CS50 Section 7
Somewhere In Between

Annaleah Ernst, TF

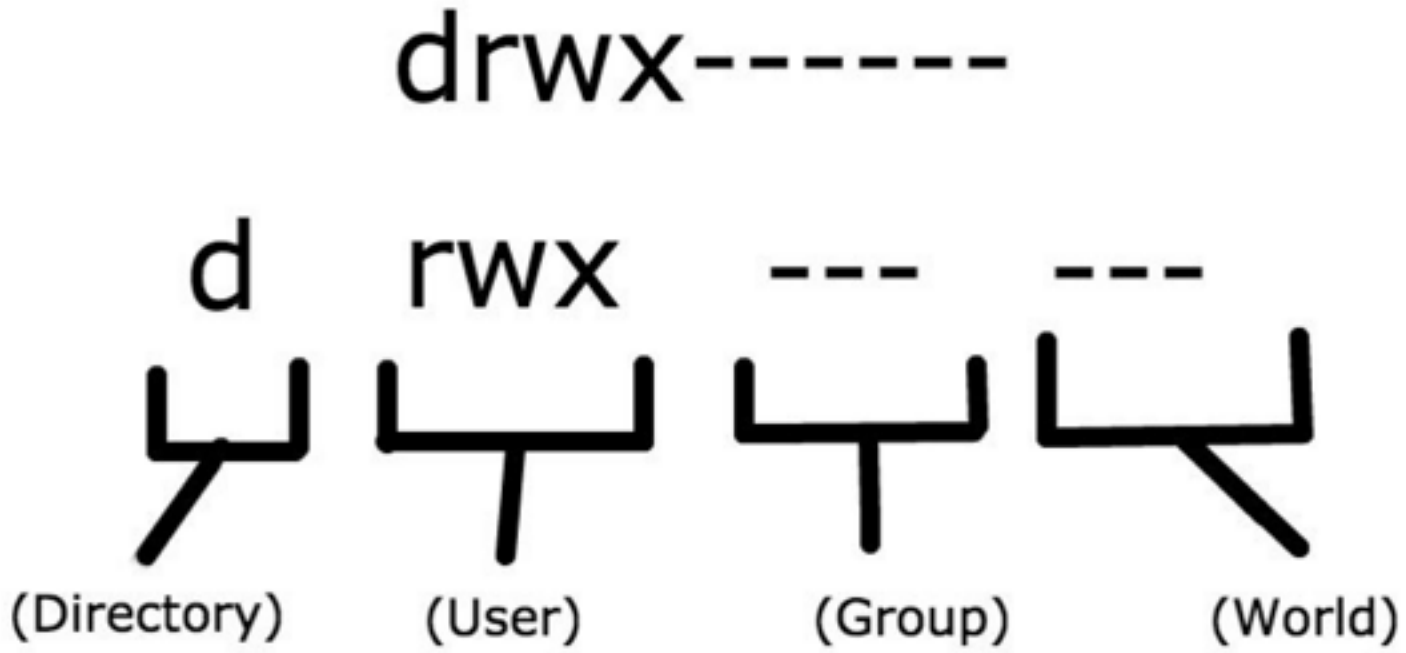
Agenda

Introducing Web Based

- Chmod
- TCP/IP
- Ports
- HTTP
- HTML & CSS
  - Resources!
  - Coding up a Webpage!
  - Instructions on Using the IDE to view webpages!

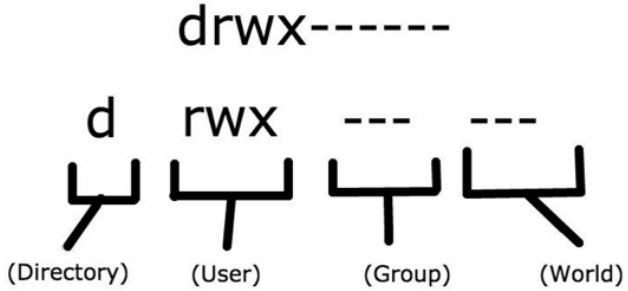
chmod

- Unix system call to change file permissions
- Use `ls -l` to see what permissions a file has



chmod

- We use chmod to change the permissions the file has
- `chmod --- <filename>` (where each dash is a number from 0 to 7)
  - `r` : 4 : readable
  - `w` : 2 : writeable
  - `x` : 1 : executable
  - So that means `rwx = r + w + x = 7`
- The three dashes represent the numeric value of the permission assigned to
  - The user
  - The group
  - The world



chmod

- `chmod <who>+<permissions>`
  - `+` adds permissions
  - `-` takes away permissions
- Permissions
  - `rwx`, `---`, etc
- Who
  - `u` : user or owner
  - `g` : group
  - `o` : others
  - `a` : all

- Or…
- `chmod ---`
  - Where each dash is a number between 0 and 7
  - Eg, 444, 711
- Used for giving potentially different permissions to all three groups at the same time

Your turn!

- Recall that `rwx --- ---` can also be represented as `700`
- What permissions would the following line grant:
  - `chmod 444 <filename>`
  - What's another way we could write this?
    - Universally readable
    - `chmod a+r <filename>`

Your turn! Translate

- Translate the following into the chmod syntax
  - `chmod 555`
    - `chmod a+rx`
  - `chmod u+x`
    - `chmod 100`
  - `chmod 640`
    - `chmod u+rw`
    - `chmod g+r`

chmod – common cases

- `chmod 711 <directory>`
  - For any directory
- `chmod 644 <file>.txt`
  - For any non Python file
- `chmod 600 <file>.php`
  - For Python files

- These are going to be particularly in a couple weeks when we make our own websites

TCP/IP

- Transmission Control Protocol / Internet Protocol
- Means of ensuring data delivery
- Gives set of standards that govern how data should be routed and received
- Increases odds of a data actually getting where you want it to go\
  - Recall David's example from lecture

Ports

- Need to tell our destination what type of data is in our packets
  - Remember, computers aren't dedicated video players – they can do lots of stuff
- Packets might be routed in various ways/paths
- Ports are a way to figure out what type of data is being transmitted
- Common ports:
  - 21 : FTP (File Transfer Protocol)
  - 25 : SMTP (email)
  - 53 : DNS (Domain Name System)
    - What is the IP address of a domain name?
  - 80 : HTTP (webpage)
  - 443: HTTPS (secure webpage)

HTTP

- HyperText Transfer Protocol
- Port 80 (from the previous slide)
- Let’s dissect this a bit..
- HyperText
  - “Text taken to the next level”
  - It’s text that gives us more information
- Transfer Protocol
  - How we request information and how the server responds
  - Governed by specific set of rules\
    - This way the internet will work no matter where you’re using it from

HTTP – example request

- `GET / HTTP/1.1`
- `User-Agent: curl/7.24.0`
- `Host: www.apple.com`
- `<name>: <value>`

Key

- Method Request URI
- Protocol Version
- Field name
- Field value

HTTP – example response

- `HTTP/1.1 200 OK`
- `Server: Apache`
- `Content-Type: text/html; charset=UTF-8`
- `Server: Apache`
- `Content-Length: 16286`
- `Connection: keep-alive`

Key

- Protocol Version
- Status Code
- Field name
- Field Value

HTTP – Status codes

- 200 – OK
- 301 – Moved Permanently
- 302 - Found
- 404 – Not Found
- 403 – Forbidden
- 418 – I’m a Teapot

HTML & CSS

- HTML: HyperText Markup Language
- You get to practice and experiment :}
  - Check out http://www.w3schools.com/ for useful tutorials!
- Best practices
  - Close all tags!
    - Note: real websites often don't, so if you're looking at a page's source for inspiration, be sure not to copy bad habits!
  - Validate your page withy W3 Validator
    - https://validator.w3.org/
  - Separate markup (HTML) and style (CSS)
    - MCV (Model View Controller) paradigm to come!

HTML & CSS

- CSS: Cascading Style Sheets
- Instead of tags, CSS uses attributes
  - Selectors will be used to match tags with attributes
  - Map to tags
- Selectors can be
  - Id: unique
    - `#<idname>` in CSS file
  - Class: can refer to multiple blocks
    - `.<classname>` In CSS file

HTML & CSS: selection code

- reference a tag directly, such as <p>

```
p {
    text-align: center;
}
```

- reference the class, such as <p class="example">

```
.example {
    color: blue;
}
```

- reference its id, such as <p id="main">

```
#main {
    background-image: url("rob_gym.jpg");
}
```

HTML & CSS: Resources

- HTML Cheat Sheet
  - http://lesliefranke.com/files/reference/htmlcheatsheet.html
- CSS Cheat Sheet
  - http://www.lesliefranke.com/files/reference/csscheatsheet.html
- W3schools (for HTML and CSS tutorials
  - http://www.w3schools.com/
- W3validator (run this like you would check50 and style50!)
  - https://validator.w3.org/
- Looking at the source code of pretty webpages, as with Google Chrome's "Inspect Element" tool (just right click on the interesting part of the webpage)

Together…

- Let’s code up a webpage! Grab your computers and get ready to Google – syntax is the name of the game.
  - Check out the attached files for our class website!