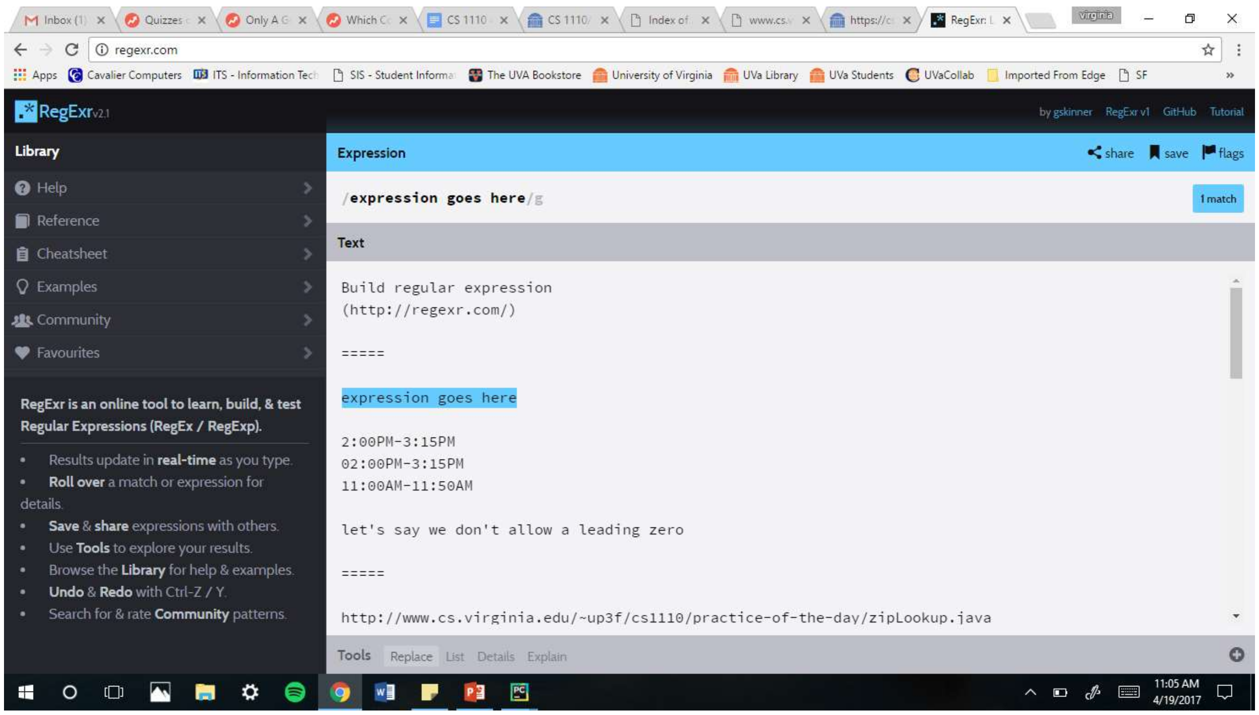
Regex Expressions Cont
Example data to work on in
http://www.cs.virginia.edu/~up3f/cs1110/practice-of-the-day/regex-sample-data.txt
Go to regexr and put this in the big text box, so we can watch what our regex describes
Should look like this (I made it find the expression I put in g):



```
/expression goes here/g
```

RegExr is an online tool to learn, build, & test Regular Expressions (RegEx / RegExp).

- Results update in **real-time** as you type.
- **Roll over** a match or expression for details.
- **Save** & **share** expressions with others.
- Use **Tools** to explore your results.
- Browse the **Library** for help & examples.
- **Undo** & **Redo** with Ctrl-Z / Y.
- Search for & rate **Community** patterns.

```
Build regular expression
(http://regexr.com/)

=====

expression goes here

2:00PM-3:15PM
02:00PM-3:15PM
11:00AM-11:50AM

let's say we don't allow a leading zero

=====

http://www.cs.virginia.edu/~up3f/cs1110/practice-of-the-day/zipLookup.java
```

I also literally put the whole data in there, but you can put specific things if you don't want to scroll through it

Building an expression, with translations; plug them all into regexr and watch it work:

Expression	Meaning
[0-9]	Any one digit
[0-9][0-9]	Any two digits
[0-9]?[0-9]	Any one or two digits
[1-9][0-9]?	Any one or two digits, as long as the first isn't zero
[1-9][0-9]?:[0-9]{2}	A time on an analog clock--one or two digits, the first not zero, then a colon, then two digits
[1-9][0-9]?:[0-9]{2}(PM|AM)	The time, followed by either AM or PM The paran makes it a group, | says either/or

[1-9][0-9]?:[0-9]{2}(PM|AM)-[1-9][0-9]?:[0-9]{2}(PM|AM)	Two times connected by a dash, showing a range

There are links in the regex sample set that take you to more unformatted data

Go to view-source:rabi.phys.virginia.edu/mySIS/CS2/page.php?Type=Group&Group=CompSci
For exam practice, try to get times from this data--if you do that, try putting it in a dictionary--if you do that, try writing a function that gets all classes starting at a certain time

***there is html language here--we aren't expected to know it; regex let's us ignore it

Look at http://www.cs.virginia.edu/~up3f/cs1110/practice-of-the-day/zipLookup.java
This is a java file we want to extract zipcodes, cities, and states from
Plug it into regex and watch it build a regex expression for that:

Expression	Meaning
[0-9]{5}	Any five digits (zipcode)
"[0-9]{5}"	The zipcode with the quotes it's in
[A-Z]{2}	Two capital letters, i.e. the state
[A-Z]+	Any number of capital letters (as long as there's at least one), i.e. the city (if one word
[A-Z]+([A-Z]+)*	The city as one word or one word and any number (0 to many) of extra words after the first separated by a space
[A-Z]{2}, [A-Z]+([A-Z]+)*	The state, then a comma, then a space, then the city, regardless of how many words it has
[A-Z]{2}, [A-Z|\s]+	Exactly the same thing as above, but a shorter way of writing it bc it uses an **|**

Use the data
up3f
sfl7ck
vlb9ae
jrw3mx
kn4vy
hf8va
mp8aa

And recognize the structure of computing ID

Expression	Meaning
[^a-z][a-z]{2,3}	The first half, with two or three lowercase letters, and not more than two or three (there has to be something other than a letter to start)
[0-9]	The digit in the middle
[a-z]{1,2}[^a-z]	The last one or two, and anything other than a letter after it
[^a-z][a-z]{2,3}[0-9][a-z]{1,2}[^a-z]	Everything all together recognizing only computer IDs

Now, put the chunk of links into regexr as our data

class schedule page source

view-source:http://cs1110.cs.virginia.edu/schedule.html#age002

let's take some examples

```
href="http://www.cs.virginia.edu/~up3f/cs1110/examples/regex/â€œ
href="http://www.spronck.net/pythonbook/pythonbook.pdf#chapter.13"
href="http://www.spronck.net/pythonbook/pythonbook.pdf#section.27.3"
href="http://www.spronck.net/pythonbook/pythonbook.pdf#section.17.2"
href="http://cs1110.cs.virginia.edu/know.html"
href="lab11-gamebox.html"
href="files/002/regex_example1.py"
href="http://www.cs.virginia.edu/~up3f/cs1110/lecture-note/"
href="http://www.cs.virginia.edu/~up3f/cs1110/practice-of-the-day/"
href="screencasts/2017-03-27-lecture.webm"
href="testing.html"
href="files/001/20170410b-bounce-speed.png"
href="files/001/2017-04-12-game_over.py"
href="files/fake-queue.csv"
href="files/001/2017-03-27-url_intro_2.pyâ€□
href="style.css"
href="bootstrap.united.min.css"
href="#cal001"
```

Regex to find these links:

Expression	Meaning
[a-z]*:\/{2}	Grabs http://
([a-z]*)(\.|\/)*	Any number of letter groups separated by a dot or a slash (e.g. www.cs.virginia.edu/fjeijfi/fjeij)

Finish this up for complex practice

Also try this easier practice problem, 20 from POTD:
http://www.cs.virginia.edu/~up3f/cs1110/practice-of-the-day/practice_20.txt