CSCI335: Software Engineering via Web Applications

Professor Sara Sprenkle

sprenkles@wlu.edu

https://cs.wlu.edu/~sprenkles/cs335/
https://pollev.com/sprenkle

Objectives

- Course overview
- Overview of web
  - Distributed structure
  - Apache web server
- Remote Access
- Version Control

What do you use the www for?

What do you use the www for?

Most Popular Web Sites?

Most Popular Web Sites?

Most Popular Web Sites

April 2024

Website	Domain name	Ranking		Type	Company	Country
		Similarweb (March-24)	Semrush (Jan-24)			
Google Search	google.com	— 1	— 1	Search engine	Google	United States
YouTube	youtube.com	— 2	— 2	Video-sharing platform	Google	United States
Facebook	facebook.com	— 3	— 3	Social network	Meta	United States
Instagram	instagram.com	— 4	▼ 8	Social network	Meta	United States
Twitter	twitter.com	— 5	— 37	Social network	X Corp.	United States
Baidu	baidu.com	— 6	—	Search engine	—	China
Wikipedia	wikipedia.org	— 7	▼ 6	Reference work	Wikimedia Foundation	United States
Yahoo!	yahoo.com	— 8	▲ 12	Web portal	Yahoo!	United States
Yandex Search	yandex.ru	— 9	— 19	Search engine	Yandex	Russia
WhatsApp	whatsapp.com	— 10	▲ 16	Instant messaging	Meta	United States
XVideos	xvideos.com	— 11	▲ 9	Pornography	—	Czech Republic
TikTok	tiktok.com	▲ 12	— 16	Social network	ByteDance	China
Amazon	amazon.com	▼ 13	▼ 13	E-commerce	Amazon	United States
Reddit	reddit.com	▲ 14	▼ 10	Social network	—	United States
Pornhub	pornhub.com	▼ 15	— 4	Pornography	Aylo	Cyprus
Yahoo! Japan	yahoo.co.jp	▲ 16	▲ 17	Web portal	Yahoo!	Japan

https://en.wikipedia.org/wiki/List_of_most-visited_websites

References

- https://en.wikipedia.org/wiki/List_of_most-visited_websites
- https://www.similarweb.com/top-websites/
- https://www.semrush.com/website/top/

Most Popular Web Sites: Alexa

April 2021

Site	Domain	Alexa top 50 global sites (As of February 8, 2021)[1]	Type	Principal country/territory
Google Search	google.com	1 (—)	Internet services and products	United States
YouTube	youtube.com	2 (—)	Video sharing	United States
Tmall	tmall.com	3 (—)	Internet-related services	China
Baidu	baidu.com	4 (—)	Internet-related services and products	China
Tencent QQ	qq.com	5 (—)	Portal	China
Sohu	sohu.com	6 (—)	Portal	China
Facebook	facebook.com	7 (—)	Social networking	United States
Taobao	taobao.com	8 (—)	Online shopping	China
Haosou	360.cn	9 (▲1)	Internet security and web search engine	China
Amazon	amazon.com	10 (▼1)	E-commerce and cloud computing	United States
Yahoo!	yahoo.com	11 (—)	Portal and Media	United States
Jingdong Mall	jd.com	12 (—)	Portal and media	China
Wikipedia	wikipedia.org	13 (—)	Encyclopedia	United States
Zoom Video Communications	zoom.us	14 (▲2)	Videotelephony	United States

Alexa: 3-month average of page views, unique site users

Most Popular Web Sites: SimilarWeb

April 2021

Rank	Website	Category	Change	Avg. Visit Duration	Pages / Visit	Bounce Rate
1	google.com	Computers Electronics and Technology > Search Engines	=	00:11:58	8.91	27.56%
2	youtube.com	Arts and Entertainment > TV Movies and Streaming	=	00:21:53	11.36	21.34%
3	facebook.com	Computers Electronics and Technology > Social Networks and Online Communities	=	00:10:13	8.47	32.85%
4	twitter.com	Computers Electronics and Technology > Social Networks and Online Communities	=	00:10:48	11.73	30.01%
5	instagram.com	Computers Electronics and Technology > Social Networks and Online Communities	=	00:07:34	11.03	35.98%
6	baidu.com	Computers Electronics and Technology > Search Engines	=	00:06:26	8.29	20.39%
7	xvideos.com	Adult	=	00:09:49	8.96	20.79%
8	wikipedia.org	Reference Materials > Dictionaries and Encyclopedias	=	00:03:55	3.00	57.74%
9	xnxx.com	Adult	=	00:08:23	11.15	13.48%
10	yandex.ru	Computers Electronics and Technology > Search Engines	=	00:11:57	9.46	25.28%
11	yahoo.com	News and Media	=	00:07:46	5.80	35.42%
12	amazon.com	E commerce and Shopping > Marketplace	=	00:06:59	8.49	36.45%
13	pornhub.com	Adult	=	00:07:58	7.01	27.14%

https://www.similarweb.com/top-websites/

Most Popular Web Sites

April 2019

Site	Domain	Alexa top 50 global sites (As of January 17, 2019)[3]	SimilarWeb top 50 sites (As of December 2018)[4]	Type	Principal country
Google	google.com	1 (—)	1 (—)	Internet services and products	U.S.
YouTube	youtube.com	2 (—)	2 (▲1)	Video sharing	U.S.
Facebook	facebook.com	3 (—)	3 (▼1)	Social networking	U.S.
Baidu	baidu.com	4 (—)	4 (▲11)	Search engine	China
Wikipedia	wikipedia.org	5 (—)	5 (—)	Encyclopedia	U.S.
Tencent QQ	qq.com	6 (▲2)	18 (▲20)	Portal	China
Taobao	taobao.com	7 (▲4)	48 (▲3)	Online shopping	China
Tmall	tmall.com	8 (▲14)	51 (▲31)	Online shopping	China
Yahoo!	yahoo.com	9 (▼2)	6 (▼2)	Portal and media	U.S.
Amazon	amazon.com	10 (▼2)	11 (▲10)	E-commerce and cloud computing	U.S.
Twitter	twitter.com	11 (▲3)	7 (▼1)	Social networking	U.S.

Most Popular Web Sites: Rank over Time

	Feb 2008	April 2010	April 2013	April 2016	April 2019	April 2021	April 2024
Google	1	2	1	1	1	1	1
YouTube	10	5	4	3	2	2	2
Facebook	9	1	2	2	3	7	3
Yahoo! (mail)	2	3	3	5	9	11	8
Wikipedia	13	16	5	7	5	13	7
eBay	6	12	8	24	?	40	47
MySpace	3	6	??	??	??	???	??

Not quite a fair comparison – different sites/metrics used over time

Who has created a web site?

I have!

Nope, not me!

Who has created a web site?

I have!

0%

Nope, not me!

0%

Who has created a web site?

I have!

0%

Nope, not me!

0%

Who has written a web page?

I have!

Not me!

Total Results: 0

When poll is active, respond at **pollev.com/sprenkle**

Text **SPRENKLE** to **22333** once to join

Who has written a web page?

I have!

Not me!

Who has written a web page?

I have!

Not me!

Discussion: What are web applications?

- Give examples of *static* web pages and *dynamic* web pages/web *applications*
  - What does static vs dynamic mean in this context? (feel free to search the Web)

Terminology

- A *dynamic* web page is a page where at least part of the content is *dynamically generated*, i.e., is generated with code
  - A *static* web page has no dynamic content.

- A *Web application* *is* a set of web pages (usually that set is larger than 1) that is generated by a common code base

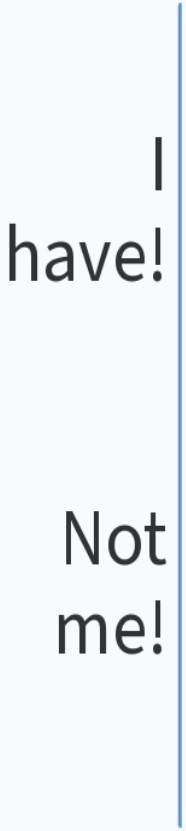
Survey: who has written JavaScript?

I have!

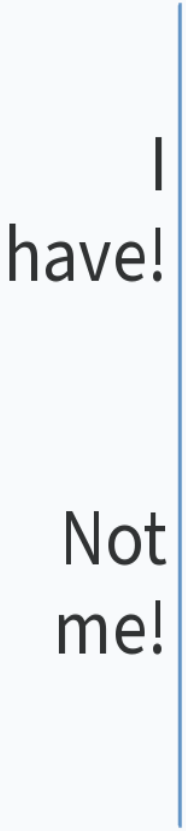
Not me!

Total Results: 0

Survey: who has written JavaScript?



Survey: who has written JavaScript?



Who has written a web application?

I have!

Not me!

Total Results: 0

Who has written a web application?

I have!

Not me!

Who has written a web application?

I have!

Not me!

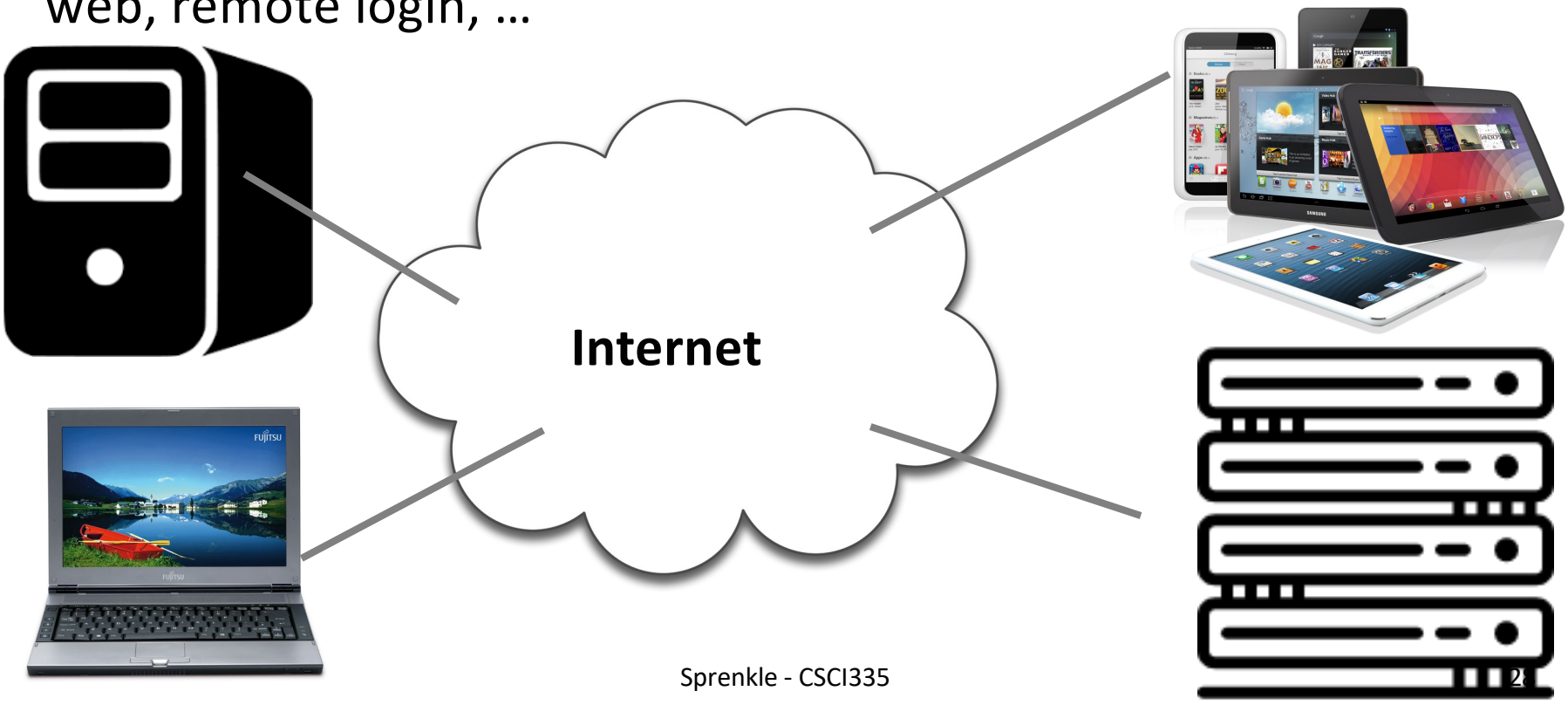
Creation of the World Wide Web

- Result of confluence of
  - Fast computers
  - Internet
  - Hypertext theory, e.g., links to other text

Web's historic logo designed by Robert Cailliau (Belgian CSist)

The Internet

- Connection of computer networks using the Internet Protocol (IP)
- Allows network applications, e.g., email, file transfer, world wide web, remote login, …



Creation of the World Wide Web

- Inventor:
  Physicist Tim Berners-Lee
- 1989 at CERN
- Web: “originally conceived and developed to meet the demand for automated information-sharing between scientists in universities and institutes around the world.”

Creation of the World Wide Web

- Inventor: Physicist Tim Berners-Lee
- 1989 at CERN
- **Vision:** make all information available to all people at all times

Evolution of WWW

- ftp/email ...
- to gopher ...
- to simple html pages ...
- to web sites ...
- to dynamic html ...
- to web commerce …
- to social media, desktop-like applications, ...

What This Course is About

- Web applications
  - Distributed computing
  - Web application technologies (server and client)
  - How to develop high-quality Web applications → *full-stack* development
    - Software tools
- Software engineering
  - Large development project
  - More software, collaboration tools
  - Emulate real-world experience with actual client
- Life-skills
  - Reading, writing, discussion, presentation

Responsibilities

Mine

- Prepare useful, interesting knowledge
- Come to class prepared, on time
- Relevant and challenging assignments
- Feedback on assignments

Yours

- Come to class prepared, on time
- Turn in assignments on time
  - Don't get behind when we only have 4 weeks!
- When you're having trouble
  - Look for help on the Web
    - 1st : use course web site!
    - Find, adapt solutions
    - Give credit to where you found solution, if novel enough
  - Ask me for help
- Learn, absorb, synthesize
  - Extra Credit: take it to the next level

Approach to Learning

- Information: comes from lectures
- Knowledge: comes from readings and assignments
- *Wisdom*: comes from experience

What To Expect

- 4-week, 4-credit Spring term course
  - Fast-paced
  - Demanding
  - But, also a great experience to draw on for interviews
- Programming intensive
  - Individual labs: small-scale, get the basics
  - Team Project: Bring all your collective skills together

Learning Objectives

- Author sophisticated, valid HTML pages using CSS
- Describe the distributed architecture of web applications
- Design and develop a significant web application using server-side technologies, such as Java servlets and JSPs
- Understand the requirements (e.g., usability, reliability, stability, performance, and security) of web applications
- Apply common software tools to developing web applications
- Test and debug web applications, using standard tools (e.g., the Eclipse IDE, Web Tools Platform) and systematic techniques

Course Grade Calculation

- 40%: Individual programming and homework assignments
- 20%: Exam
- 35%: Group Project
- 5%: Participation and attendance

Intro to

DISTRIBUTED SYSTEMS

Distributed Systems

- WWW is a distributed system
- *Distributed System*: multiple machines, communicating over the network via some protocol, working together for some goal
- Common model: clients/servers model

CS Lab Architecture: Authentication

Authentication Server: Active Directory

Active Directory (AD): Microsoft's proprietary directory service

- Runs on Windows Server
- Manage permissions, groups, access to resources

CS Lab Architecture: Authentication

Lab machine

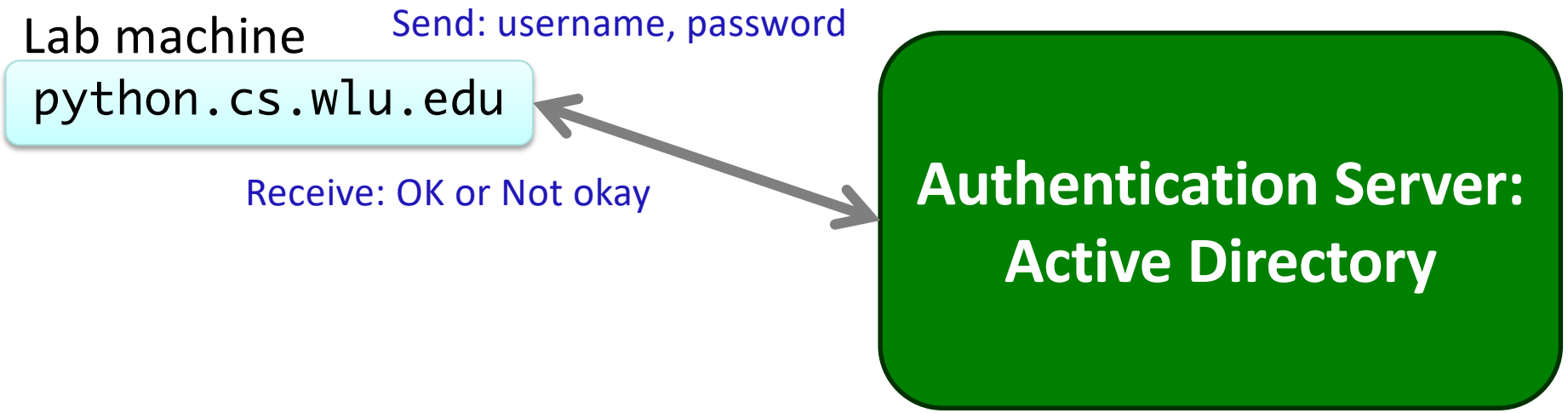
`python.cs.wlu.edu`

Authentication Server: Active Directory

Active Directory (AD): Microsoft's proprietary directory service

- Runs on Windows Server
- Manage permissions, groups, access to resources

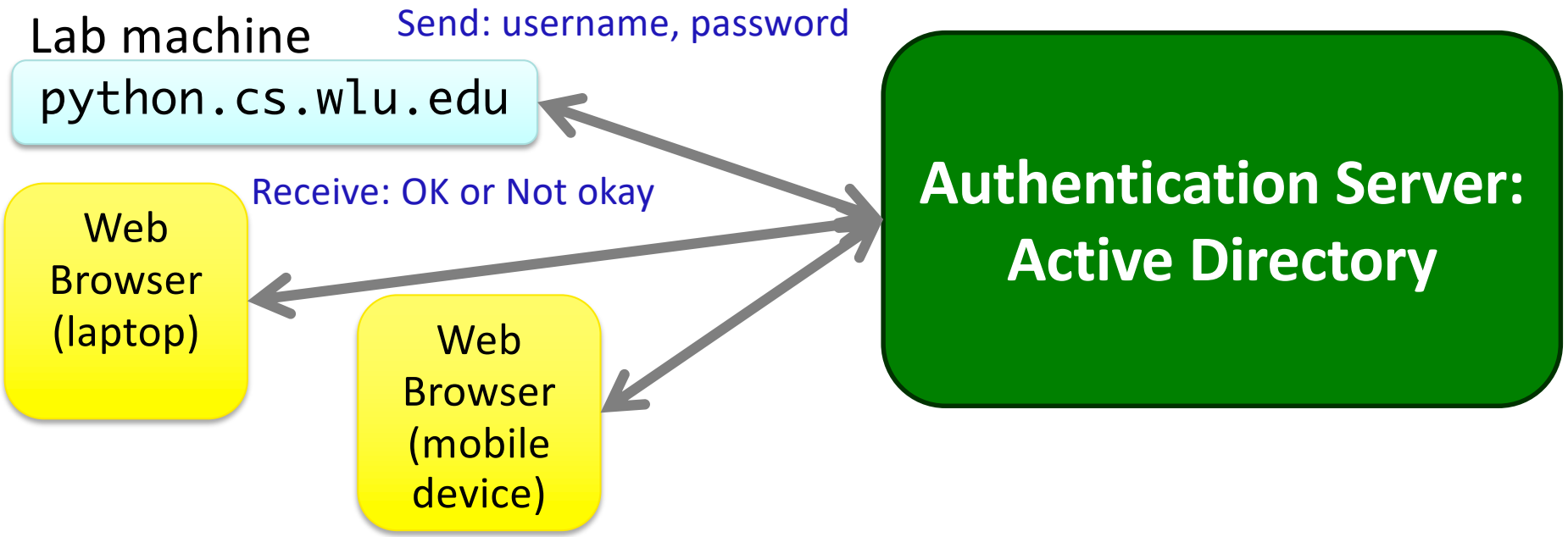
CS Lab Architecture: Authentication



Active Directory (AD): Microsoft's proprietary directory service

- Runs on Windows Server
- Manage permissions, groups, access to resources

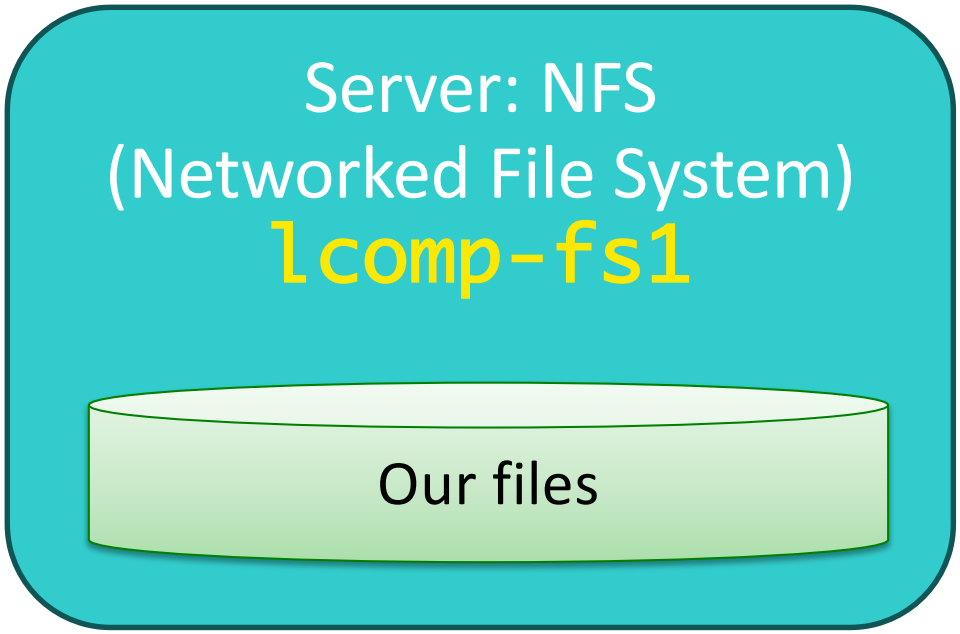
CS Lab Architecture: Authentication



Active Directory (AD): Microsoft's proprietary directory service

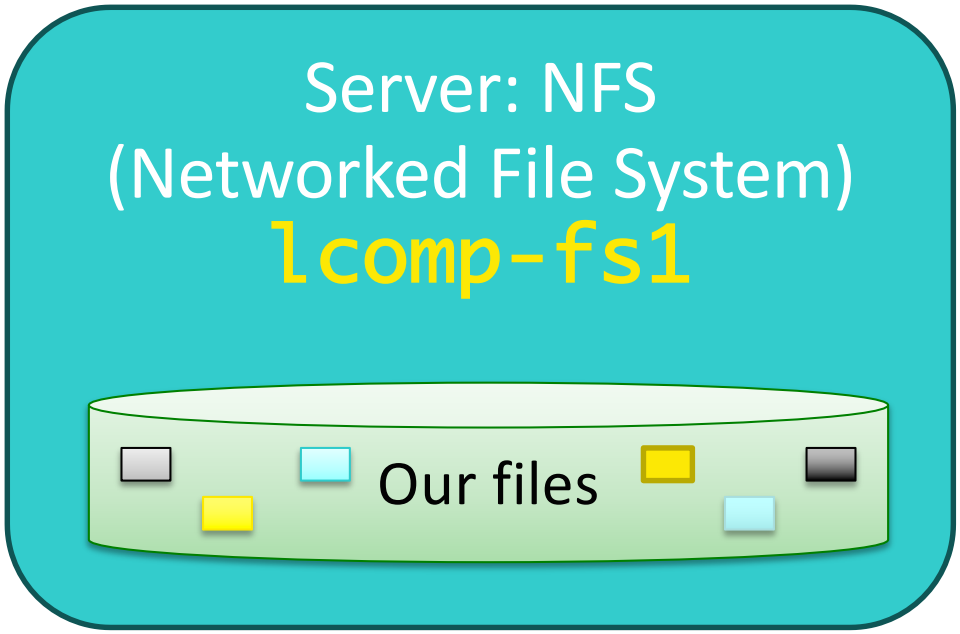
- Runs on Windows Server
- Manage permissions, groups, access to resources

CS Lab Architecture: File Server



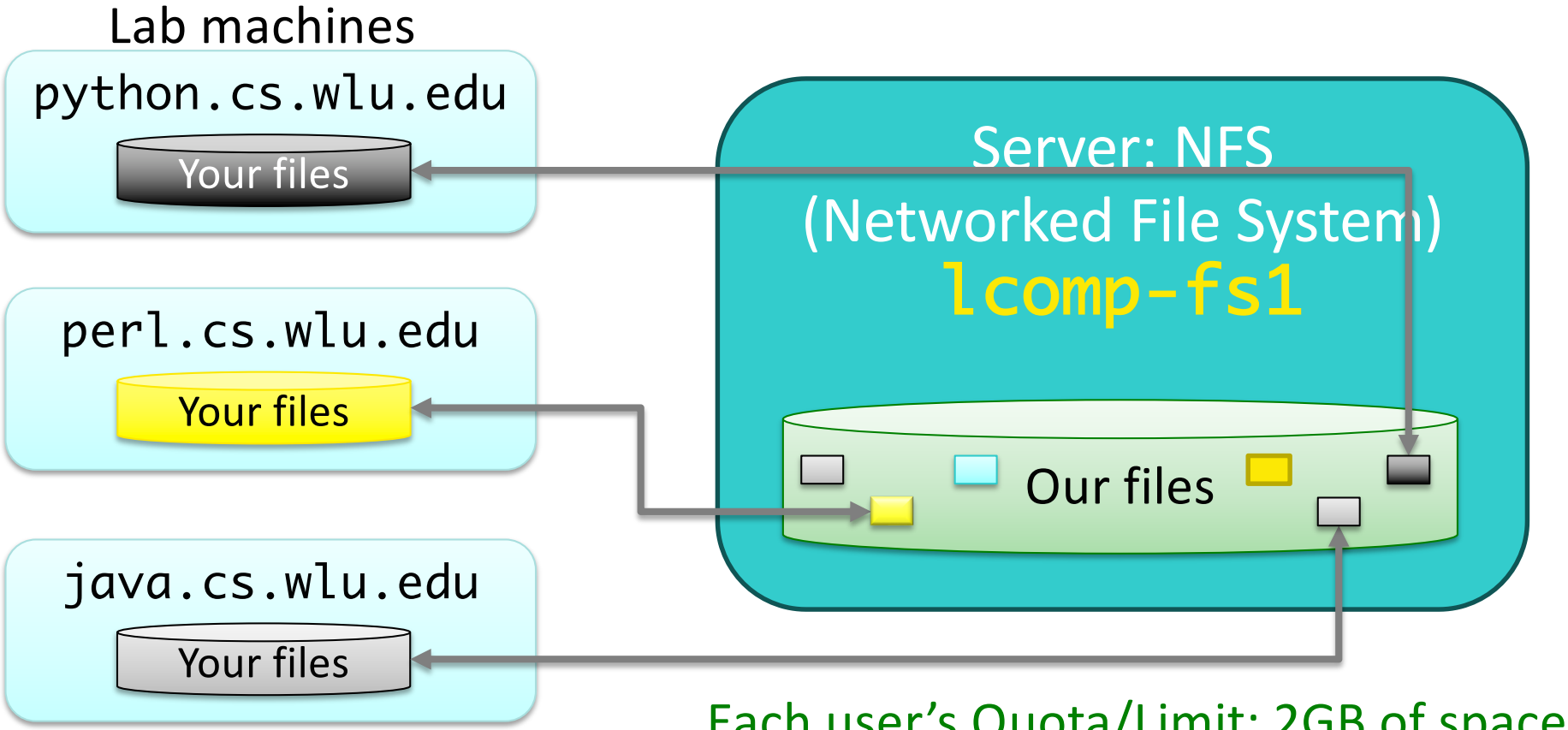
- Stores files for the Computer Science department
  - Individuals' files, shared files for courses, …

CS Lab Architecture: File Server



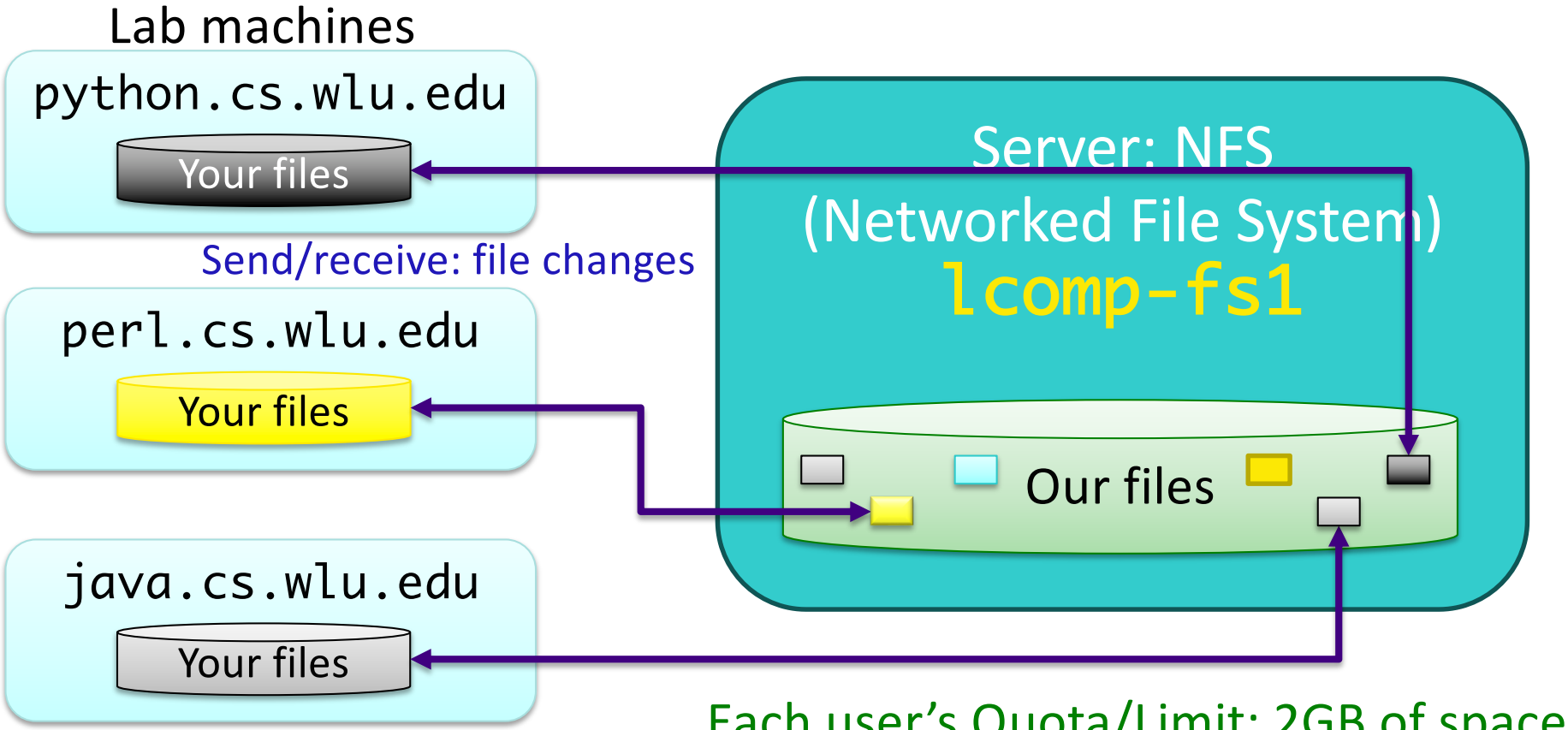
- Stores files for the Computer Science department
  - Individuals' files, shared files for courses, …

CS Lab Architecture



Each user's Quota/Limit: 2GB of space

CS Lab Architecture



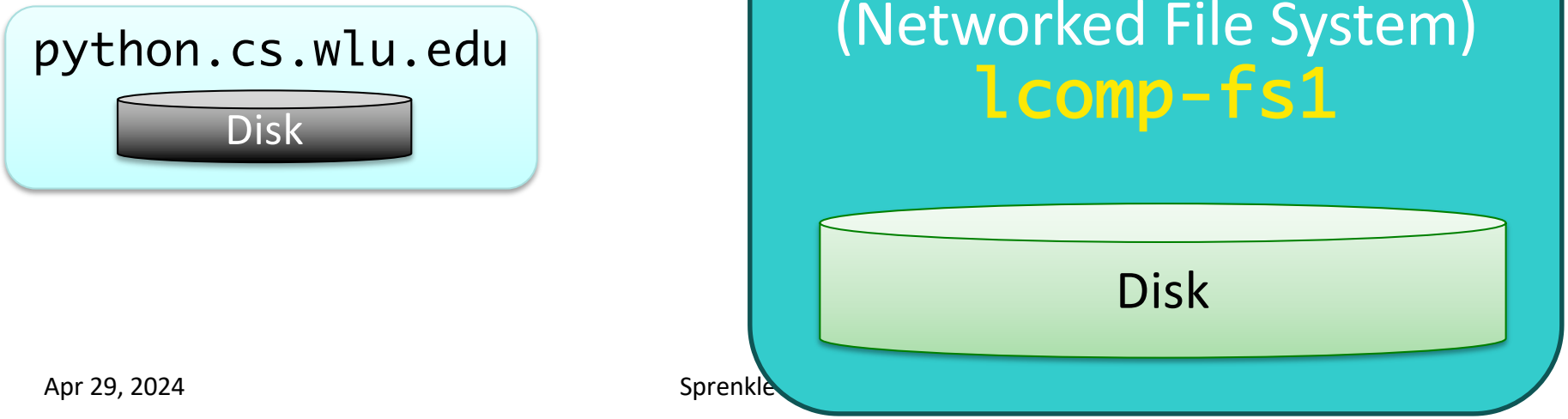
CS Lab Architecture

CS Lab Machine Paths

- `/home/username`
- `/csci/`

CS File Server Paths

- `/export/users/username`
- `/export/csci/`



CS Lab Architecture: NFS Set Up

- On File Server, in /etc/exports

```
/export 137.113.118.0/255.255.255.0(rw,sync,fsid=0,no_subtree_check)
/export/users 137.113.118.0/255.255.255.0(rw,sync,no_subtree_check)
/export/csci 137.113.118.0/255.255.255.0(rw,sync,no_subtree_check)
```

- On Lab Machine, in /etc/fstab

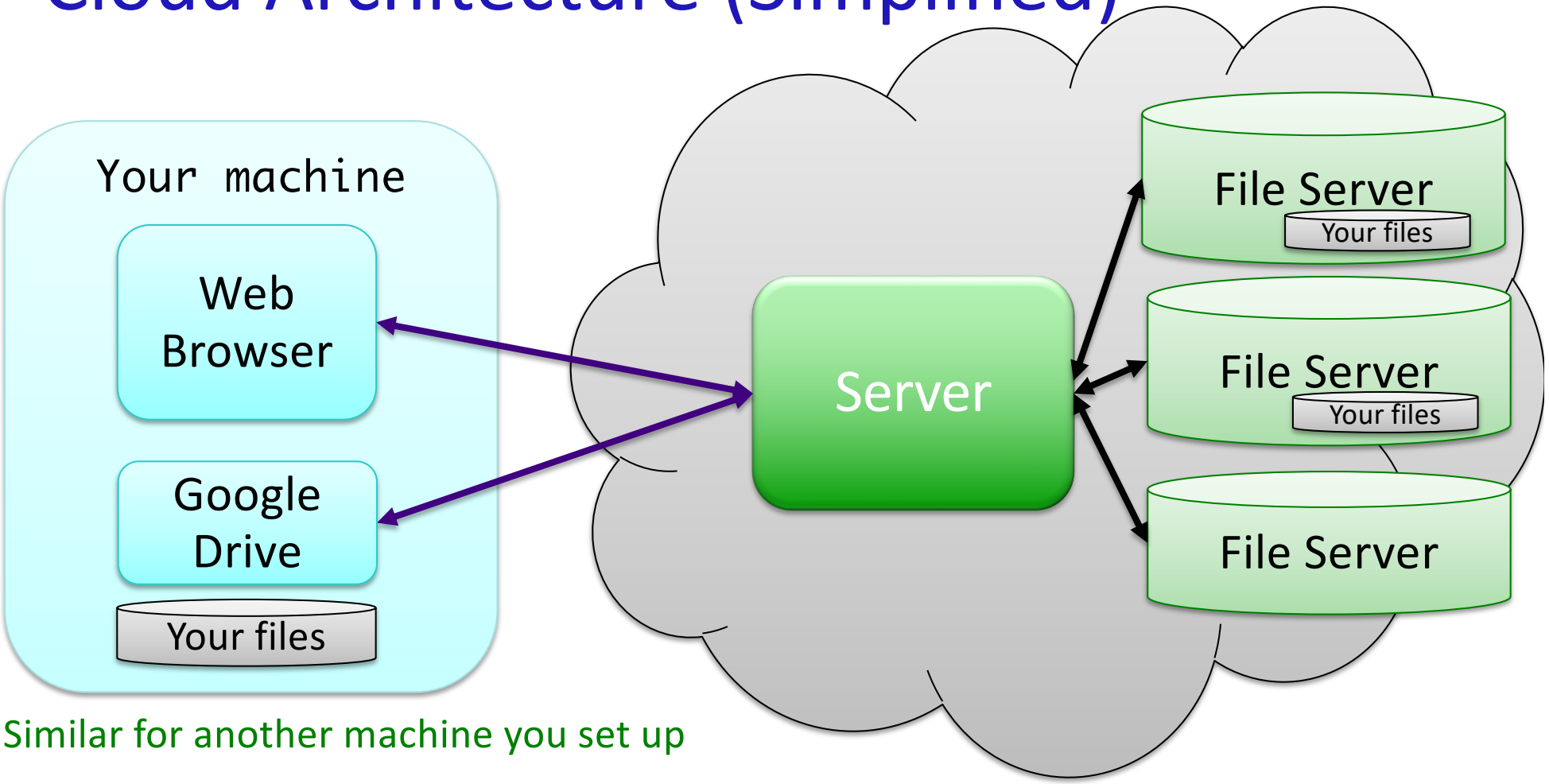
```
lcomp-fs1:/csci /csci nfs defaults 0 0
```

On the Lab Machine

- df (**d**isk **f**ree): report file system disk space usage

```
sprenkles@43374-CSCI-ILAB:/$ df -h
Filesystem                                 Size  Used Avail Use% Mounted on
udev                                       7.7G     0  7.7G   0% /dev
tmpfs                                      1.6G  2.6M  1.6G   1% /run
/dev/nvme0n1p2                              96G   35G   56G  39% /
tmpfs                                      7.7G  196K  7.7G   1% /dev/shm
tmpfs                                      5.0M  4.0K  5.0M   1% /run/lock
tmpfs                                      7.7G     0  7.7G   0% /sys/fs/cgroup
/dev/nvme0n1p3                             137G   28K  130G   1% /mnt/local_scratch
/dev/nvme0n1p1                             511M  6.1M  505M   2% /boot/efi
lcomp-fs1:/csci                            2.0T  801G  1.1T  42% /csci
tmpfs                                      1.6G   20K  1.6G   1% /run/user/125
/dev/loop2                                 106M  106M     0 100% /snap/core/16574
/dev/loop0                                  64M   64M     0 100% /snap/core20/2182
/dev/loop3                                 104M  104M     0 100% /snap/core/16928
/dev/loop1                                  64M   64M     0 100% /snap/core20/2264
lcomp-fs1:/users/sprenkles@ad.wlu.edu      2.0T  801G  1.1T  42% /home/sprenkles@ad.wlu.edu
tmpfs                                      1.6G   40K  1.6G   1% /run/user/1387608248
```

Cloud Architecture (Simplified)



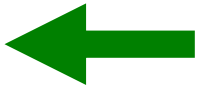
Similar for another machine you set up

Challenges with Distributed Systems

- Multiple sources of error/failure
  - More than one machine
  - The network
  - The communication
- Asynchronous communication, logging
  - Need to keep track of order in which things happened

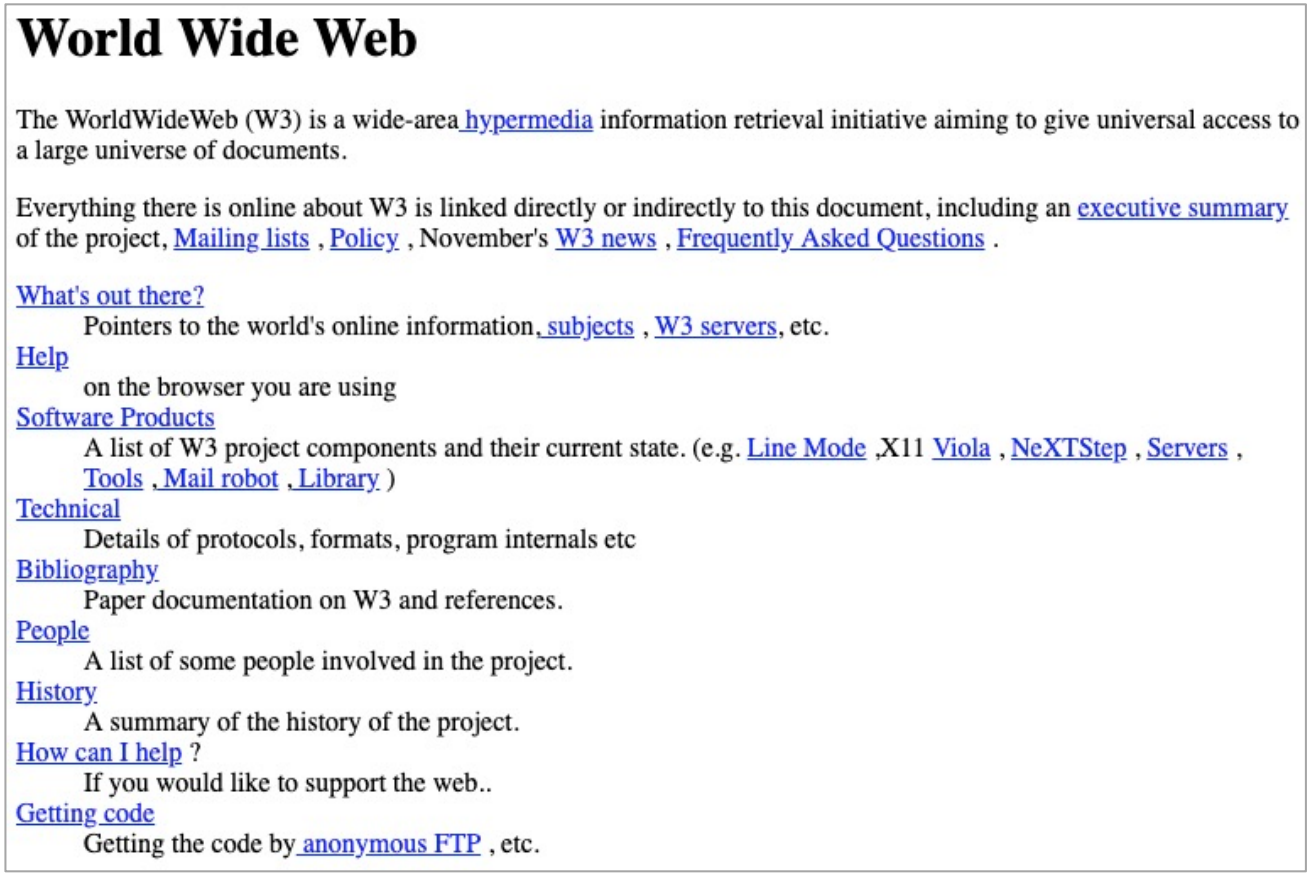
INTRODUCTION TO THE WEB

Evolution of WWW

- ftp/email ...
- to gopher ...
- to simple html pages ...  Where we're starting
- to web sites ...
- to dynamic html ...
- to web commerce…
- to social media, desktop-like applications, ...

The World's First Web Page

April 30, 1993



World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

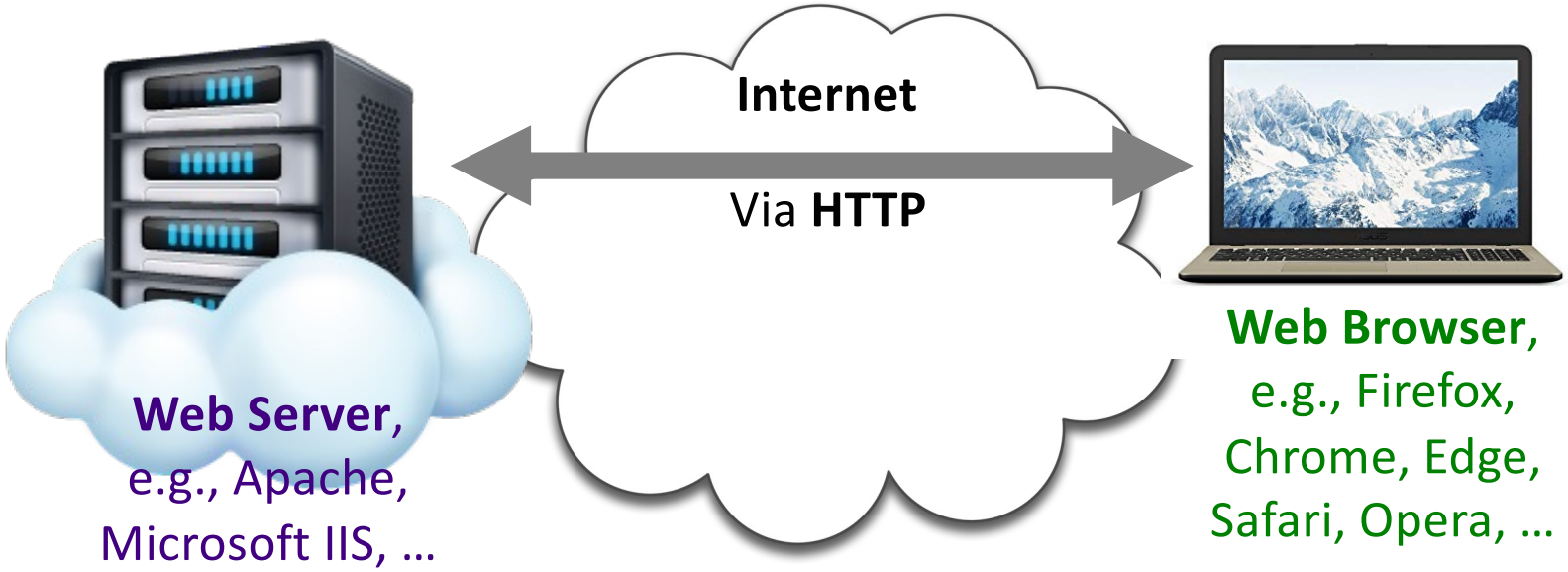
Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
Details of protocols, formats, program internals etc
Bibliography
Paper documentation on W3 and references.
People
A list of some people involved in the project.
History
A summary of the history of the project.
How can I help ?
If you would like to support the web..
Getting code
Getting the code by anonymous FTP , etc.

The Page: `http://info.cern.ch/hypertext/WWW/TheProject.html`

World Wide Web

- Built on top of the Internet
- Web browsers and Web servers
- Communicate using HTTP (Over IP/TCP)



How Does The Browser Get a Page?

- In Web browser, enter a URL
  - URL: Uniform Resource Locator

```
http://cs.wlu.edu
```

```
protocol://host
```

  - May not have explicitly typed in "http"
    - Other protocols: https (becoming the default), ftp

URL: **U**niform **R**esource **L**ocator

- Specifies the location of a *resource*
- Format: `<protocol>://<host>/<path>`
  - Examples:

`https://cs.wlu.edu/~sprenkles/cs335/`

`https://cs.wlu.edu/~sprenkles/cs335/schedule.php`

How Does The Browser Get a Page?

- Look up host's IP Address using DNS
  - Need to be able to find host on the Internet
  - Routing through Internet is by IP address
- Domain Name System (DNS)
  - Set of servers that map domain name to IP Address(es) and vice versa

Naming Lookups

- Unix commands **`host`** and **`nslookup`** can lookup this information

```
$ host cs.wlu.edu
cs.wlu.edu has address 137.113.150.207
```

- And, in reverse:

```
$ host 137.113.150.207
207.150.113.137.in-addr.arpa domain name pointer
lcomp-ws1.
```

Open Systems Interconnection (OSI) Model

- Reference model to help coordinate standards for connection

OSI model

Layer			Protocol data unit (PDU)	Function[27]
Host layers	7	Application	Data	High-level protocols such as for resource sharing or remote file access, e.g. HTTP.
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Frame	Transmission of data frames between two nodes connected by a physical layer
	1	Physical	Bit, Symbol	Transmission and reception of raw bit streams over a physical medium

https://en.wikipedia.org/wiki/OSI_model

How Does The Browser Get a Page?

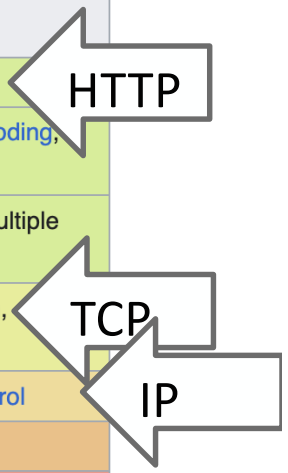
- Browser now makes the request using **HTTP**
  - HTTP: HyperText Transfer Protocol
- Common Types of HTTP Requests:
  - GET: request data/page
  - POST: request data/page, typically given some data
  - HEAD: just get the "header" of a response (not the body)
- For our example, browser makes request `GET /`

Open Systems Interconnection (OSI) Model

- Reference model to help coordinate standards for connection

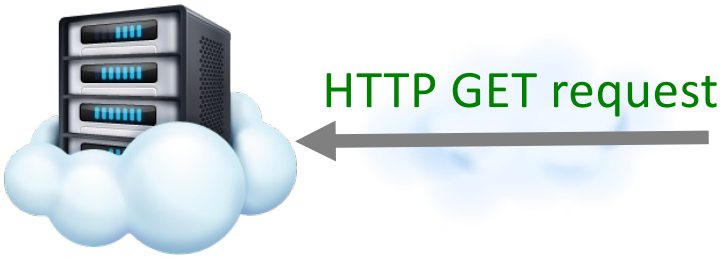
OSI model

Layer			Protocol data unit (PDU)	Function[27]
Host layers	7	Application	Data	High-level protocols such as for resource sharing or remote file access, e.g. HTTP.
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Frame	Transmission of data frames between two nodes connected by a physical layer
	1	Physical	Bit, Symbol	Transmission and reception of raw bit streams over a physical medium

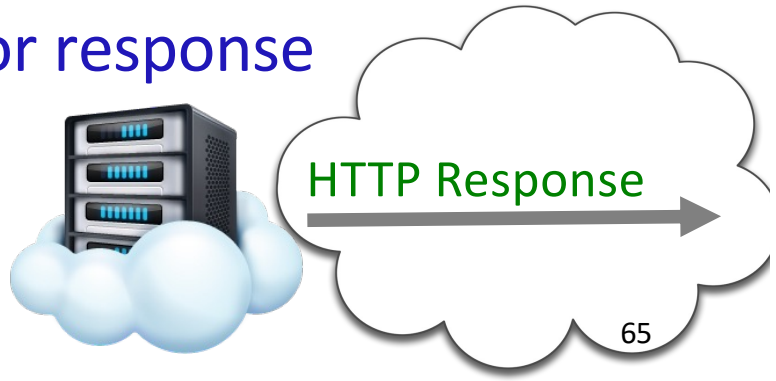


https://en.wikipedia.org/wiki/OSI_model

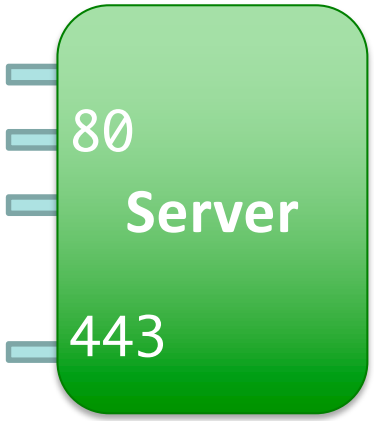
How Does the Web Server Serve a Web Page?



1. Receives request for a *resource* on TCP port 80
2. Looks for the resource in the Web Document directory
   - Not all files on a Web server are meant for others to see
   - Specific directory holds these files
3. If the file is found, server sends an HTTP 200 response with the requested document
   - Otherwise, sends appropriate error response



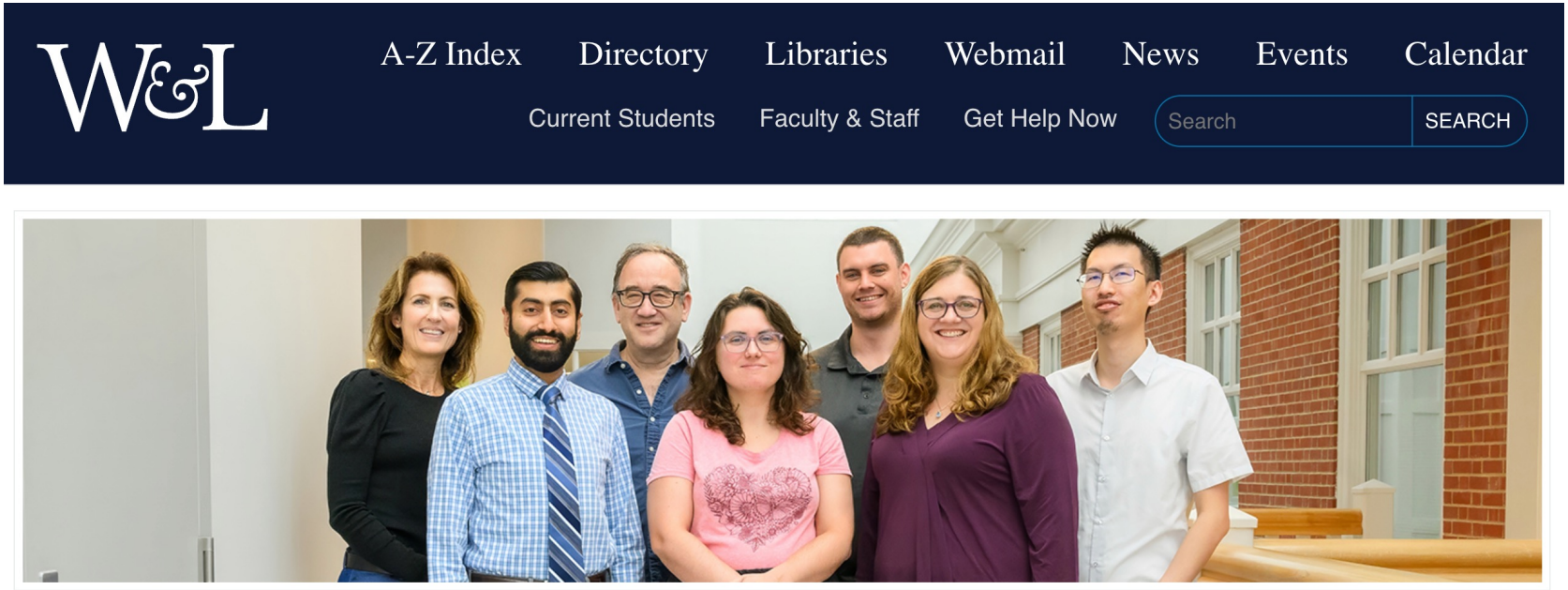
Ports, in Brief



- A ***port*** is an endpoint of communication
- A software abstraction of a physical space through which a client and a server can send messages
  - Think of it like a *mailbox*
  - Servers listen on a port and respond to client requests
- Well-known/Dedicated port: below 1024
  - HTTP: port 80
  - HTTPS: port 443

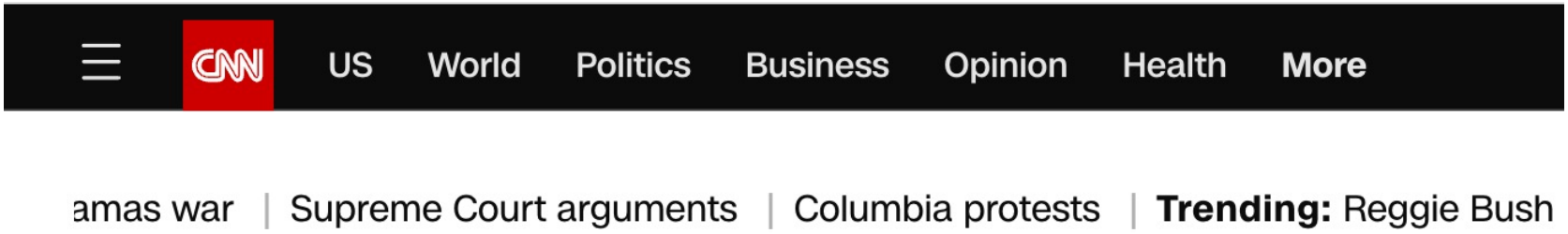
How Does the Browser Get a Page?

- Receives response from server
- Displays file in appropriate format



Evolution of CNN's Site

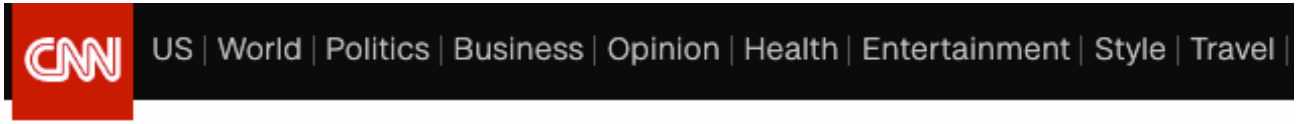
- Recent look



2024:

amas war | Supreme Court arguments | Columbia protests | **Trending:** Reggie Bush

2019:



LIVE UPDATES: Measles outbreak | **TRENDING:** Chaplain hero | Notre Dame damage | Ozarks

2016:

2013:

2010:

2008:

Common HTTP Status Codes

Code	Meaning
200	OK: Request succeeded
3xx	Redirection (temporary or permanent)
403	Error: No permission
404	Error: File not found
500	Internal server error

The Process Without GUI (Browser)

```
sprenkle@perl:~$ telnet cs.wlu.edu 80
Trying 137.113.150.207...
Connected to cs.wlu.edu.
Escape character is '^]'.
GET /
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>302 Found</title>
</head><body>
<h1>Found</h1>
<p>The document has moved <a href="https:///">here</a>.</p>
<hr>
<address>Apache/2.4.41 (Ubuntu) Server at 127.0.0.1 Port
80</address>
</body></html>
Connection closed by foreign host.
```


Not possible over https


APACHE
HTTP SERVER PROJECT

Apache Web Server: httpd

https://httpd.apache.org/

- Open-source HTTP server by Apache Software Foundation
- Colloquially called “Apache”
  - But Apache makes more than the web server
- httpd: **http d**aemon
  - *Daemon* – server, process running in the background
- Most popular web server
- Highly configurable with ***modules***

CS Web Server Set Up:

https://cs.wlu.edu

- CS department web server is `cs.wlu.edu`
- Configuration:
  - `ServerRoot "/etc/apache2"`
    - Config files and modules
    - All readable by viewable
  - `Listen 80`

CS Web Server Configuration

- <Directory />
    AllowOverride none
    Require all denied
  </Directory>

Web server can't access any files

Which files are accessible

- DocumentRoot "/var/www/html"

- <IfModule mod_dir.c>
    DirectoryIndex index.html index.php index.xhtml index.htm
  </IfModule>

If someone accesses a directory, look for file named index.html or …

PHP

- Recursive acronym:

  PHP: Hypertext Preprocessor

- Popular general-purpose scripting language
  - Commonly used for web development
- Fairly easy to write in, security issues
  - I use for my course web pages because it's easy and I don't have security concerns

CS Web Server Configuration: PHP

- Module enabled: php7.4

https://cs.wlu.edu/~sprenkles/cs335/

Allow Users to Have Web Pages

```
<IfModule mod_userdir.c>
        UserDir public_html
        UserDir disabled root
        <Directory /home/*/public_html>
            AllowOverride FileInfo AuthConfig Limit Indexes
            Options MultiViews Indexes SymLinksIfOwnerMatch
IncludesNoExec
            Require method GET POST OPTIONS
        </Directory>
</IfModule>
```

/home/yourusername/public_html

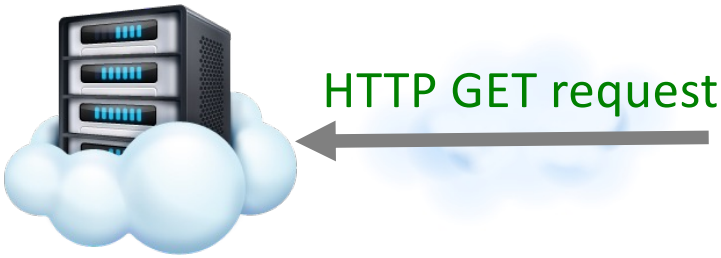
Permissions in Your public_html Directory

- Web server can't access all of your files by default
- Files must have *read* permission by *others*
- Directories must have *read* and *execute* permission by *others*
- Example:

owner group

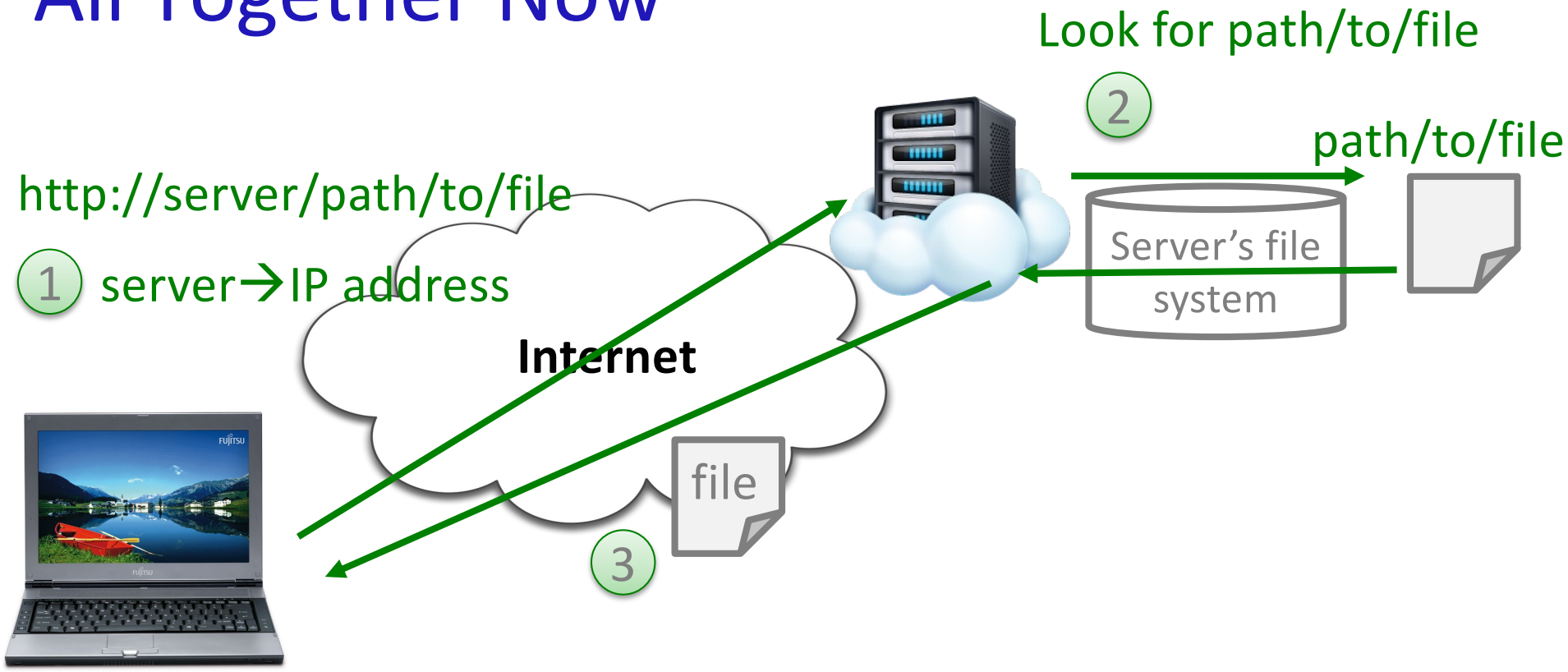
```
-rw-r--r--.  1 sprenkle domain users   494 Apr 23 10:10 index.php
drwxr-xr-x.  3 sprenkle domain users  4096 Apr 23 10:16 css
-rw-r--r--.  1 sprenkle domain users   577 Apr 23 10:16 project.php
-rw-r--r--.  1 sprenkle domain users  6473 Apr 23 10:21 project.html
```

How Does the Web Server Serve a Web Page?



1. Receives request for a resource on TCP port 80
2. Looks for the resource in the Web Document directory
   - Not all files on a Web server are meant for others to see
   - Specific directory holds these files
3. If the file is found, server sends an HTTP 200 response with the requested document
   - Otherwise, sends appropriate error response

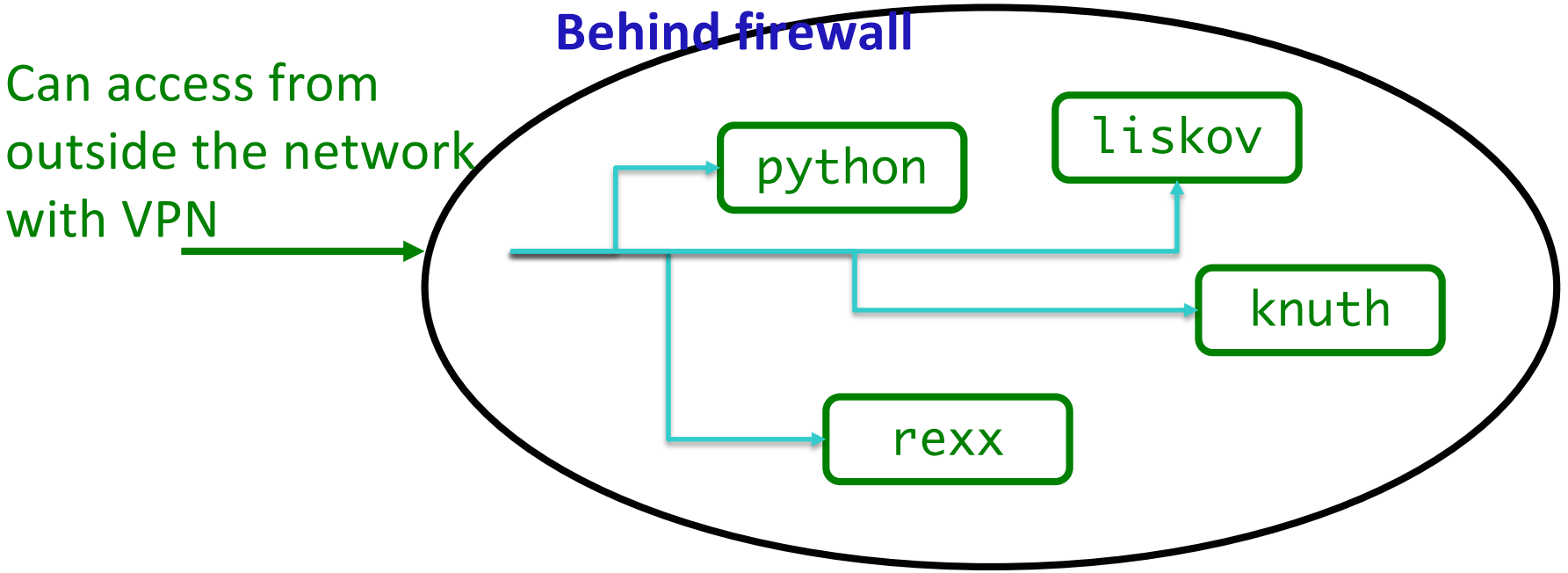
All Together Now



LAB 0

Accessing W&L CS Lab Machines

- Goal: able to work remotely on web pages in your public_html directory



Git and GitHub

- What is Git/version control in general?
- What are the benefits of git (and version control systems, in general)?
- What is GitHub?
- What are the commands we use for git?
- What is our workflow for git?

Version Control Benefits

- Backup and Restore
  - Files are saved as they are checkpointed
  - Revert to a specific version/checkpoint
- Collaboration
  - Lets people share files
  - Stay up-to-date with the latest version
- Track changes to code
  - Save comments explaining why change happened
  - Stored in the VCS, not the file
  - Track how, why a file evolves over time
- Track Ownership
  - Tags every change with the name of the person who made it

Version Control Benefits

- Short-term undo
  - Messed up a file? Go back to the last **good** version
- Long-term undo
  - Created a bug a year ago? Jump back to see change you made.
- Sandboxing
  - Making a big change? Make temporary changes in isolated area, test, work out kinks before "checking in" your changes
- Branching and merging
  - Branch a copy of your code into a separate area, modify it in isolation (tracking changes separately)
  - Later, merge work into common area.

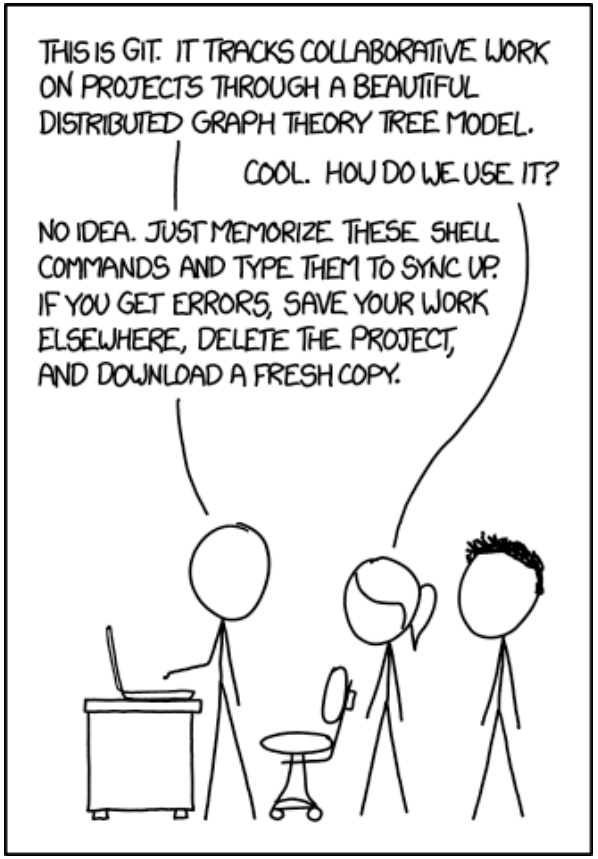
Git & GitHub

- We're going to use Git
  - Distributed version control system
- Our repositories will be hosted by GitHub
  - How you'll get code from me
  - How you'll submit assignments

GitHub's Octocat

Common Git Commands

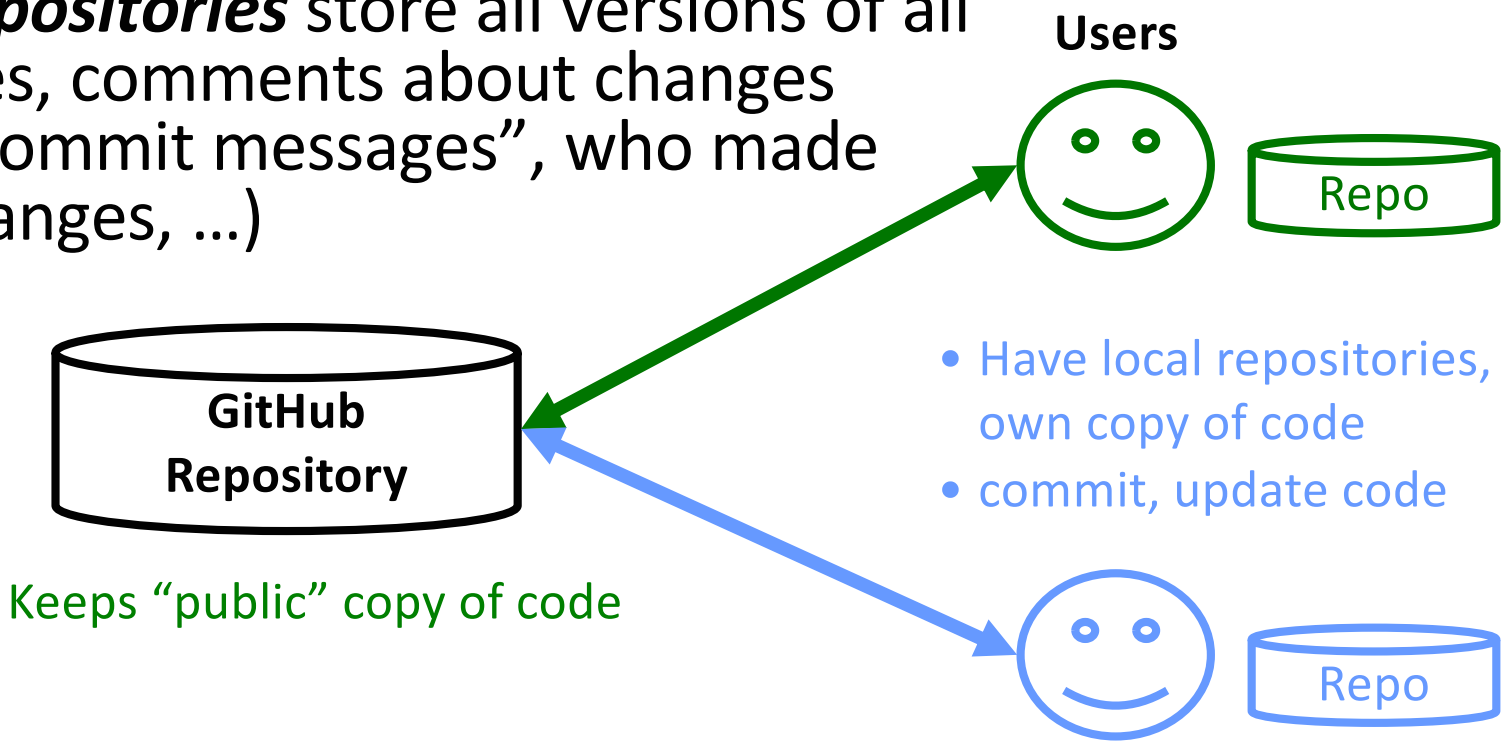
Command	What it does
clone	Clones a repository – sets up your repository so that you can coordinate
add <*file*>	Adds the *file* to the staging area
commit	Commits all the staged files (locally)
push	Push all your changes to the remote → You need your code to be pushed so that I can see it.
branch	List all local branches
branch <*name*>	Creates a new branch named *name*
checkout <*name*>	Switches to the branch named *name*



https://xkcd.com/1597/

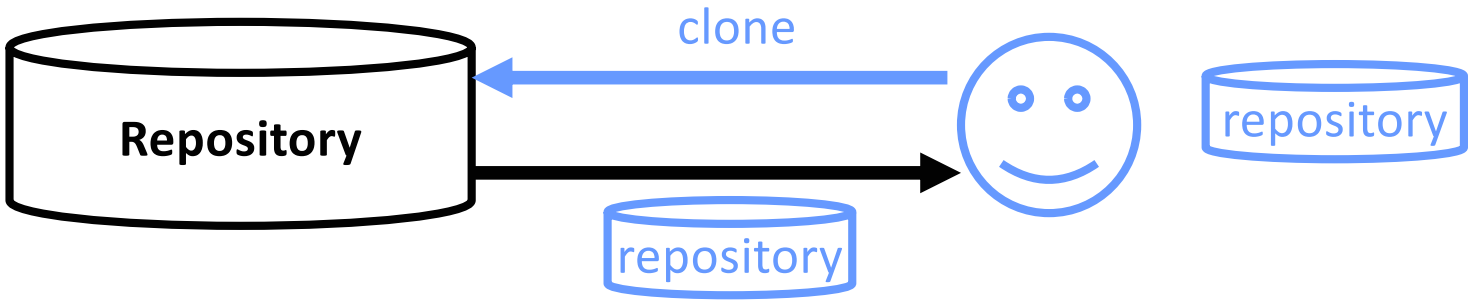
Using Git

- Git is a *distributed* VCS
- ***Repositories*** store all versions of all files, comments about changes ("commit messages", who made changes, …)



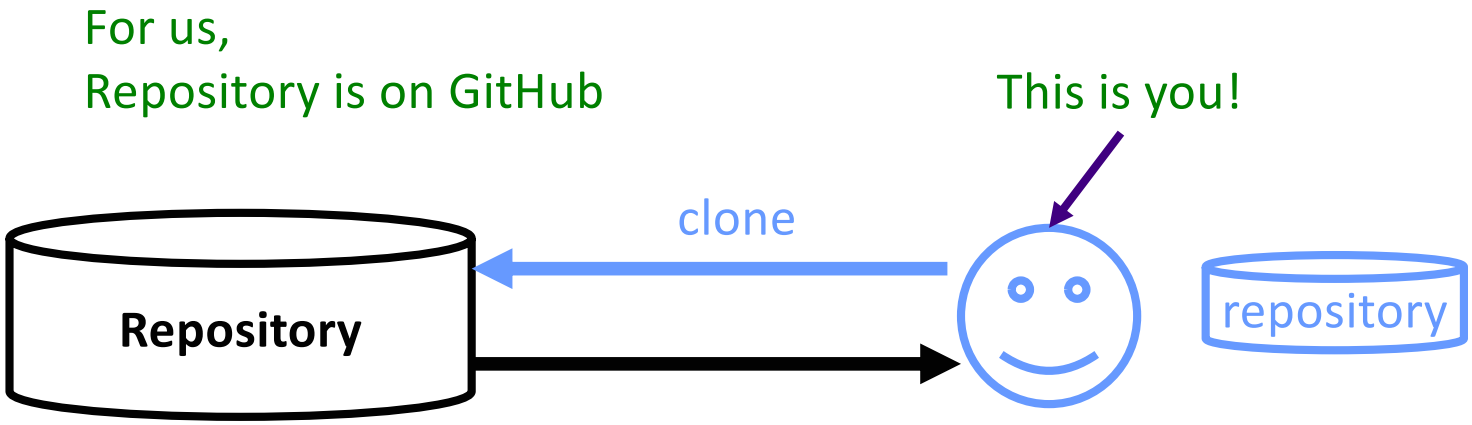
Using Version Control: `clone`

- To start, need to **clone** the repository



Using Version Control: `clone`

- To start, need to **clone** the repository



Using Version Control: `commit`

- After you make changes that you want to document, **commit** your version
  - Include comments about what changes you made and ***why***

  - Updates each modified file
  - Records comments with updated files

Using Version Control: `add, delete`

- You need to **add** and **delete** files and directories to the *staging area*, then **commit**

- Marks the files that will be part of the next commit
- When you commit, these files are added to your local repository

- Add, delete files and directories

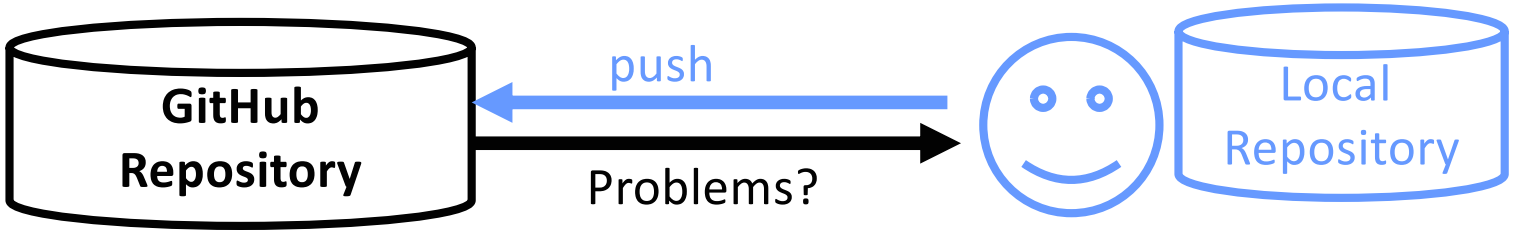
Using Version Control: Commit Messages

- Many different conventions
- Make your messages meaningful and descriptive
  - Emphasis on the *why*
  - Your future self and contributors will thank you
  - Especially as you move on to bigger and better projects

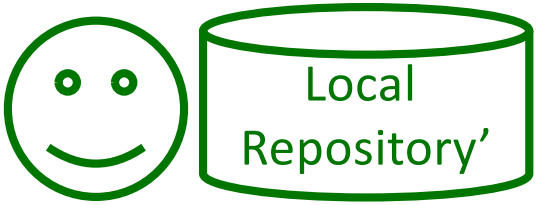
AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Using Version Control: **push**

- After you make changes that you want others (at first, that's just me) to see, **push** your version
  - Sends your previous commits and associated comments



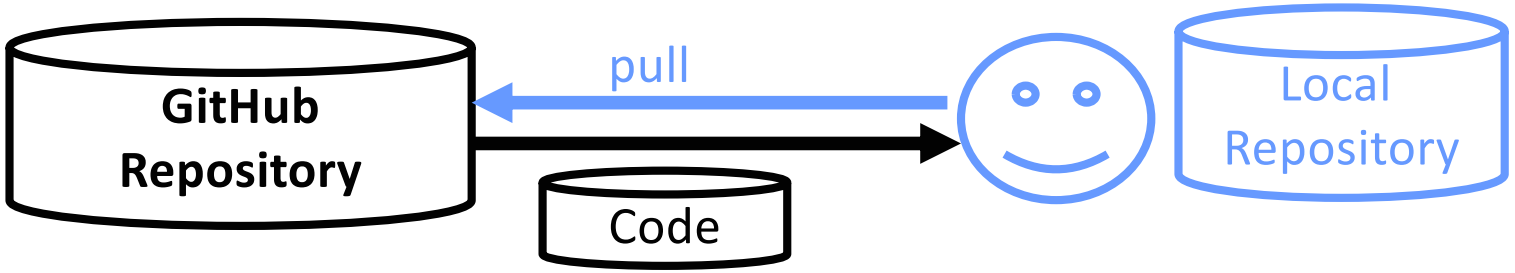
- Updates each modified file
- Records comments with updated files



Other people's code doesn't change

Using Version Control: `pull`

- To see the *current* version of the code in the remote repository, **pull**
  - Resolve conflicts (more on this later this term)



Using Git: Branches

- We create branches when we want to create a new “sandbox” to play in
  - New functionality
  - Bug fixes
  - Different approach

Outline for the Course

- Static Web Pages
  - HTML5, Presentation (CSS)
- Usability
- Dynamic Web Pages, Applications
  - Server-side (Servlets, JSPs, …)
- Testing Web Applications
- JDBC
- Other Java frameworks/tools
- Java Script, AJAX
- JSTL
- Security

Project Development

Planning for the Semester

- Today: typical of semester
  - Part lecture, part discussion, part lab
- Tomorrow
  - HTML

Lab 0

- Remote access to the CS lab machines
- Reviewing (?) Git and GitHub
- Interactive textbook

Looking Ahead

- Lab 0 – due tonight at midnight
- Explore Course Web Site
  - Particular attention to the schedule, examples
  - Read over the project page on the course web site
- Read about how Google search works
  - Write summary in Canvas discussion by Tuesday midnight
- Tomorrow: HTML