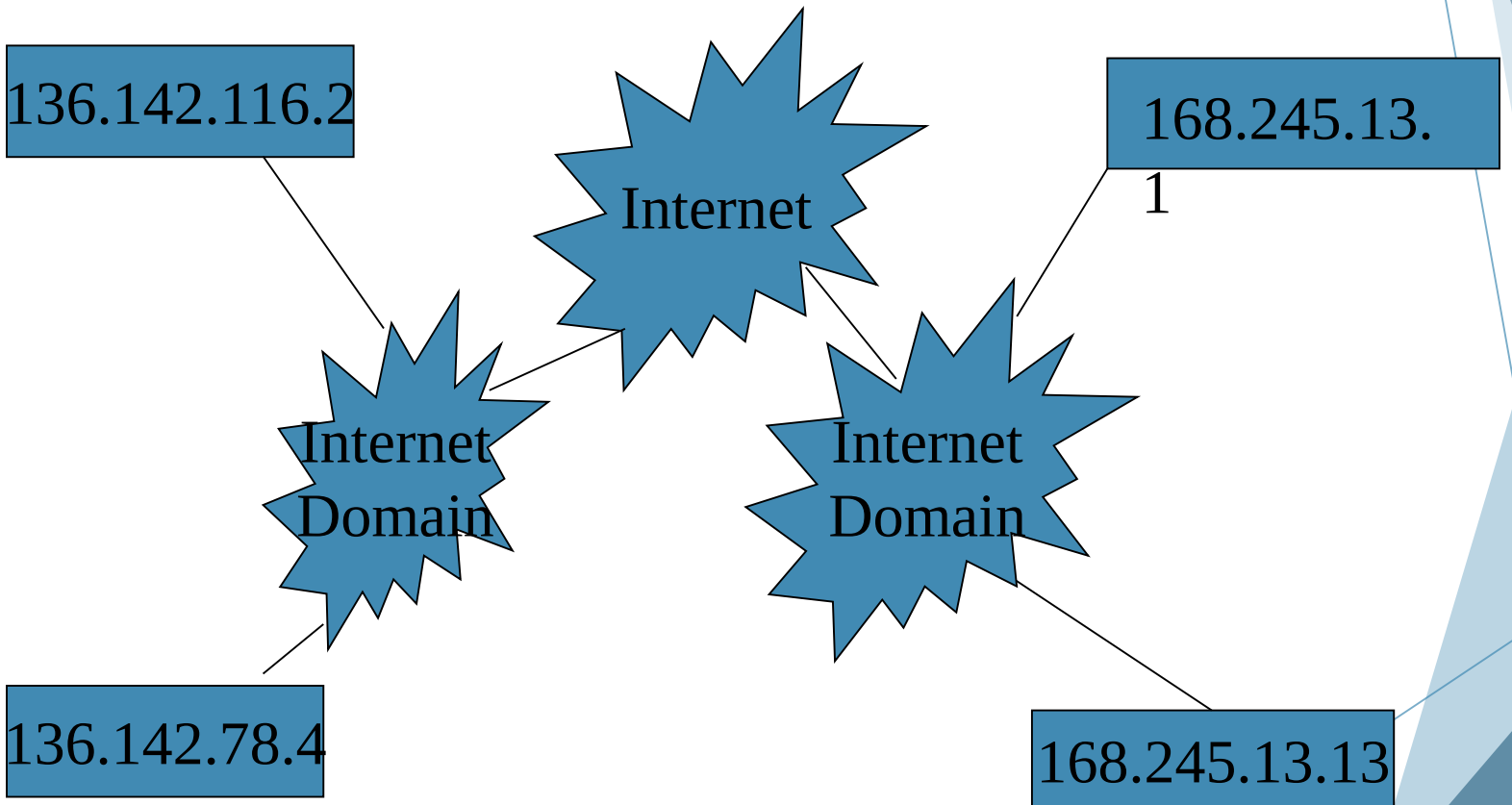
Web in Perspective



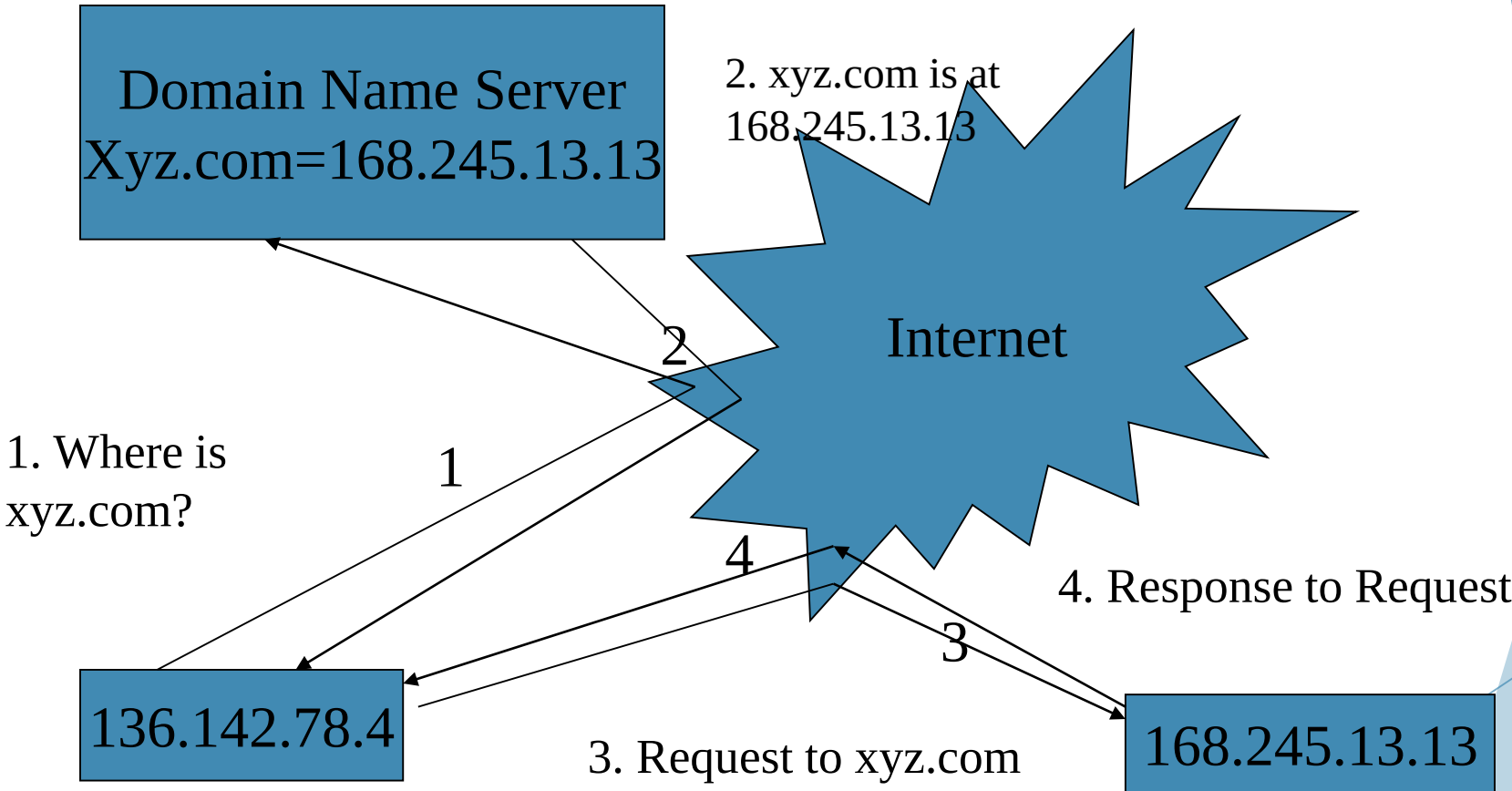
The Internet

- The internet is a set of communicating machines
- The basis for communications is:
  - a shared machine address space (IP)
  - A name lookup mechanism -- Domain Name Space (DNS)
  - A protocol for integral messaging (TCP)
  - A protocol for doing business (http)
  - Software to interpret the messages exchanged

The Internet Generically



An Internet Transaction



The World Wide Web (Parts)

- A method for defining resource locations
- A simple protocol
  - GET, POST
  - PUT, HEAD, OPTIONS, TRACE, DELETE
- A standard document format
  - SGML>>HTML>>XML
- Additional Capability
  - An increasingly complex server (state, authentication, encryption, application serving)
  - An increasing complex client (parse a variety of documents, trace links, spawn applications)

The Essence of the Web

```
{501}paradox:spring$ telnet www.sis.pitt.edu 80
Trying 136.142.116.9...
Connected to starfire1.sis.pitt.edu.
Escape character is '^]'.
GET /~spring/index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 12 Jan 2010 17:45:55 GMT
Server: SISWEB
Last-Modified: Fri, 11 Mar 2005 14:20:09 GMT
ETag: "5d5f46-119-4279b840"
Accept-Ranges: bytes
Content-Length: 281
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=ISO-8859-1

<html><head><title>Michael B. Spring</title></head>
<frameset cols= "20%, 80%">
<frame  name = "left" src="fr_toc.html" scrolling="auto">
<frame  name = "right" src="fr_wel.html" scrolling="auto">
</frameset>
<noframes>
<body>
You Need Frames for this.
</body>
</noframes>
</html>
Connection to starfire1.sis.pitt.edu closed by foreign host.
```

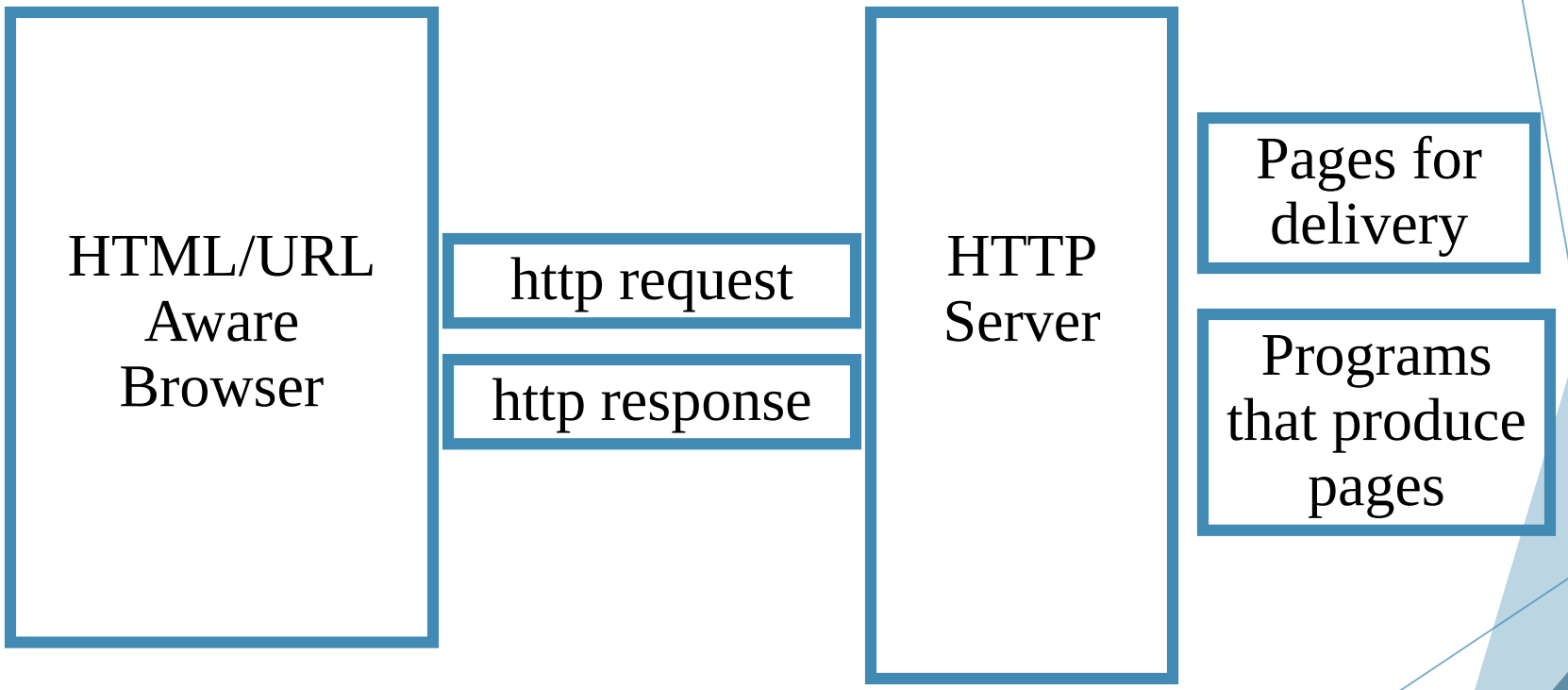
Connect to Port 80

Make Request

Read and act on response

Parse Document for additional sources

Basic Web Technology



Resource Identifier

- The WWW's most important contribution is a distributed system for identifying electronic resources.
- The WWW's Universal Resource Locator (URL) is one form of Universal Resource Identifier (URI).
- A URL is made up of five parts:
  - A protocol specification://
  - A host identifier
  - :A port specification (if not well known)
  - A relative path specifier
  - #A fragment identifier or ?data
- Examples include:

  http://abc.com/dir1/dir2/filename.html#locinfile

  http://abc.com/dir1/dir2/filename.html?xyz=abc&wtu=def

The http process

- The web protocol is very robust and very simple
- For each request, the client:
  - Does a DNS lookup if needed
  - Opens a connection to the server
  - Sends a request for a resource
- The server
  - Checks the availability of the resource
  - Returns the resource or an error message
  - Closes the connection

The structure of requests and responses

- Requests have a header and optionally a body
- The header may have a number of lines.
- The first line specifies the request type, resource, and http version.
- There are seven request types
  - GET, HEAD, POST, PUT, DEL, OPTIONS, TRACE
- The body is null for five of the request types and contains data for the POST and PUT types

- Responses have a header and a body
- The header has many lines but:
  - Begins with a status
  - Ends with a content type
- The body contains either the resource or an explanatory message

A Sample Request

- The user types the following in their client:

  http://www.sis.pitt.edu/~cascade/index.html

- The client sends a request consisting of a header:

  GET /~cascade/index.html HTTP/1.0
  If-Modified-Since: Fri, 10 Oct 1997 17:35:54 GMT;
  User-Agent: Mozilla/4.7 [en] (X11; I; SunOS 5.6 sun4u)
  Pragma: no-cache
  Host: www.sis.pitt.edu
  Accept: image/gif, image/jpeg, image/pjpeg, image/png, */*
  Accept-Encoding: gzip
  Accept-Language: en-US, en
  Accept-Charset: iso-8859-1,*,utf-8

A Sample Response

HTTP/1.1 200 OK

Date: Wed, 01 Dec 1999 16:11:19 GMT

Server: Apache/1.3.1 (Unix)

Last-Modified: Wed, 12 May 1999 20:31:56 GMT

ETag: "7a108-16c2-3739e53c"

Content-Length: 5826

Connection: close

Content-Type: text/html

<HTML>

<HEAD>

<TITLE> CASCADE </TITLE> </HEAD>…

<BODY>…

Request Types

- Get – to get a document or run a CGI script. When a form is used, the form info is placed in the request
- Post – to get a document or run a CGI script. When a form is used, the form info is placed in the body
- Head – used to ask for header info only
- Put – used to place a document on the server
- Delete – deletes a document on a server
- Options – tells what options the server has
- Trace – for debugging, tells what the server is doing

Status Codes

- Five categories of status code
  - 1xx: informational – used for development
  - 2xx: Successful response
  - 3xx: Redirection
  - 4xx: Client Error
  - 5xx: Server Error
- Frequently used codes:
  - 200 -- success
  - 301 and 302 – moved permanently or temporarily
  - 400 – bad request
  - 401 – unauthorized
  - 403 – forbidden
  - 404 – not found

Content Types

- The content types for WWW documents are all allowable mime types
- Whether the browser can handle them is not the concern of the server
- The common content-types are:
  - text/ascii text/html
  - image/jpg image/gif
  - application/pdf application/msword

More on Headers

- Accept
- Accept-Charset
- Accept-Encoding
- Accept-Language
- Accept-Ranges
- Age
- Allow
- Authorization
- Cache-Control
- Connection
- Content-Encoding
- Content-Language
- Content-Length
- Content-Location
- Content-MD5
- Content-Range
- Content-Type
- Date
- ETag
- Expect
- Expires
- From
- Host
- If-Match
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Last-Modified
- Location
- Max-Forwards
- Pragma
- Proxy-Authenticate
- Proxy-Authorization
- Range
- Referer
- Retry-After
- Server
- TE
- Trailer
- Transfer-Encoding
- Upgrade
- User-Agent
- Vary
- Via
- Warning
- WWW-Authenticate

HTML

- The body of an http message may be anything, but frequently it is a document encoded using HTML
- HTML is “Document Type Definition” (DTD) based on SGML
- HTML is a technically weak DTD (Schema)
  - It defines a very weak structure (e.g. H3 anywhere)
  - Some tags (e.g. bold) are too procedural

An Structure of an HTML Doc

- An HTML document has a <HEAD> and a <BODY>
  - Don't confuse with protocol the header and body
- The <HEAD> of an html document contains control information (meta tags, title, keywords, scripts, etc.)
  - The title element is actually required by the standard
- The <BODY> of an html document contains all of the elements that will normally appear in the browser window

A minimalist HTML Document

```
<html>
<head>
<title>some title</title>
</head>
<body>
</body>
</html>
```

HTML Elements

- HTML elements fall into ten categories
  - Overall document structure -- head and body
  - Text level formatting – bold, italic
  - Block level -- quote
  - List tags
  - Hyperlink tags
  - Image related tags
  - Table Tags
  - Form Tags
  - Frame Tags
  - Executable Content tags

Anchors and Hyperlinks

- HTML defines an element known as an Anchor
  - <a>This is an anchor
- A property or attribute of an anchor is its href – Hypertext Reference
  - Web HREF values are Universal Resource Locator
- Google Link
- A URL is made up four parts
  - A service identifier – e.g. http://
  - An Internet Address – e.g. www.sis.pitt.edu
  - A port overriding the default service specification – e.g. 8080
  - A absolute path ~spring/index.html

Forms Construction

- A form is an element in the body of an HTML document.
- A form element has two attributes – method and action
  - The method specifies which http protocol will be used
  - The action specifies the program that will process the data
- A form will have one or more inputs elements

A Sample Form

```
<from
   method = "POST"
   action = "http://augment.sis.pitt.edu/cgi-bin/form.cgi">

<p>name:
   <input type="text" size = "40"  maxlength ="80"
   name ="name" value     = "anonymous“ /></p>
<p>subject:
   <input type="text" size = "40" maxlength ="80"
   name ="subject" value = "none"/></p>
<p>
   <input type = "submit" name = "ssc" value = "send comment"/>
   <input type = "reset"  name = "clr" value = "clear omment">/></p>
</form>
```

Other elements

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
...	Set ... in boldface
...	Set ... in italics
...	Center ... on the page horizontally
...	Brackets an unordered (bulleted) list
...	Brackets a numbered list
...	Brackets an item in an ordered or numbered list
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
<img src="...">	Displays an image here
 ... 	Defines a hyperlink

Styles

- The desire to control styling of HTML documents has led to a variety of tools
- Most browsers allow the user some control of appearance
- In addition, style information can be included directly or by reference with a document
- For HTML documents, there are two versions of what is known as Cascading style sheets

Basic Rules for StyleSheets

- Style information may be embedded or in a separate file
  - Embedded -- **<style type = "TEXT/CSS">info </style)**
  - Separate file -- **<script type="text/javascript" src="styles-scripts/stdparts.js"></script>**
  - Local to an element **-- **
- Style information may be associated with an element, a class, and id or any combination
  - Element (all H2s) **H2 {color: green; font-size 12pt}**
  - Class (all elements with class="main) **.main {color: green; font-size 12pt}**
  - ID (the element with ID="first" **#first {color: green; font-size 12pt}**
  - Combo (all H2s with class "import") **H2.import {color: green; font-size 12pt}**

A Simple Style

```
<HTML><HEAD>
<STYLE TYPE = "TEXT/CSS">
    P {FONT: 12PT ARIAL; COLOR: GREEN;}
    H1 {FONT: 24PT ARIAL; COLOR: RED;}
    H2 {FONT: 8PT ARIAL; COLOR: YELLOW;}
</STYLE>
</HEAD><BODY>
    <H1> HERE IS AN H1</H1>
    <P>HELLO</P>
    <H2> HERE IS AN H2 </H2>
    <P>GOODBYE</P>
</BODY></HTML>
```

A More Complex Stylesheet

```
H2{
        FONT-WEIGHT: bold;
        FONT-SIZE: 14pt;
        COLOR: #01184c;
        FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
        TEXT-DECORATION: none;}

IMG.left{margin-top: 14px; margin-right: 14px; margin-bottom: 14px;}

IMG.center{margin: 14px;}

BODY{
        BACKGROUND-COLOR: #FFFFFF;
         margin-top: 0;    margin-left: 0;
        FONT-WEIGHT: normal;
        FONT-SIZE: 10pt;
        COLOR: #01184c;
        FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
        TEXT-DECORATION: none; }
```

IS 2560: The Evolution of HTML
Focus on HTML 5

Graduate Program Information Science and Technology

School of Information Sciences

University of Pittsburgh

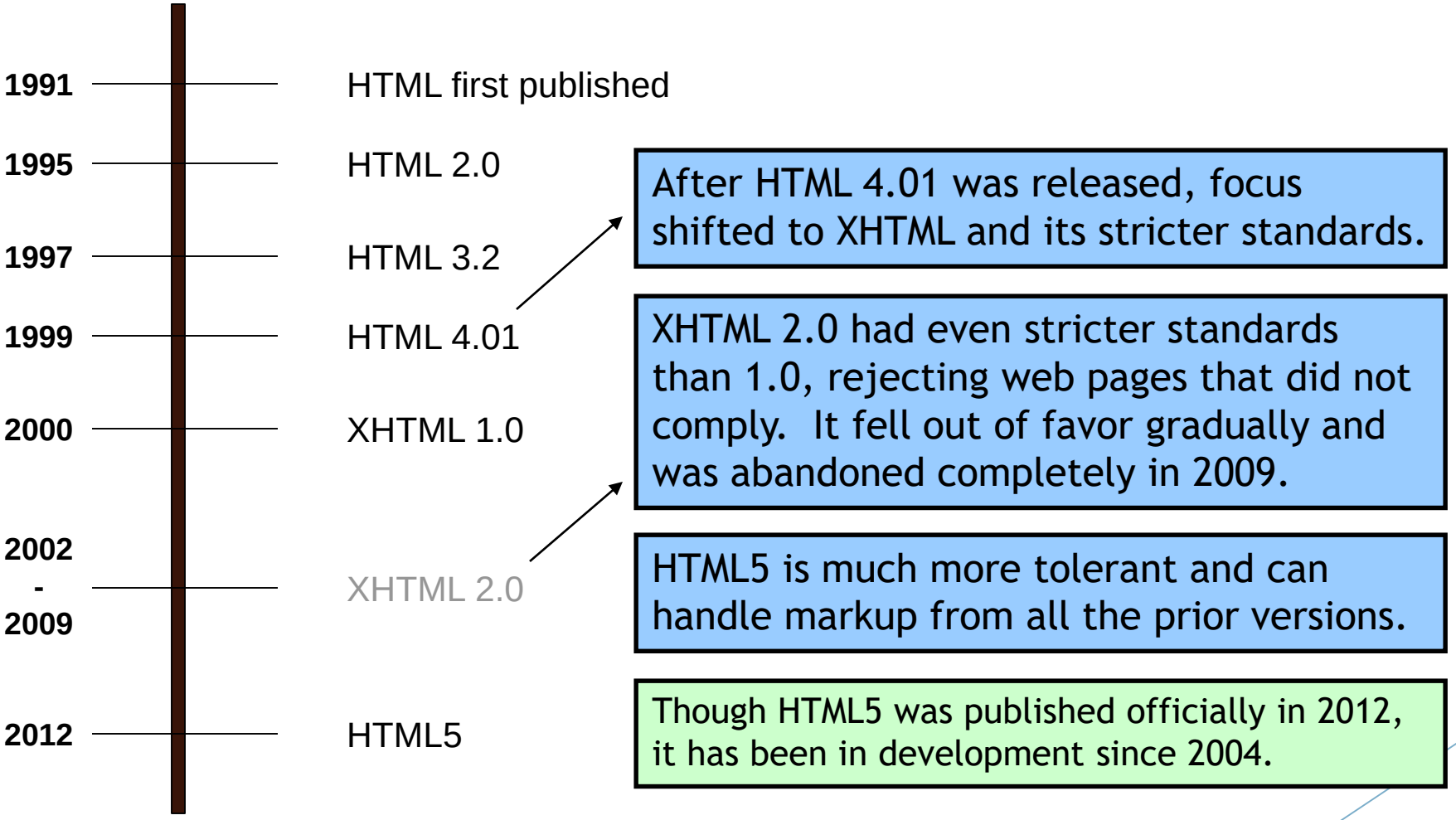
Overview

- HTML Starts
- Standardized HTML (2.0)
- HTML 4
- Focus on HTML 5
  - Motivation
  - Multimedia
  - New and Deprecated Elements
  - Enhanced DOM Model
  - New API's

The First HTML

- In 1991, Berners-Lee provided the first simple HTML document elements
- It was based on an SGML DTD used at CERN
  - It included the heading elements
  - Some paragraph and list elements
  - The anchor element
- As time went on new elements were appearing based on the browsers

History Of HTML



Motivation for HTML 5

- The primary motivation for HTML 5 is to update HTML 4, which was adopted in 1997 and last update in 1999. It incorporates:
  - Better support for a complete DOM and the supporting JavaScript API.
  - Recognizes the evolution and role of CSS
  - Provide support in a standard fashion for multimedia.
  - Increases the amount of structural markup
- It seeks to create one standard that encompasses HTML 4 and XHTML 1

Goal of HTML5

- Support all existing web pages. With HTML5, there is no requirement to go back and revise older websites.
- Reduce the need for external plugins and scripts to show website content.
- Improve the semantic definition (i.e. meaning and purpose) of page elements.
- Make the rendering of web content universal and independent of the device being used.
- Handle web documents errors in a better and more consistent fashion.

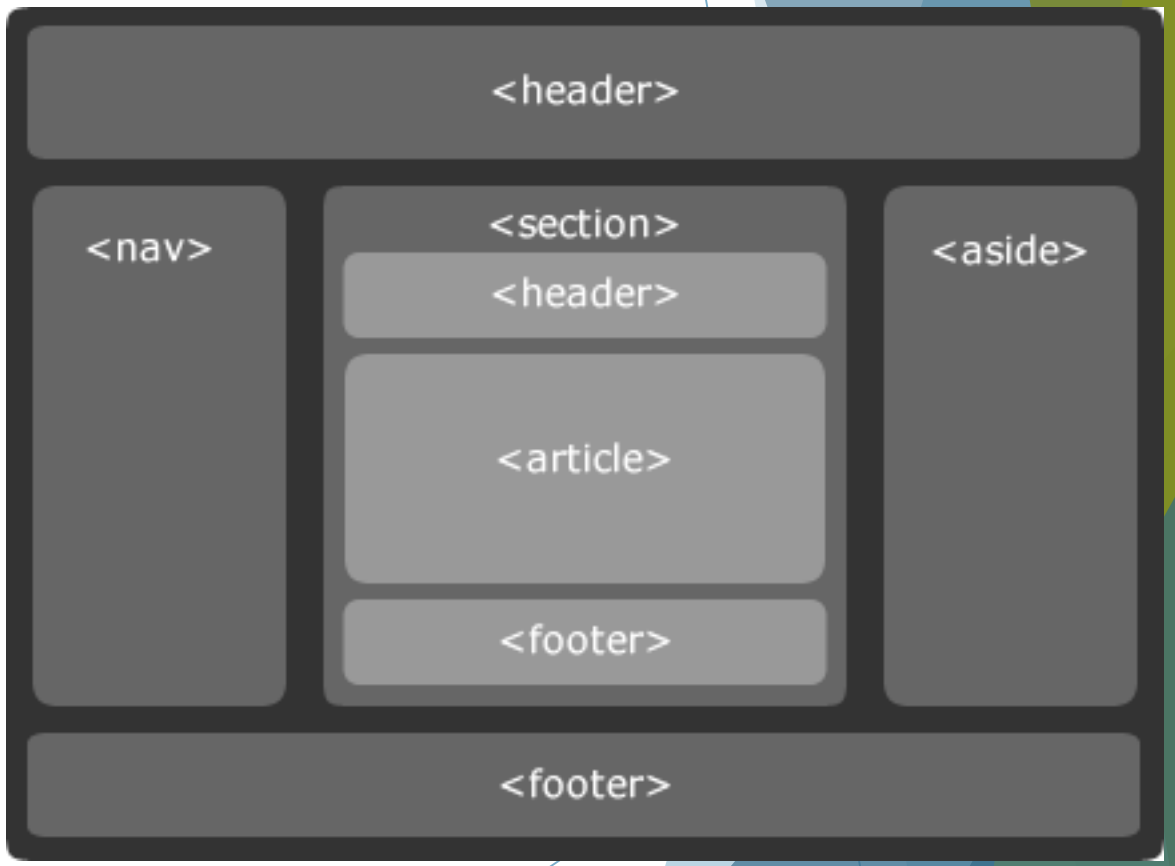
Multimedia

- The object tag in HTML 4 is now replaced with several tags for multimedia:
  - video provides for movie clips of video streams
  - audio provides for sounds music or audio stream
  - canvas provides for graphics defined by a script
  - track provides for text tracks for media players
- In addition SVG becomes integrated into content and SVG and MathML can be used inline.
- Currently, there are 3 supported video formats for the video element:

Format	IE	Firefox	Opera	Chrome	Safari
Ogg	No	3.5+	10.5+	5.0+	No
MPEG 4	No	No	No	5.0+	3.0+
WebM	No	No	10.6+	6.0+	No

Structure and Semantics

- There are new and redefined tags that begin to provide more structure and some level of semantics. Selected new elements include:
  - article and section provide containers for heading and content
  - header and footer which are additional document or article sections. Headers might include nav sections and footers might include author, contact and copyright information.
  - nav summary, details and aside provide other specialized parts of larger documents and have suggested content



Video Example

```
<!DOCTYPE html>
<html>
<body>

<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>

</body>
</html>
```

- Source: W3schools

Media Playback

- Both the video and audio objects have a variety of methods and properties that may be accessed.
- They can be looped, paused and played
- Their length (duration) can be obtained and their current position, or initial position can be ascertained.
- There are also interfaces and API's for media that have multiple tracks.
- Finally a media controller allows for the synchronization of multiple media elements.

Other Form Elements

- Forms have new elements including data list for input options, keygen for keys to authenticate users, and output for information written by a script
- In addition input fields now have a number of attribute types that are defined, such as tel (telephone number), url, email, date, time, number, color, etc.
- Further, form elements have several new attributes such as autofocus (element that gets initial focus), autocomplete (there is an autocomplete function), form, novalidate, required, etc.

Deprecated Elements

- Several elements are deprecated moving further from presentation elements:

  - Acronym
  - Applet
  - Basefont
  - Big
  - Center
  - Dir
  - Font
  - Frame
  - Frameset
  - Isindex
  - Noframes
  - Strike

New API's

- The canvas element for immediate mode 2D drawing. See Canvas 2D API Specification 1.0 specification
- Timed media playback
- Offline storage database (offline web applications)
- Document editing
- Drag-and-drop
- Cross-document messaging
- Browser history management
- MIME type and protocol handler registration
- Microdata

Drawing

- Beyond support for SVG, HTML5 will also provide support for the Canvas element.
- JavaScript will be used to write to the canvas element.
- For the most part, once written, “objects” will exist simply as parts of the canvas bitmap.
- Beyond rendering, the canvas element will also have all of the image manipulation functions one might expect – clipping regions and matrix transformations,

Local Storage

- Local storage historically has been limited to cookies.
- Support for local storage is now expanded to provide for larger amounts of data and includes two objects:
  - Local storage with no time limit
  - Session storage

Drag and Drop

- An extensive API is being included in HTML5 to allow for drag and drop actions.
- Making use of div elements, it will be possible to define them as “dragable”
- Several events will be tied to javascript event code to handle the actions. These include:
  - ondragstart
  - ondragend
  - ondragover
  - ondrop
- The functions will look something like the following:

```
function dragDrop(ev) {
var idDrag = ev.dataTransfer.getData("Text"); // Get id of the item
var dnode = document.getElementById(idDrag); // get the node
ev.target.appendChild()); // Append the dragged item
ev.preventDefault(); // Turn off the default behaviour
}
```

Browser History

- The development of dynamically updated pages using AJAX presents problems for the window history. Historically, the history API provided three methods – back(), forward(), and go():
- The HTML5 History API provides two new methods:
  - pushstate()
  - replacestate()
- They work with the window popstate() method to access previous window states.
- Basically, rather than access a URL, the popstate method accesses a state object saved by the browser. It is currently limited in size to an object less than 640K. The popstate method ust be prepared to deal with the JSON object appropriately

Cross Document Messaging

- Historically, browsers have erected walls to prevent documents in different domains form interacting.
- Cross site scripting has been a hack that attempted to bridge this gap using JavaScript.
- In HTML5, a javaScript attached to a document from one domain can "post" a message to a document from a different domain.
- The receiving document uses an event listener that can restrict incoming messages to approved domains.

Assignment 1

Due September 17 at midnight.

For this assignment you need to create your own personal website and upload it to Pitt server. If your PittID is aaa221 your URL will be

http://www.pitt.edu/~aaa221/

Here is a tutorial on how to upload documents to that server.

http://technology.pitt.edu/help-desk/how-to-documents/creating-your-own-web-site

Here is a tutorial on HTML and CSS.

https://www.w3schools.com

Below is the criteria for the site.

Don't forget to upload it to your Git repository(Use upload file there).

At least 5 separate pages xml declaration for HTML 5 All open tags must be closed All tags must be lowercase Empty tags must be closed Attributes must always be quoted Meet web accessibility guidelines Alt attributes for images, src	Table headers Include simple JavaScript Include a simple navigation scheme Include one or more images Usage of 3 New HTML5 tags Some use of SVG Make use of a CSS style sheet Overall Aesthetics

Short demo

- Inspecting HTML Document
- Mozilla developer network MDN

https://developer.mozilla.org/en-US/docs/Web/SVG/Element/defs

- Increase your website quota
- Upload to your Pitt server
  - **Host unixs.cssd.pitt.edu**
  - Port 22
  - Username : your pitt id
  - Password: your pitt password

For Next Lecture

- Reading
  - Web Programming Chapters 1, 3 and 12
  - Make W3schools your BEST FRIEND
    - https://www.w3schools.com/bootstrap/bootstrap_get_started.asp
    - https://www.w3schools.com/html/default.asp
    - https://www.w3schools.com/html/html_css.asp
  - Prepare for the In-class activity which will be related to the topics we will discuss(HTML5 page with Canvas + bootstrap)
  - You should be familiar with FTP uploads to your Pitt Server
- Assignment 1(due 9/17)
- Try to access your pitt.edu domain(Try creating one page and upload it)
- Get familiar with the tools
- Bring your Laptop
- Form Groups(Use the discussion board to reach other students)
- Install the tools before the class. If you have any issue-> email me
- Create git repository account if you don't have one