!dumb CLI tricks!

(you never knew you wanted)

output: input

\t

recipe

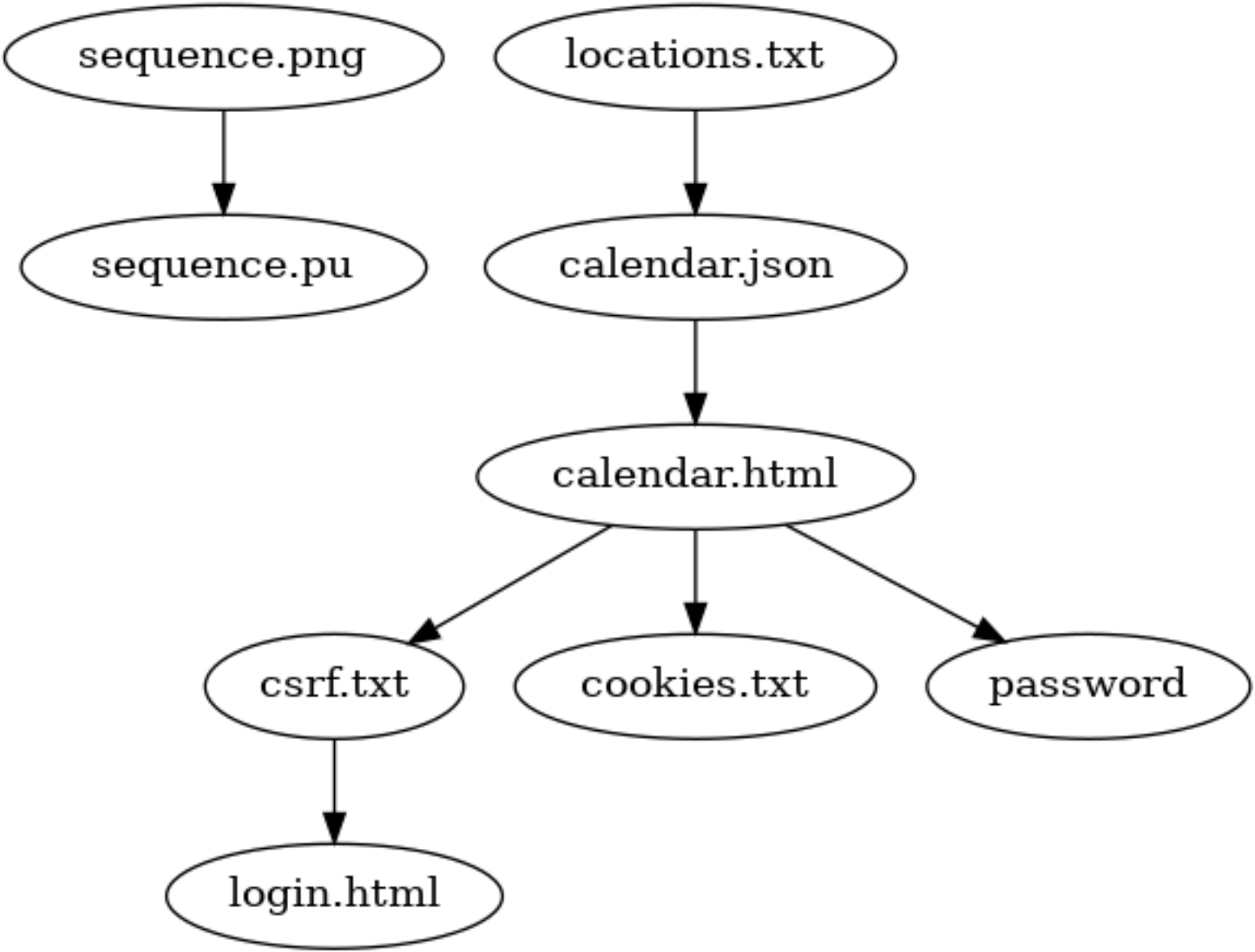
$@ % is a wildcard

Scraping RC Calendar For Fun And Profit

```
# `make` is a build tool. The syntax can be intimidating but it's pretty easy to
# get started. The Makefile consists of rules that tell it how to create outputs
# from inputs. This might be a decent intro:
# https://www.olioapps.com/blog/the-lost-art-of-the-makefile/

# Most of these commands include documentation! Just run `man curl` to get
# documentation for curl, for example.

# This is a variable that is used to login to RC. Change it to yours if you want
# to try running this Makefile!
EMAIL=joe@mou.fo

# If `make` is called without arguments it will run the first rule in the
# Makefile. By convention this rule is usually called `all`. In this case we
# will run the HTTP server, which includes building the slides.
all: server

# `convert` is a command included with ImageMagick for image manipulation.
# Adjust levels to get rid of whiteboard glare, etc.
# https://imagemagick.org/script/command-line-options.php#level
%.png: %.jpg
	convert $< -level 25%,45% $@

# Generate sequence diagrams from text files using PlantUML.
# http://plantuml.com/sequence-diagram
%.png: %.pu
	plantuml $<

# If we try to access the calendar without being logged in, we will be
# redirected to the login page. `curl` downloads it and stores the cookies.
cookies.txt login.html:
	curl -fsSL -o login.html -c cookies.txt https://www.recurse.com/calendar

# The login page includes a Cross-Site Request Forgery token that needs to be
# used to login. `cut` gets the part between quotes, and `tr -d` is used to
# remove trailing whitespace.
csrf.txt: login.html
	grep csrf-token login.html | cut -d\" -f4 | tr -d '\n' > csrf.txt

# Remove trailing whitespace from the password too.
password: rawpassword
	tr -d '\n' < rawpassword > password

# To login we send the CSRF token and cookies we got in the previous steps. We
```

```
# also send a email and password. Note the email is substituted with the
# variable specified at the top of the file.
calendar.html: cookies.txt csrf.txt password
	curl -fsSL -o calendar.html -b cookies.txt https://www.recurse.com/sessions -F email=$(EMAIL) -F 'password=<passwo
rd' -F 'authenticity_token=<csrf.txt'

# To get the JSON embedded in the HTML, we need to unescape some HTML entities.
# We do this using `sed`, which can substitute text using regular expressions.
calendar.json: calendar.html
	grep RC.Calendar calendar.html | cut -d\" -f4 | sed -e 's/&quot;/"/g' -e 's/&amp;/\&/g' -e 's/&lt;/</g' -e 's/&gt;
/>/g' -e "s/&#39;/'/g" > calendar.json

# Python comes with a built-in `json.tool` pretty printer! To get the frequency
# of conference rooms, `grep` to look for occurrences, then `sort | uniq -c`
# to get a count of how many times each room appears, then numerically `sort`
# and `tail` to the 4 lines that are relevant for us.
locations.txt: calendar.json
	python -m json.tool calendar.json | grep -e Sammet -e Hopper -e Turing -e Presentation | sort | uniq -c | sort -n
| tail -n4 > locations.txt

# `dot` is a command in Graphviz, which renders graphs from text files. Here we
# use it to generate a dependency graph of a subset of this Makefile.
%.png: %.dot
	dot -Tpng -o $@ $<

# `enscript` turns text into images (PostScript to be specific).
%.ps: %
	enscript -rp $@ $<
%.ps: %.txt
	enscript -rp $@ $<

# These rules turn images into PDFs.
%.pdf: %.ps
	ps2pdf $< $@
%.pdf: %.png
	convert $< $@

# Make the slides by concatenating all the PDFs together! `pdfunite` is part of
# poppler.
slides.pdf: a.pdf b.pdf seq.pdf Makefile.pdf dep.pdf locations.pdf y.pdf z.pdf
	pdfunite $^ $@

# Python includes an HTTP server! If you are using Python 2, the command is
# `python2 -m SimpleHTTPServer`.
```

```
server: slides.pdf
	python3 -m http.server

# This will delete any generated files! This is helpful when running `make`
# multiple times.
clean:
	git clean -fx -e password

# "Phony" rules don't actually create any output files. They are used as
# convenient ways to run commands related to the build.
.PHONY: all server clean
```


sequence.png
locations.txt
sequence.pu
calendar.json
calendar.html
csrf.txt
cookies.txt
password
login.html

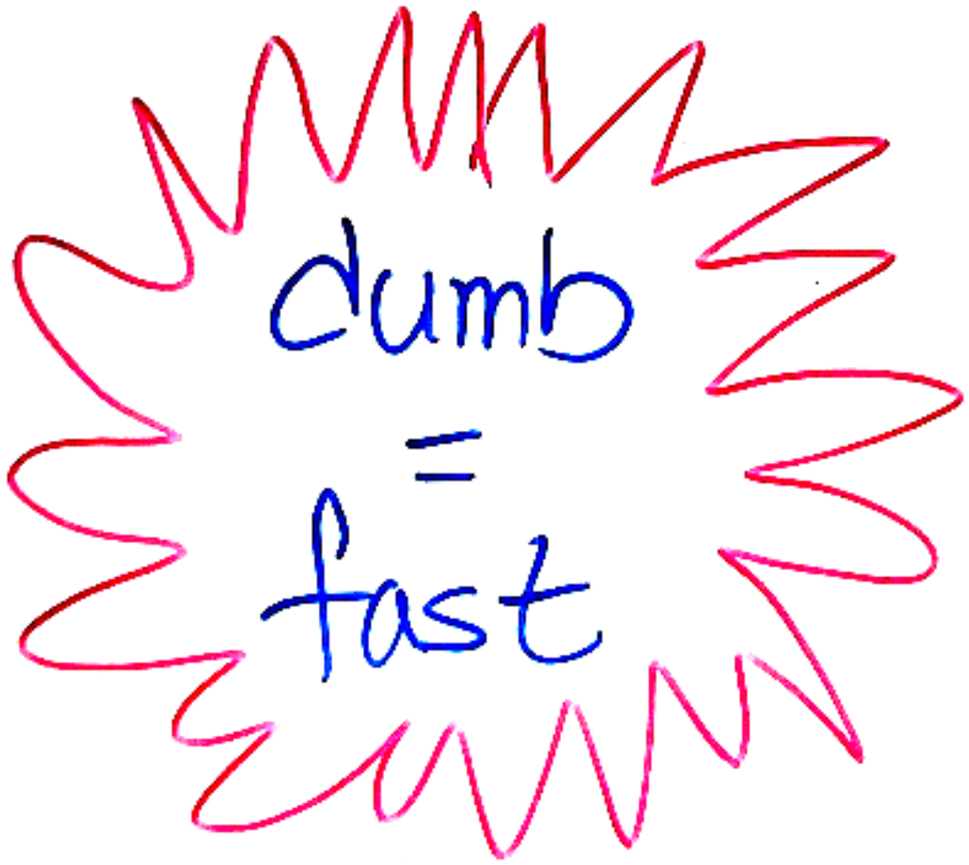
```
 12                "name": "Turing - Recurse Center"
 35                "name": "Hopper - Recurse Center"
100                "name": "Sammet - Recurse Center"
140                "name": "Presentation Space - Recurse Center"
```

learning is FUN!!!

- Convert (ImageMagick)
- make
- curl
- plantuml
- man ascii

- grep/sed/etc (coreutils)
- python -m json.tool
- dot (graphviz)
- python3 -m http.server

http://10.0.17.177:8000/slides.pdf



http://github.com/yoyoyo-jpmo/dumb-cli-tricks