

CSCI 311

Course Intro

Administrivia

- ▶ Lecture:
 - ▶ Tuesday, Thursday 10:00 - 11:50
- ▶ Labs:
 - ▶ Start next week
- ▶ Instructor: Sarah Carruthers (sarah.carruthers@viu.ca)
- ▶ Office Hours: Tuesday and Thursday 13:30-14:15
- ▶ Course homepage:
 - ▶ csci.viu.ca/~carruths/Courses/CSCI311
- ▶ Syllabus:
 - ▶ <http://csci.viu.ca/~carruths/Courses/CSCI311/CSCI3112020Outline.html>
- ▶ Texts:
 - ▶ Required: [The Missing Link: An Introduction to Web Development and Programming](#) Michael Mendez

Administrivia

- ▶ Mark breakdown:
 - ▶ Exams: 40%
 - ▶ Lab Exam
 - ▶ Final exam
 - ▶ Labs: 20%
 - ▶ Final Project: 40%
- ▶ To pass the course you must pass **both** the lab portion of the course **and** both exams

Labs

- ▶ Roughly bi-weekly hands-on exercises
- ▶ Alternating with time to work on team project
- ▶ Submitted via VIU Learn

Project

- ▶ Final group project to demonstrate what you've learned
- ▶ Teams of three (formed next week during lab)
- ▶ Details found on course page
- ▶ Worth 40% of final mark
- ▶ Demo in final weeks of lab
- ▶ Presentation at end of course

Project Deliverables

Deliverable	Due Date	What is submitted
Team Formation	January 17, 20:00	CSCI311SkillsInventory
Proposal	January 26, 18:00	Proposal.pdf
Functional Prototype	February 16, 18:00	All html, css (.zip) MidwayReport.pdf
Peer Evaluation and assessments	February 23, 18:00	Two Completed Evaluations (pdf) of other project Team/individual assessments
Working Project	March 27, 18:00	All code
Stretch Goal Presentation	End of term	
Live Demo	March 30, in lab	n/a
Self-Assessment II	April 10, 18:00	Team an Individual Assessments

How this course will work

- ▶ Pre-assigned readings and tutorials
- ▶ In class we will:
 - ▶ go over concepts in broad sweeps
 - ▶ do demos
 - ▶ do activities to cement concepts
- ▶ Labs:
 - ▶ simple practice
- ▶ Project:
 - ▶ design and develop a dynamic web site

Course Topics

- ▶ The internet and the web
- ▶ HTML for structure
- ▶ CSS for layout
- ▶ JavaScript and Forms
- ▶ Dynamic HTML
- ▶ PHP
- ▶ MySQL
- ▶ Responsive Design
- ▶ Security
- ▶ jQuery, AJAX, Node

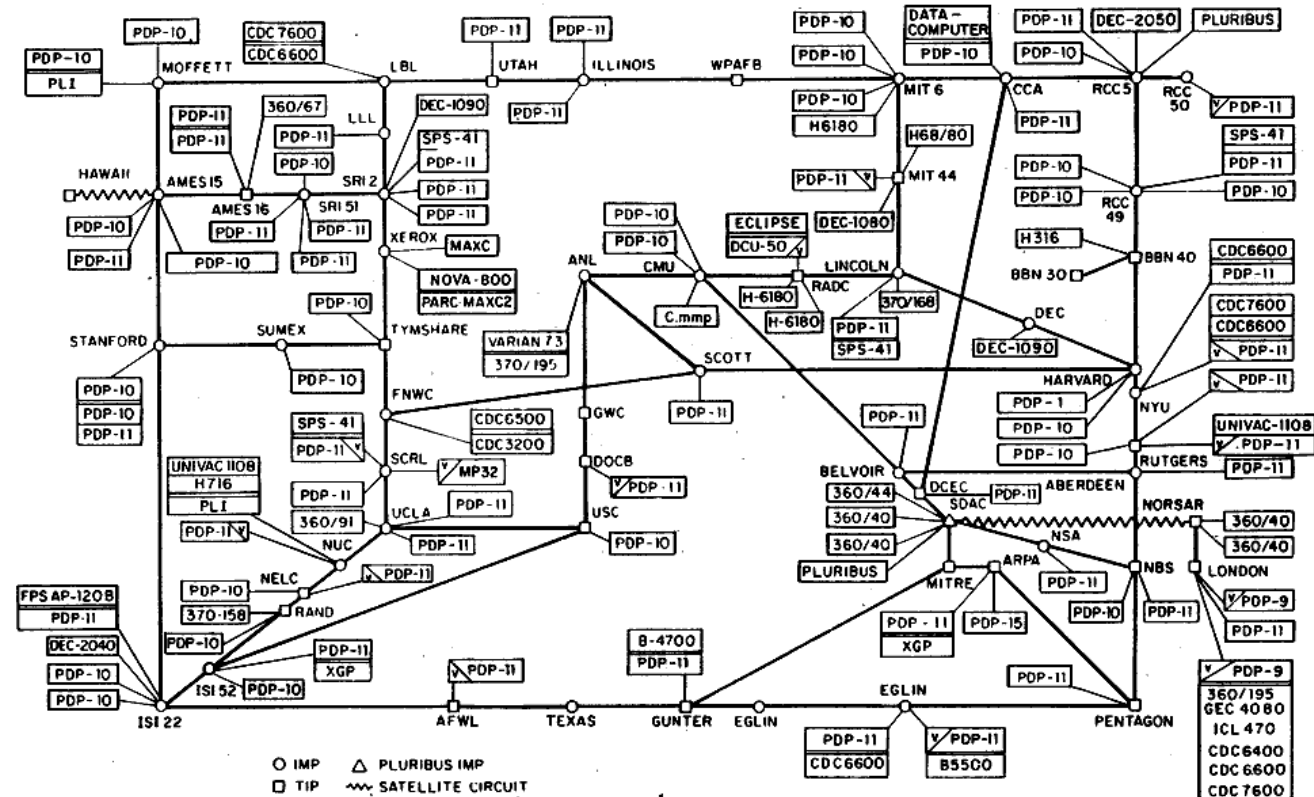
This and next week's Learning Objectives

- ▶ Course Objectives and Outline
- ▶ Understand what the web is
- ▶ Brief history of the internet
- ▶ Become familiar with the web's technologies and protocols
- ▶ Introduce how to develop and serve web pages
- ▶ Learn basic html and css

A Brief History of the Internet



ARPANET LOGICAL MAP, MARCH 1977



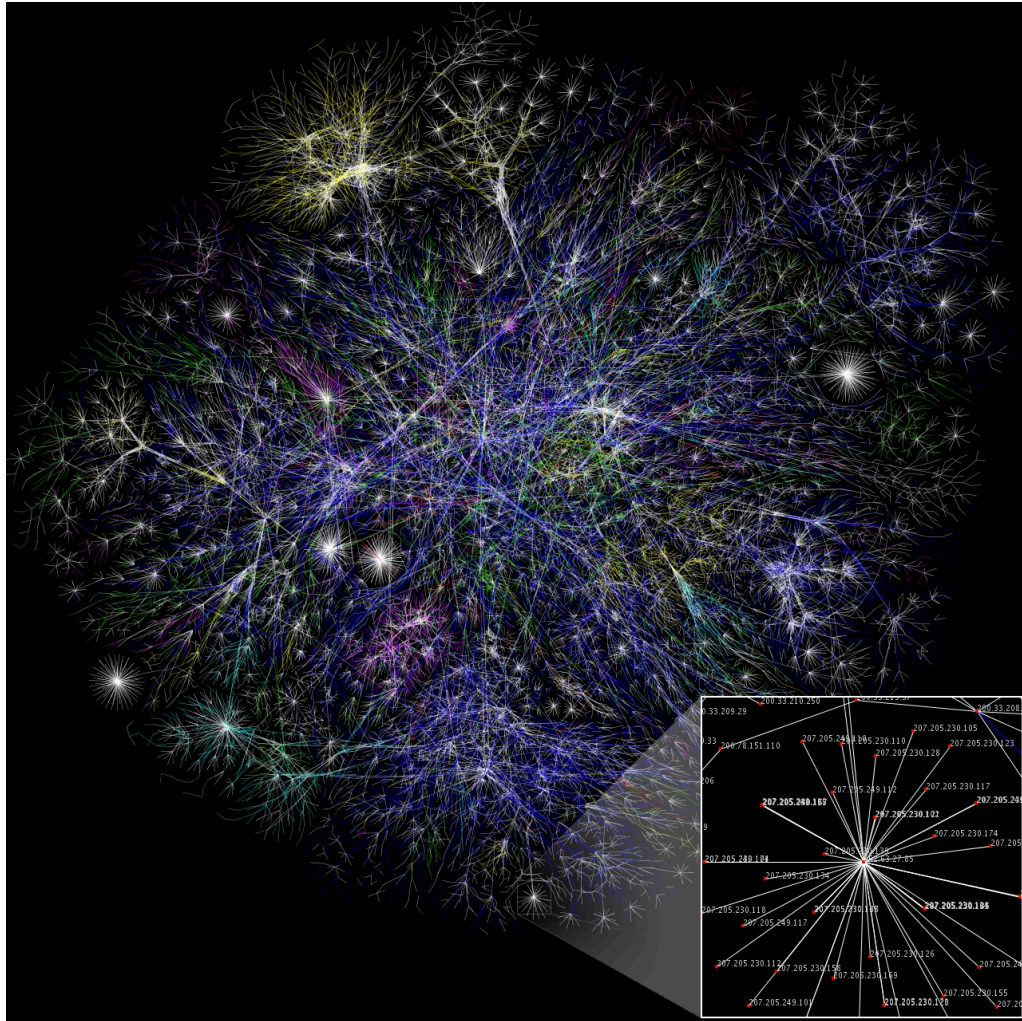
(PLEASE NOTE THAT WHILE THIS MAP SHOWS THE HOST POPULATION OF THE NETWORK ACCORDING TO THE BEST INFORMATION OBTAINABLE, NO CLAIM CAN BE MADE FOR ITS ACCURACY)

NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

<https://upload.wikimedia.org/wikipedia/commons/7/7b/Chosin.jpg>

https://upload.wikimedia.org/wikipedia/commons/b/bf/Arpanet_logical_map,_march_1977.png

The Web and the Internet



a look at the backbone of the internet
from:
https://en.wikipedia.org/wiki/Internet_backbone

Standard Web Technologies

- ▶ World Wide Web Consortium (W3C) (<https://www.w3.org/>)
 - ▶ international consortium that develops and publishes open standards
 - ▶ aims to ensure long-term growth of web
 - ▶ recommends a list of standard web technologies:
 - ▶ HTTP
 - ▶ HTML5
 - ▶ CSS3
 - ▶ JavaScript
 - ▶ DOM
 - ▶ PHP
 - ▶ MySQL
 - ▶ DHTML

HTTP

- ▶ HyperText Transfer Protocol
 - ▶ Has been in use since 1990
 - ▶ Current version HTTP/1.1, HTTP/2 supported by major servers
 - ▶ application-level protocol for distributed, collaborative, hypermedia information systems
 - ▶ generic and stateless
 - ▶ can be used for hypertext as well as other kinds of objects:
 - ▶ e.g., name servers, error codes, headers
 - ▶ request/response protocol

HTML5

- ▶ Hypertext Markup Language
- ▶ Provides structure for your content
- ▶ Newest version of HTML
- ▶ HTML5 is also often used to mean “open web platform”
- ▶ Goals of HTML5:
 - ▶ eliminate need for plugin services (like Flash)
 - ▶ improve security
 - ▶ replace scripting with markup
 - ▶ device independent

HTML5

- ▶ Introduces new audio and video elements
- ▶ provide a dynamic drawing area, or canvas
- ▶ makes forms easier to use
- ▶ drag and drop programming
- ▶ client-side file access
- ▶ MathML for math
 - ▶ XML-based
 - ▶ for presenting equations
- ▶ SVG for 2D graphics
 - ▶ Scalable Vector Graphics
 - ▶ XML-based vector image format

CSS

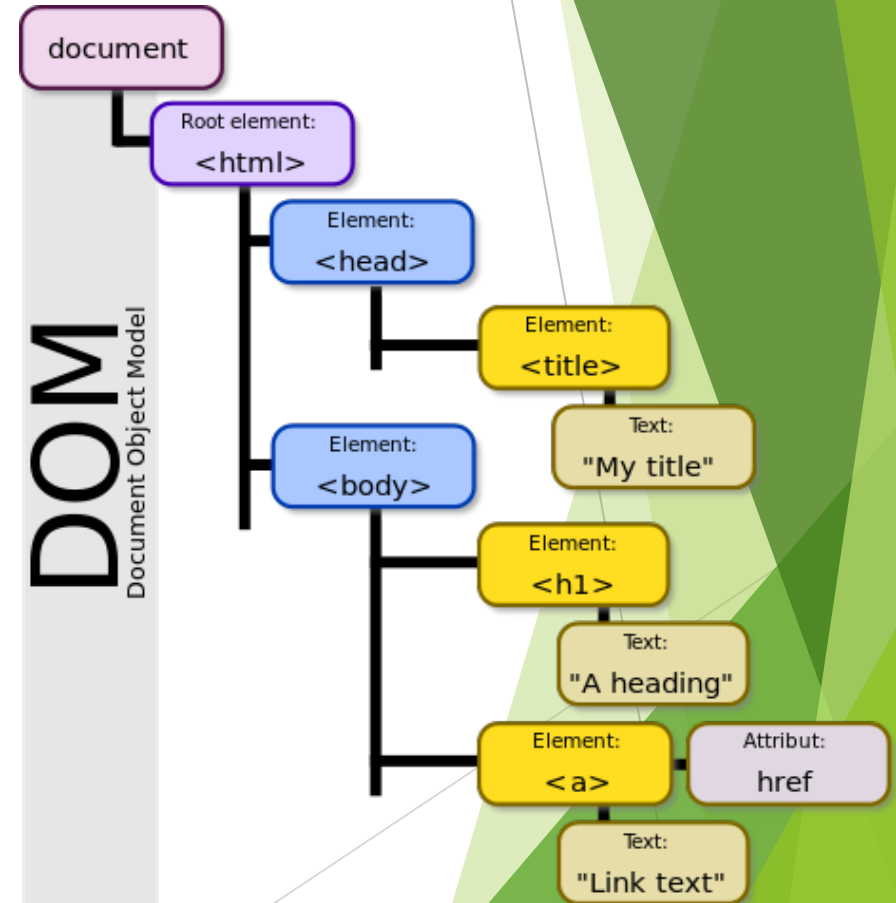
- ▶ Cascading Style Sheets
- ▶ Simple way to add style to your html and web documents
- ▶ add colour, whitespace, font, styling
- ▶ makes it possible to
 - ▶ separate the content from the presentation
 - ▶ share presentation rules across multiple pages
 - ▶ make accessibility easier to achieve
 - ▶ make responsive design easier

JavaScript

- ▶ Client-side scripting language (mainly)
- ▶ shares some syntax with Java, but is a distinct language
- ▶ imperative
- ▶ interpreted
- ▶ dynamically-typed
- ▶ object oriented
- ▶ <https://www.destroyallsoftware.com/talks/wat>

DOM

- ▶ Document Object Model
- ▶ platform-neutral and language-neutral interface
- ▶ lets programs/scripts dynamically access:
 - ▶ content
 - ▶ structure
 - ▶ style
- ▶ Treats XML, HTML, XHTML document like a tree



PHP



- ▶ Originally stood for: Personal Home Page
- ▶ PHP: Hypertext Preprocessor
 - ▶ a fun recursive acronym
- ▶ used mainly for web development
- ▶ can be embedded in HTML

```
<body>  
    <?php echo '<p>Hello World!</p>'; ?>  
</body>
```

- ▶ typically interpreted
- ▶ used to dynamically generate content
- ▶ unlike JavaScript, is server-side

MySQL

- ▶ Open-sourced relational database management system (RDBMS)
- ▶ central component of LAMP
 - ▶ Linux
 - ▶ Apache
 - ▶ MySQL
 - ▶ Perl/PHP/Python
- ▶ used by many existing applications (WordPress, Google, ...)
- ▶ can be run on cloud computing platforms

DHTML

- ▶ Dynamic HTML
- ▶ client-side system for:
 - ▶ responsive web pages
 - ▶ interactive web pages
- ▶ composed of:
 - ▶ mark up language (like HTML)
 - ▶ client-side scripting language (like JavaScript)
 - ▶ presentation language (like CSS)
 - ▶ DOM
- ▶ allows a web page to be dynamically changed by a scripting language **after** the web page is loaded (unlike a dynamic web page)

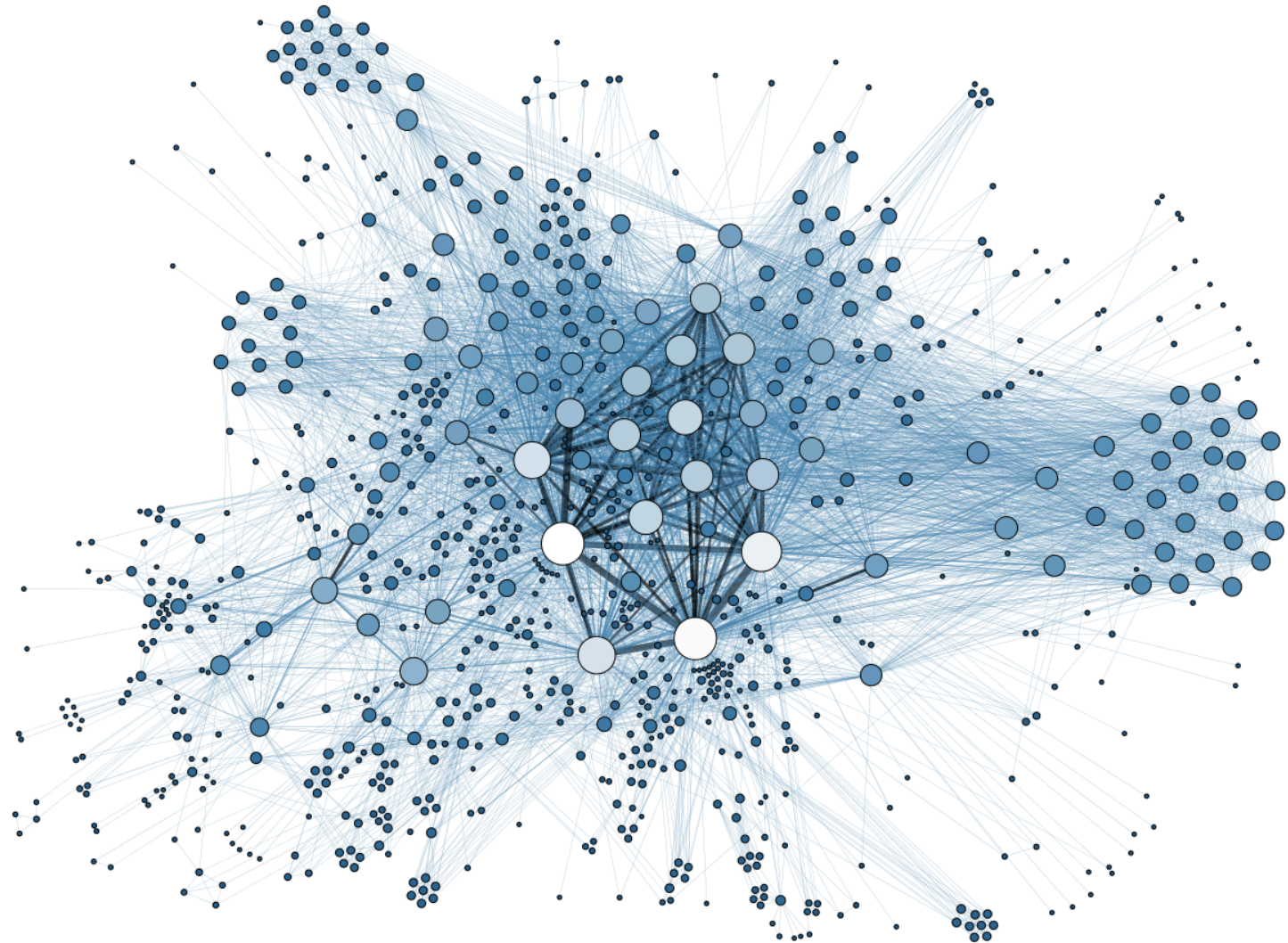
Mobile websites

- ▶ more and more content is accessed via mobile
- ▶ makes developing content that works on mobile more vital



https://upload.wikimedia.org/wikipedia/commons/4/47/Virato_Multiscreen.jpg

The Internet and the WWW



Other dedicated computers

- ▶ Hubs
- ▶ Switches
- ▶ Routers
- ▶ Bridges
- ▶ Gateway

Networking Protocols

- ▶ Networking protocols deal with many things...
- ▶ Protocols are just a set of rules that govern how two or more entities transmit information to each other

Activity



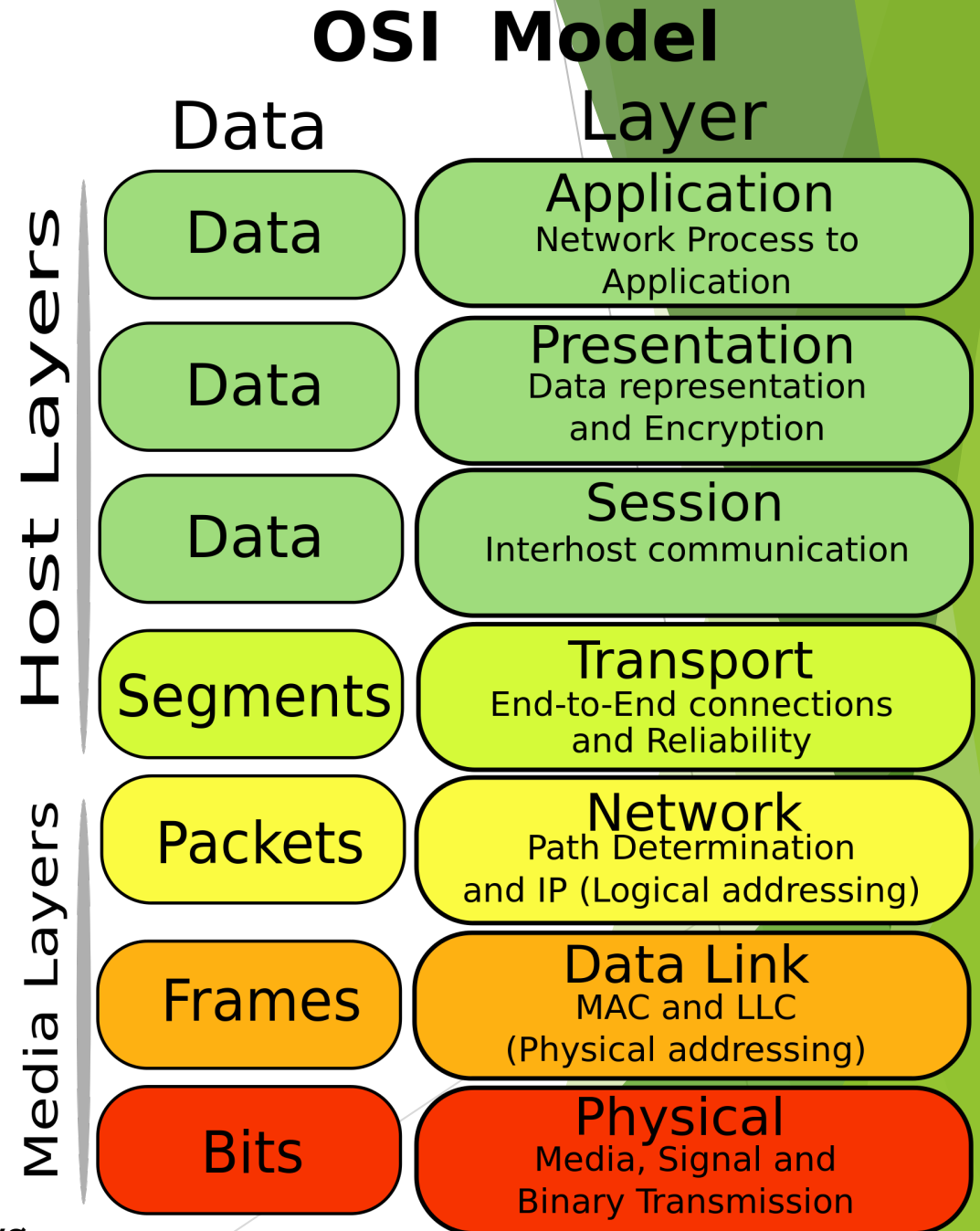
Networking Protocols

- ▶ Open Systems Interconnection Model (OSI Model)
 - ▶ Conceptual model of communication over a network
 - ▶ A kind of 7-layer dip...



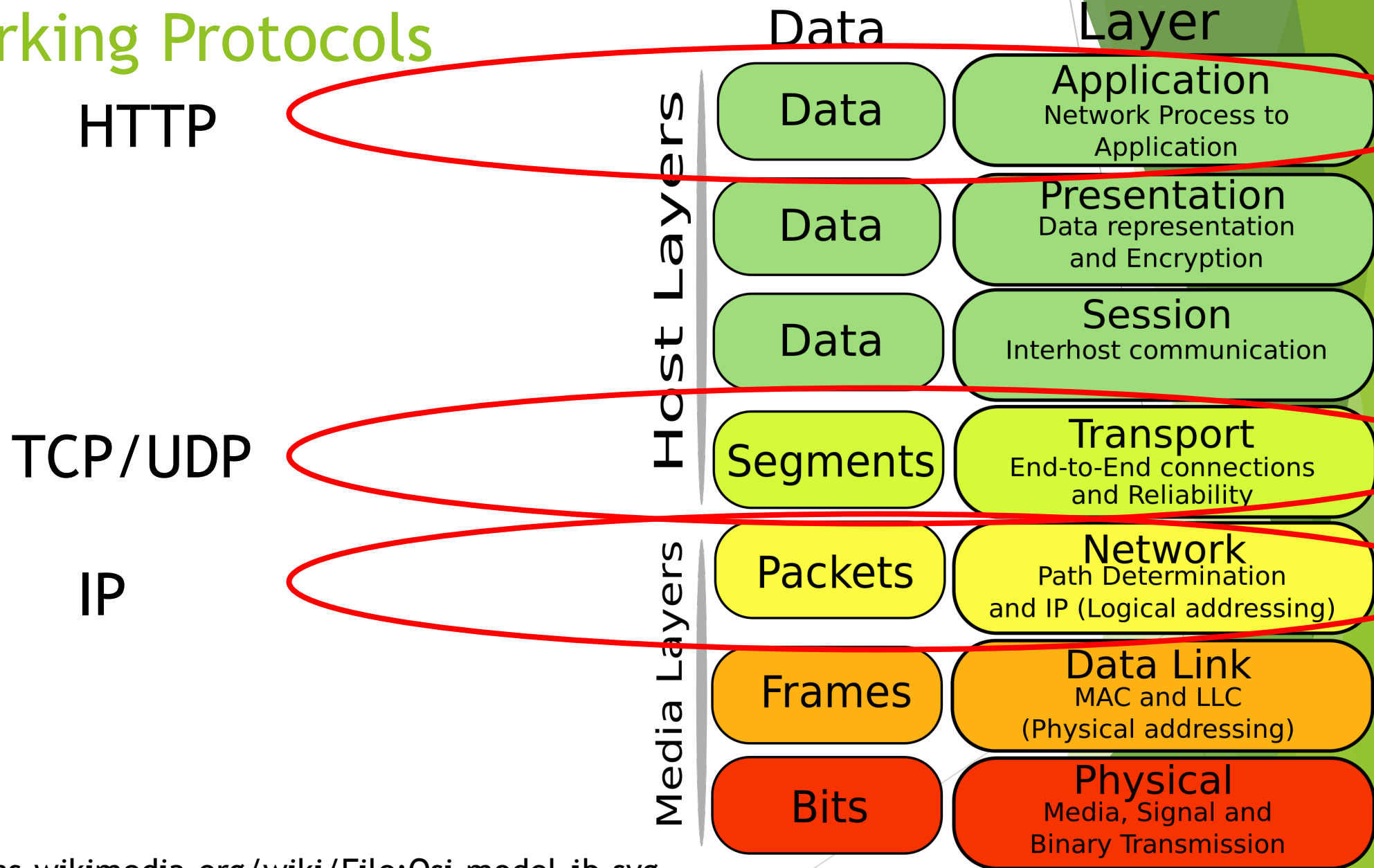
Networking Protocols

- ▶ Open Systems Interconnection Model (OSI Model)
 - ▶ Conceptual model of communication over a network



Networking Protocols

OSI Model



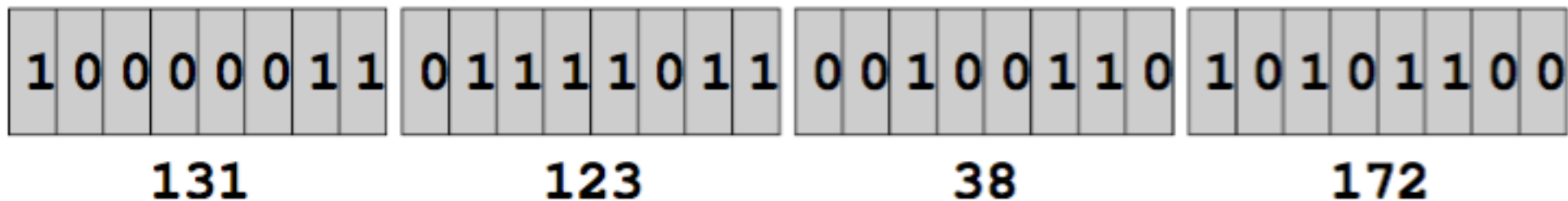
HTTP

- ▶ Request-response protocol
- ▶ This protocol lets intermediate network elements facilitate communication between User Agents (and other agents) and Servers
- ▶ Request message
 - ▶ Different methods: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT
- ▶ Response message:
 - ▶ Various status codes can be returned

IP Addresses and Domain Names

- ▶ IPv4 Address
 - ▶ a unique address for each host
 - ▶ 32-bit number
 - ▶ uses dot-notation, 255/byte
 - ▶ easy for computers to use
 - ▶ harder for humans to remember

FIGURE 1.2: IP Address



IP Addresses and Domain Names

- ▶ IPv6 Address
 - ▶ 128-bit number
 - ▶ Roughly 3.403×10^{38} addresses

An IPv6 address (in hexadecimal)

2001:0DB8:AC10:FE01:0000:0000:0000:0000



2001:0DB8:AC10:FE01:: Zeroes can be omitted

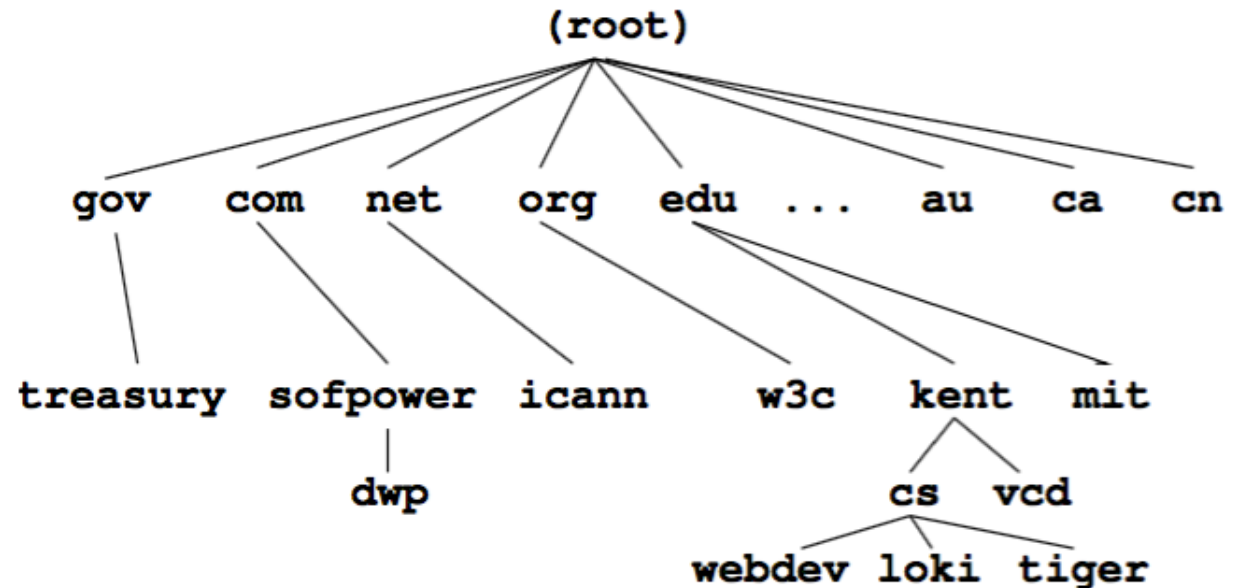
0010000000000001:0000110110111000:1010110000010000:1111111000000001:
0000000000000000:0000000000000000:0000000000000000:0000000000000000

Domain-based Name

- ▶ e.g., viu.ca
- ▶ generic top-level domain (Generic TLD)
 - ▶ .edu
 - ▶ .org
 - ▶ .com
- ▶ Internet domain name hierarchy (figure)
- ▶ easier for humans to use

tiger.cs.kent.edu.
dwp.sofpower.com.

FIGURE 1.3: The Domain Name Hierarchy

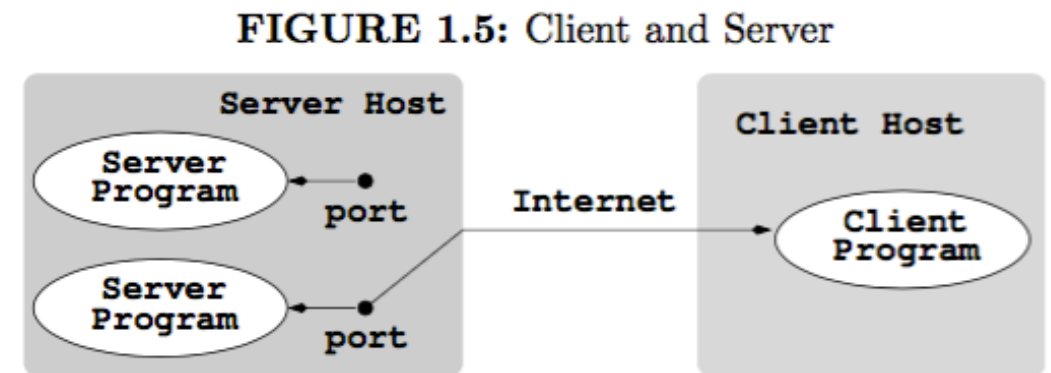


Domain Name System (DNS)

- ▶ The set of all domain names for hosts on the Internet
- ▶ Changes with time as hosts are added and removed
- ▶ Can be updated at any time without a central control
- ▶ The DNS name service is a distributed database service
 - ▶ dynamic update and retrieval of info in the database
- ▶ Browsers and other client programs typically contact a DNS name server before contacting a server

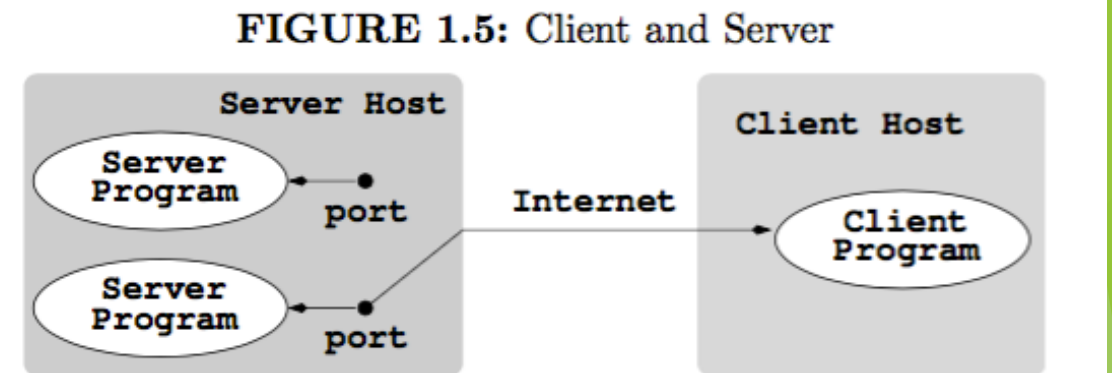
Web Clients and Servers

- ▶ Server program:
 - ▶ provides services on host
 - ▶ email (SMTP)
 - ▶ remote login (SSH/SFTP)
 - ▶ www
 - ▶ each service will have its own port
 - ▶ port + IP identifies the program



Web Clients and Servers

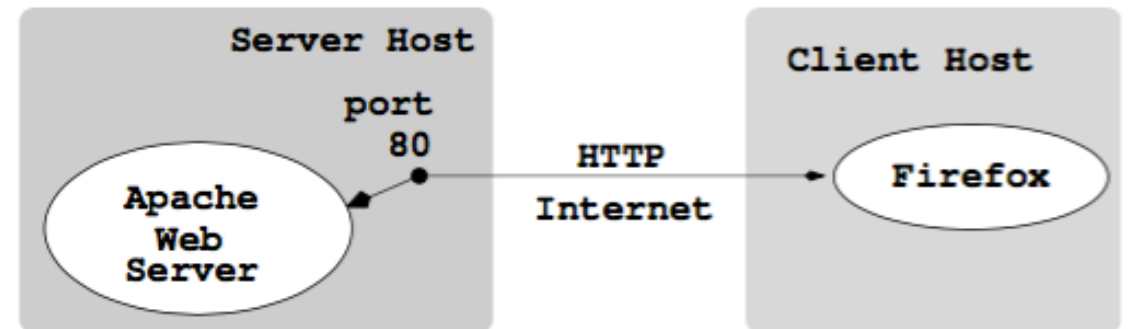
- ▶ Client program:
 - ▶ connects a computer to a server on another host to get a service
 - ▶ email programs (Thunderbird, Outlook)
 - ▶ Browsers (Firefox, Chrome...)
 - ▶ OpenSSH



Web Hosts and Web Clients

- ▶ Web host
 - ▶ a computer that stores files and other resources
 - ▶ can run web servers
- ▶ Web client
 - ▶ a user agent program that accesses the Web
 - ▶ e.g., a web browser

FIGURE 1.6: Web Client and Server



Uniform Resource Locators (URLs)

- ▶ Identify resources available on the web
- ▶ full form:
 - ▶ *scheme://server:port/pathname?query_string*

Web Hosting

- ▶ Publishing on the web involves
 - ▶ designing and constructing web pages
 - ▶ writing programs for the website
 - ▶ placing the files and folders on a hosting site
- ▶ As a university student you'll have access to free hosting
- ▶ There are numerous hosting companies that will host for a fee

Web Servers

- ▶ A web host provides storage space
- ▶ Must also run a Web Server for files to be accessible
- ▶ For this class we'll use Apache
 - ▶ free
 - ▶ open source
 - ▶ developed and maintained by the Apache Software Foundation (non-profit)
 - ▶ runs on multiple OSs, though mainly on Linux
 - ▶ 60.31% of all web servers are listed as running Apache

AMP Pages

- ▶ Accelerated Mobile Pages
- ▶ Publishing Technology
- ▶ AMP Framework:
 - ▶ AMP HTML
 - ▶ AMP JavaScript
 - ▶ AMP caches
- ▶ Faster load
- ▶ Use less data

Dynamic Generation of Webpages

- ▶ Some pages are static:
 - ▶ they are the same, every time they are served
- ▶ Web servers can also deliver dynamic pages
 - ▶ generated on the fly
- ▶ Advantages of dynamic pages:
 - ▶ user login
 - ▶ interactive sessions
 - ▶ customizing documents
 - ▶ collecting user input
 - ▶ retrieving and updating information from databases
 - ▶ redirection to appropriate pages (for instance mobile pages)

Dynamic Generation of Webpages

- ▶ web server gets a program to compute or generate a document to be returned
- ▶ server-side program is called and passed arguments
- ▶ Active Server Pages
 - ▶ uses ASP, JSP (Java Server Pages), PHP
 - ▶ active parts encoded inside script bracket
 - ▶ embedded in HTML
 - ▶ dynamically computed and inserted in HTML

Database access

- ▶ database access is increasingly important for serving web pages
- ▶ MySQL is a free open-source relational database system
 - ▶ uses tables to organize and retrieve data
- ▶ Many websites provide access to databases
- ▶ Websites also use databases to organize:
 - ▶ user accounts
 - ▶ product inventory
 - ▶ blogging support

Client-side Scripting

- ▶ Can be used to customize user experience
- ▶ Makes pages more responsive
- ▶ JavaScript is the standardized scripting language
 - ▶ but of course there are others:
 - ▶ vbscript (Based on Visual Basic)
 - ▶ tcl (Tool Command Language)
- ▶ Useful for data validation
- ▶ Downside: all your code is visible to client

Web Development Principles

- ▶ It is important to create websites that are:
 - ▶ attractive
 - ▶ functional
 - ▶ user-friendly
- ▶ Start with
 - ▶ what is the website?
 - ▶ what is the audience?
- ▶ Design navigation based on audience and purpose
- ▶ Provide multiple links from different contexts to **vital parts** of site

Web Development Principles

- ▶ Keep it Simple Stupid (KISS)
 - ▶ avoid lots of video and audio
 - ▶ choose familiar and appropriate
 - ▶ fonts
 - ▶ colours
- ▶ use contrast and whitespace
- ▶ ensure fast load
- ▶ use a fluid layout for different screen sizes
- ▶ use open standards
- ▶ test your site on many browsers

In summary

- ▶ We've seen the basic technologies of the Internet and the Web
- ▶ Gained an understanding of how all the technologies work together to make it all work

Homework

- ▶ Read How Stuff Works articles on internet and internet infrastructure
- ▶ Read Section 1 of text The Missing Link: Web Development
- ▶ Do HTML Readings and Tutorial