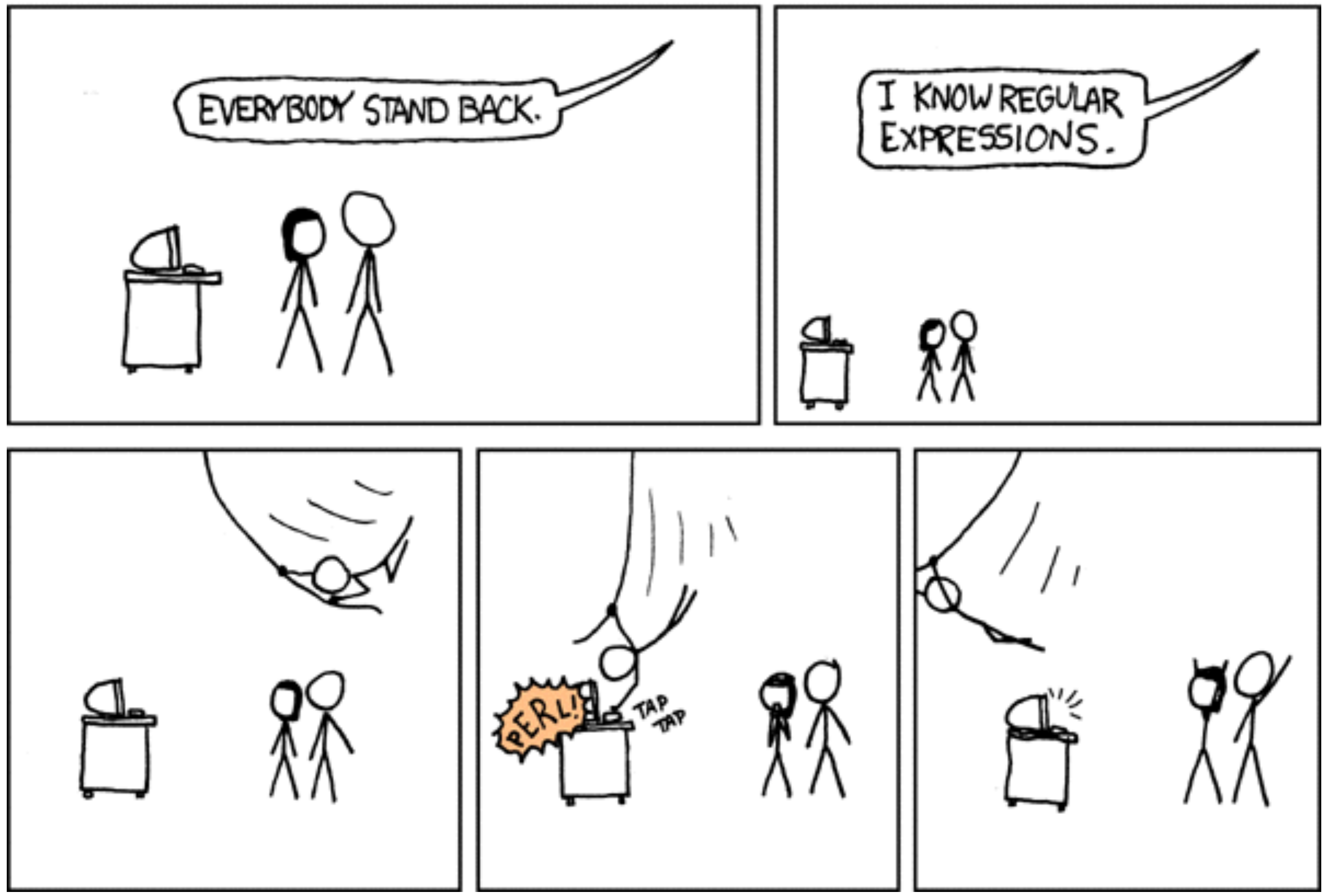
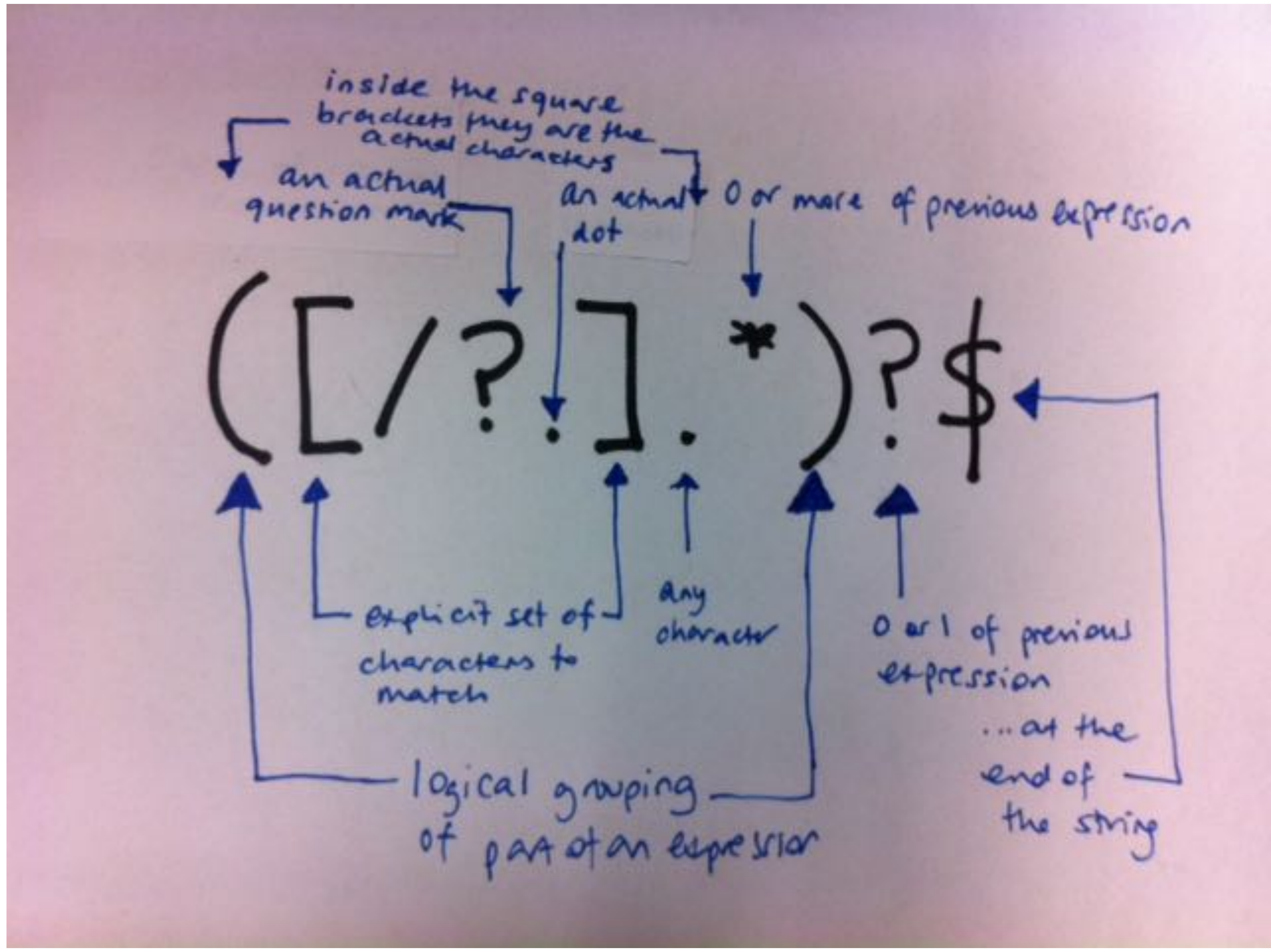
Intro & Regular Expressions

Fall 2018
PCfB Class 1
August 31, 2018

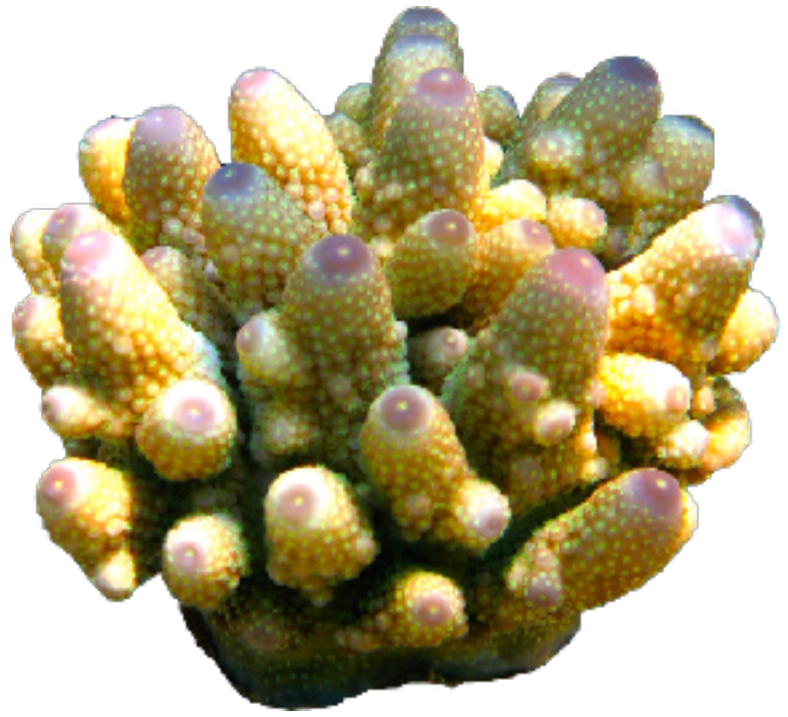



inside the square brackets they are the actual characters
an actual question mark
an actual dot
0 or more of previous expression
([/?.].*)?$
explicit set of characters to match
any character
0 or 1 of previous expression
...at the end of the string
logical grouping of part of an expression

Intros

PhD - Evolutionary genetics

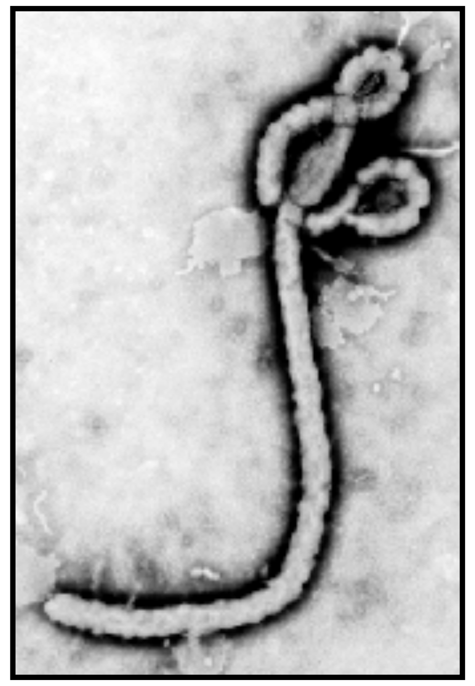
HOPKINS
MARINE STATION
OF STANFORD UNIVERSITY



PostDoc - Pathogen genomics

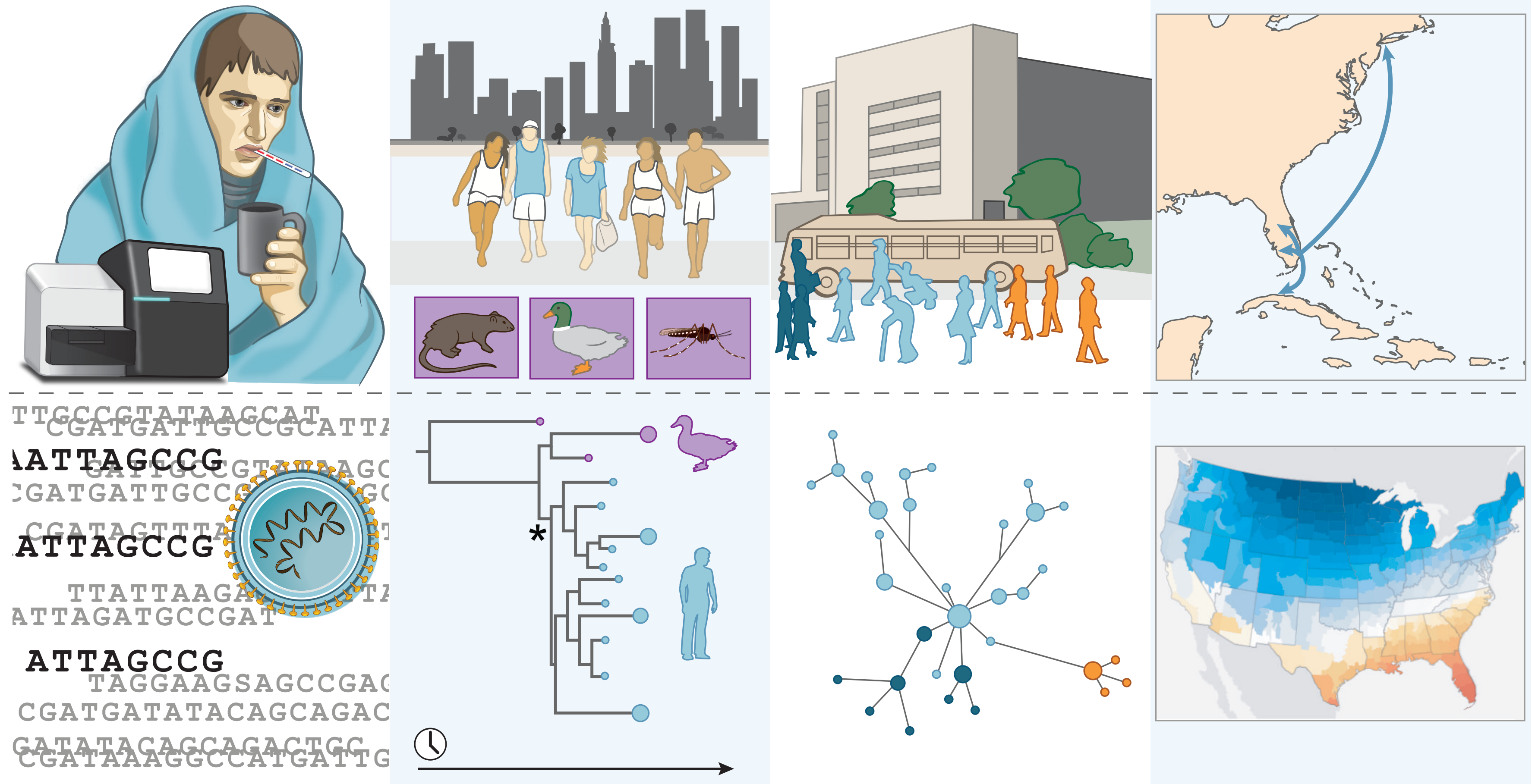
USAMRIID
United States Army
Medical Research Institute
of Infectious Diseases

Biodefense solutions to protect our nation



PATHOGEN & MICROBIOME INSTITUTE
NAU & TGen

Assistant Professor
Dept of Biological Sciences
Pathogen and Microbiome Institute



Intros

1. **Your name**
2. **Your research focus**
3. **What you hope to get from this class**

What this course is:

- Intro to general computing techniques broadly applicable to many research-related tasks

What it isn’t:

- A bioinformatics class

Syllabus

(https://github.com/jtladner/Courses/tree/master/PracticalComputing/Fall_2018)

GitHub

Grading

This seminar is pass/fail. There will not be graded assignments. However, **attendance and participation are required.**

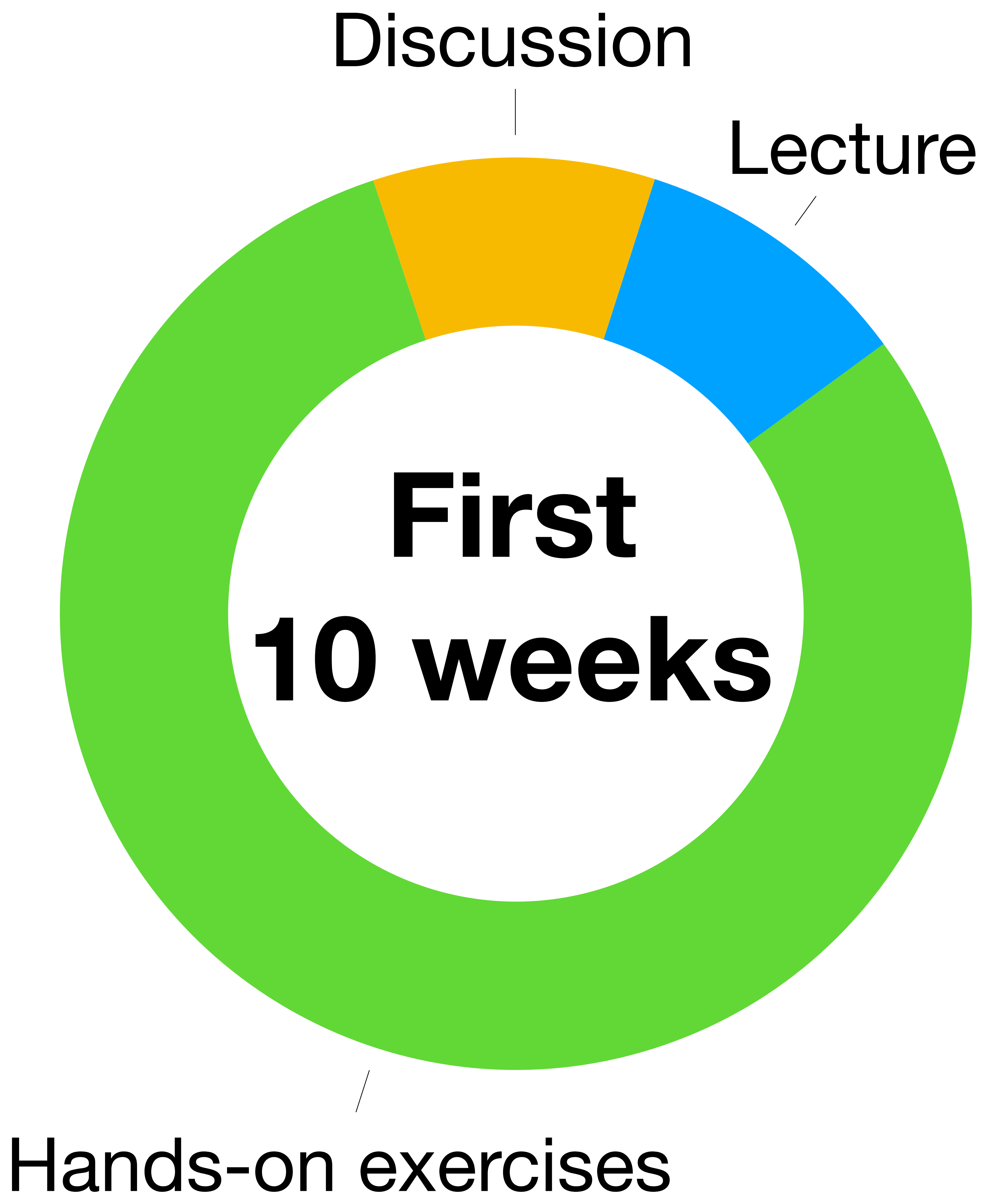
Required text

- Haddock, S. H. D. and Dunn, C. W. (2010). Practical Computing for Biologists. Sinauer Associates

- http://practicalcomputing.org/

- Reading must be complete **PRIOR** to class

Class organization



"Hack-a-thons"
(Last 4 weeks)

- Work in groups to write custom scripts to solve real world problems
- Please submit ideas for problems from your own research
    - Must be willing to share associated data with class

The Learning Studio

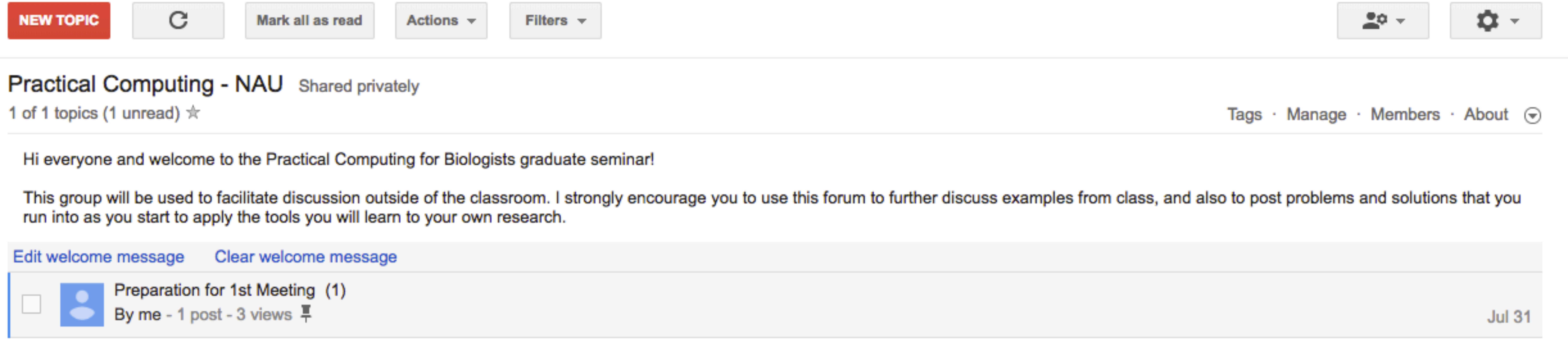
https://nau.edu/library/learning-studio/

Cline Library, Room 249

Nov. 9th, 16th, 30th and Dec. 7th

Google group

- For discussion, outside of class, regarding topics covered in class:
  - questions
  - solutions
  - brainstorming



Practical Computing - NAU Shared privately

1 of 1 topics (1 unread) ☆

Hi everyone and welcome to the Practical Computing for Biologists graduate seminar!

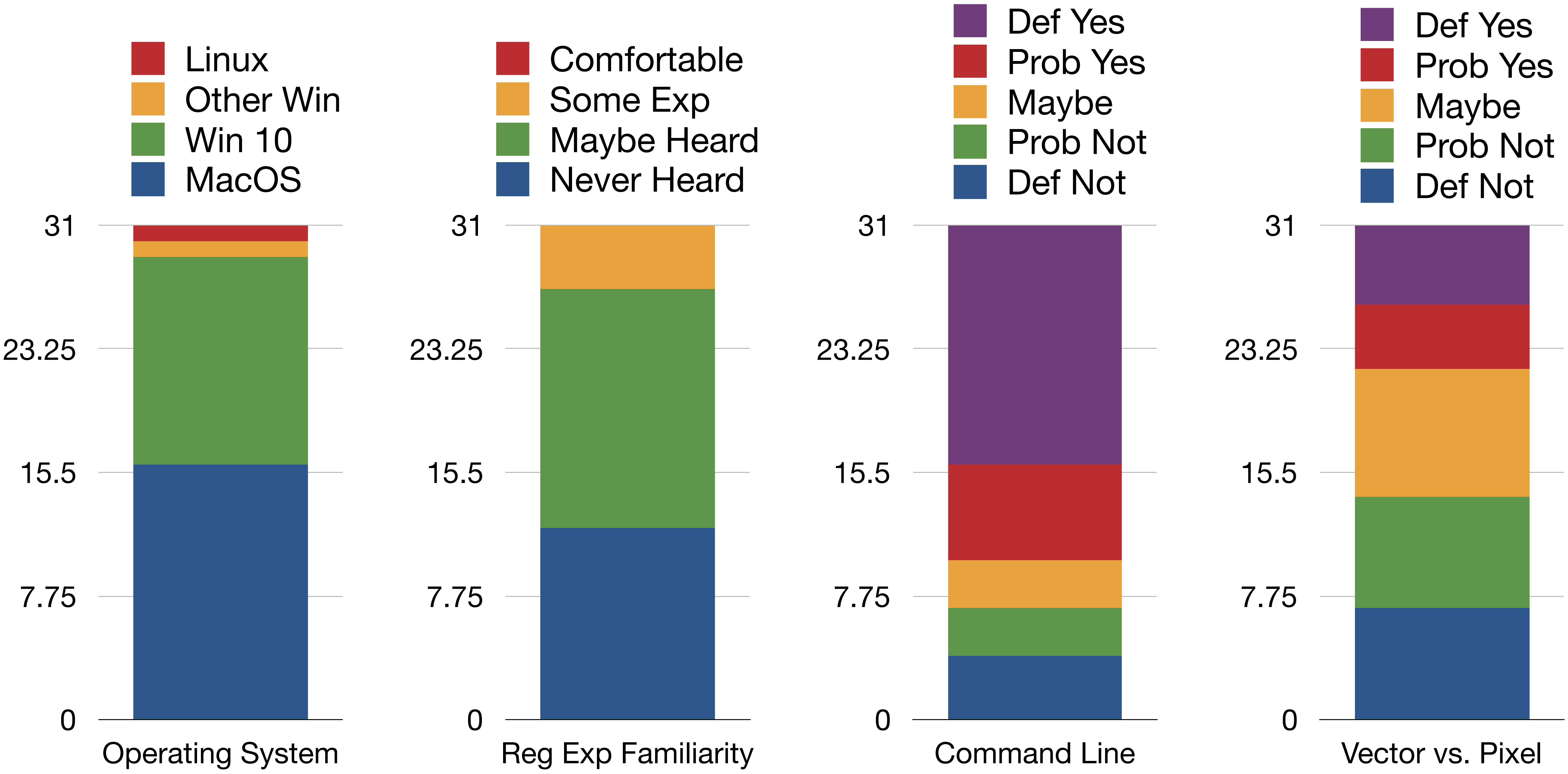
This group will be used to facilitate discussion outside of the classroom. I strongly encourage you to use this forum to further discuss examples from class, and also to post problems and solutions that you run into as you start to apply the tools you will learn to your own research.

□ Preparation for 1st Meeting (1)
By me - 1 post - 3 views Jul 31

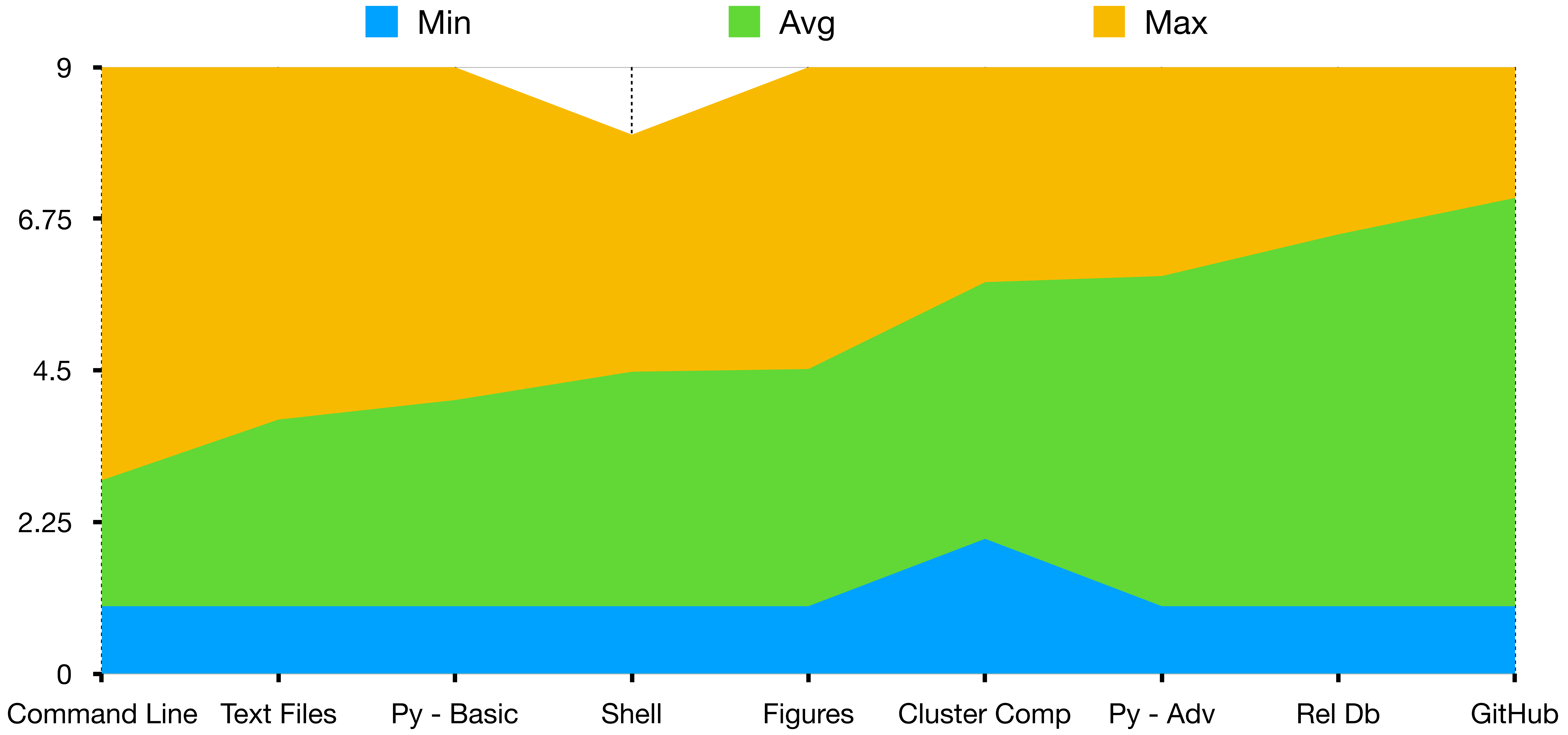
Absences

- Let me know when you will have to miss class
- Expected to work through exercises on your own
- Expected to post something related to the class topic within the google group

Pre-class survey



Areas of interest



Computer setup

(https://github.com/jtladner/Courses/tree/master/PracticalComputing/Fall_2018/Getting%20Started)

- Text Editor
- GitHub Repository
- Command line terminal

Plain text file

- “Pure sequence of character codes”
    - each character is actually represented by a number
- No formatting (e.g., text size, color, font, spacing)
- Human and machine readable
- Standardized

End of line charcters

- Line feed (LF, \n) - Mac OSX, Linux
- Carriage return (CR, \r) - Mac OS9 and earlier
- Carriage return + line feed (CRLF, \r\n) - Windows

Which of these formats are NOT plain text?

Excel (.xlsx)

html

OpenOffice (.odf)

Google Sheet

text (.txt)

fasta

markdown

xml

Google Doc

nexus

json

Word (.doc)

python script (.py)

rich text (.rtf)

phylip

Which of these formats are NOT plain text?

Excel (.xlsx)

html

OpenOffice (.odf)

Google Sheet

text (.txt)

markdown

fasta

xml

nexus

Google Doc

json

Word (.doc)

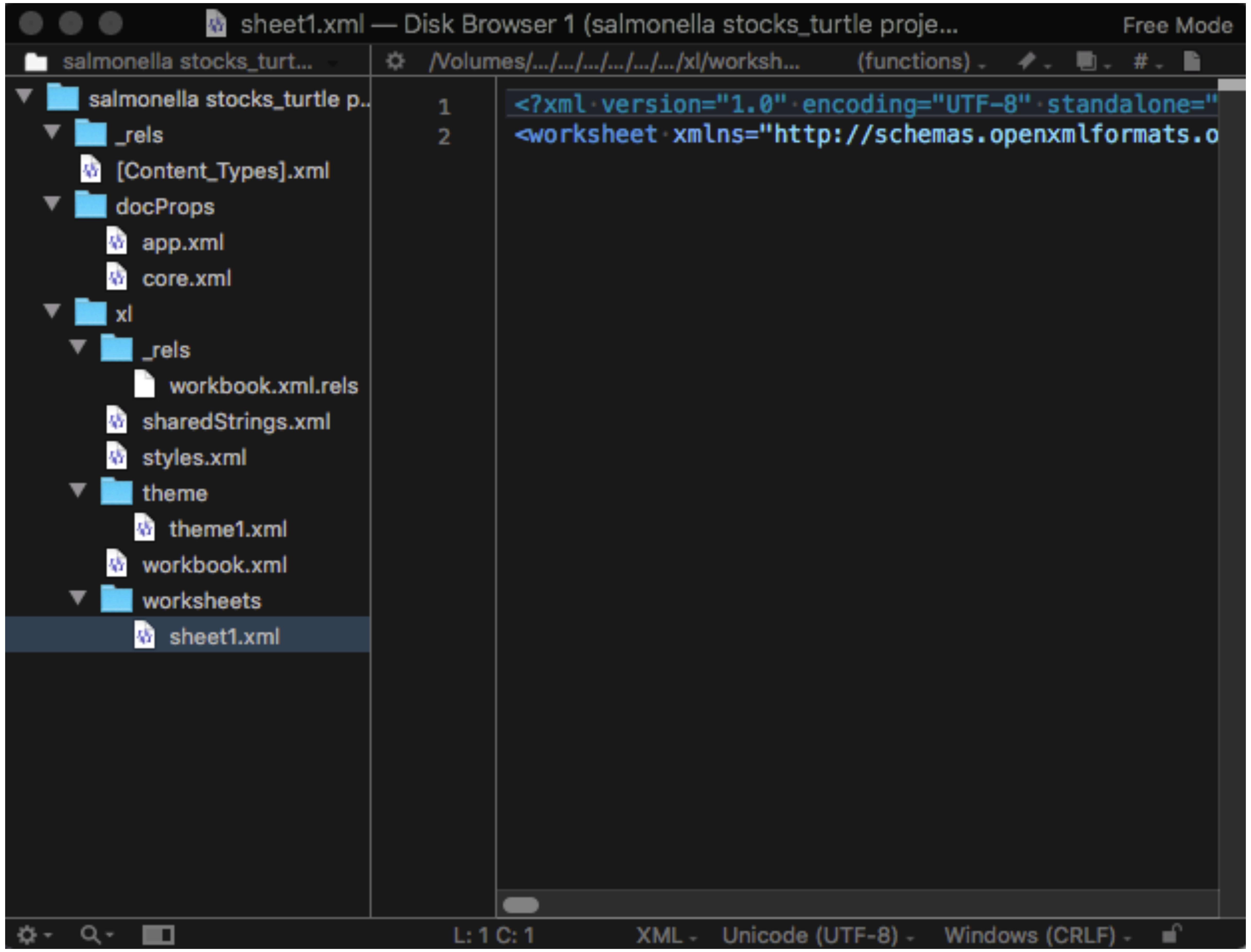
phylip

rich text (.rtf)

python script (.py)

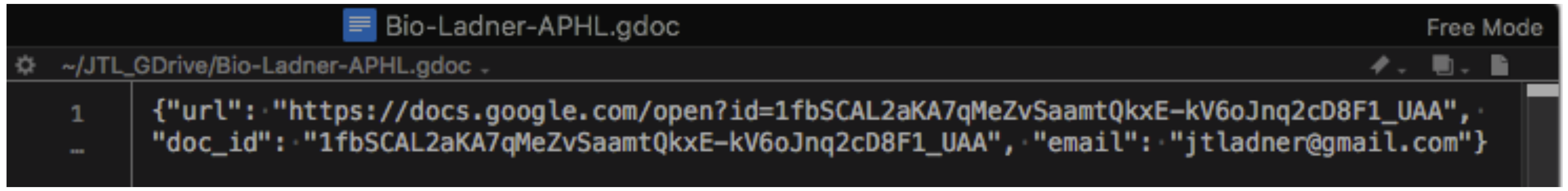
Viewing non-plain text in text editor

.xlsx/.docx



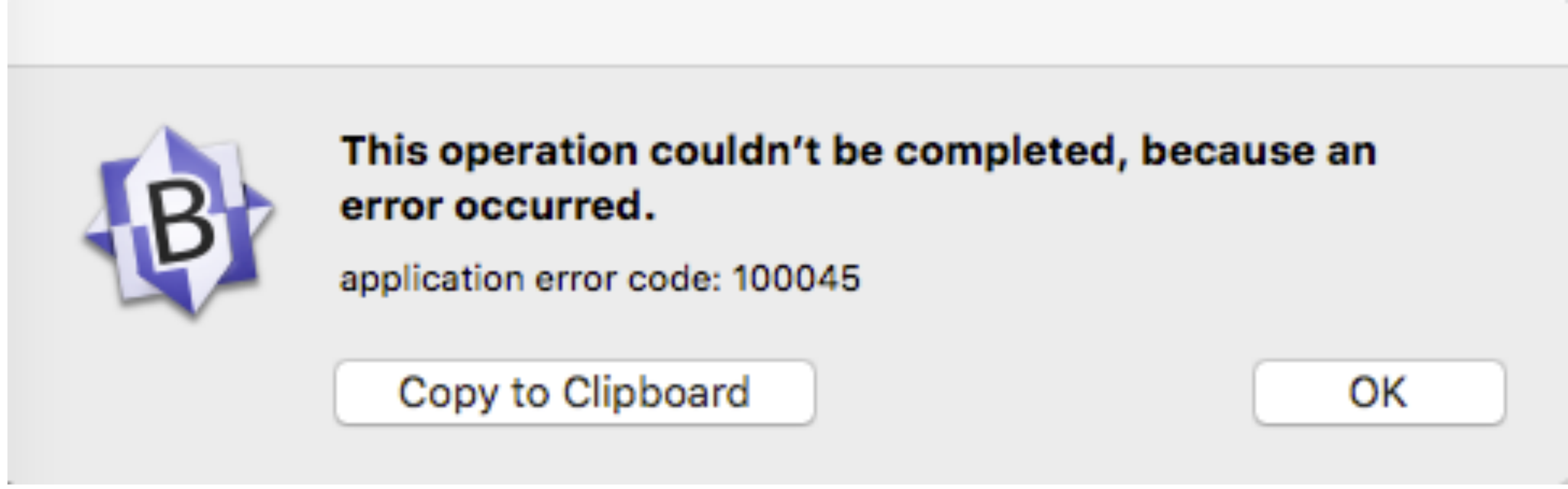
```
<?xml version="1.0" encoding="UTF-8" standalone="
<worksheet xmlns="http://schemas.openxmlformats.o
```

Google Doc



```
{"url": "https://docs.google.com/open?id=1fbSCAL2aKA7qMeZvSaamtQkxE-kV6oJnq2cD8F1_UAA", 
"doc_id": "1fbSCAL2aKA7qMeZvSaamtQkxE-kV6oJnq2cD8F1_UAA", "email": "jtladner@gmail.com"}
```

Google Sheet



This operation couldn't be completed, because an error occurred.

application error code: 100045

Regular expressions

(a.k.a. regex, regexp)

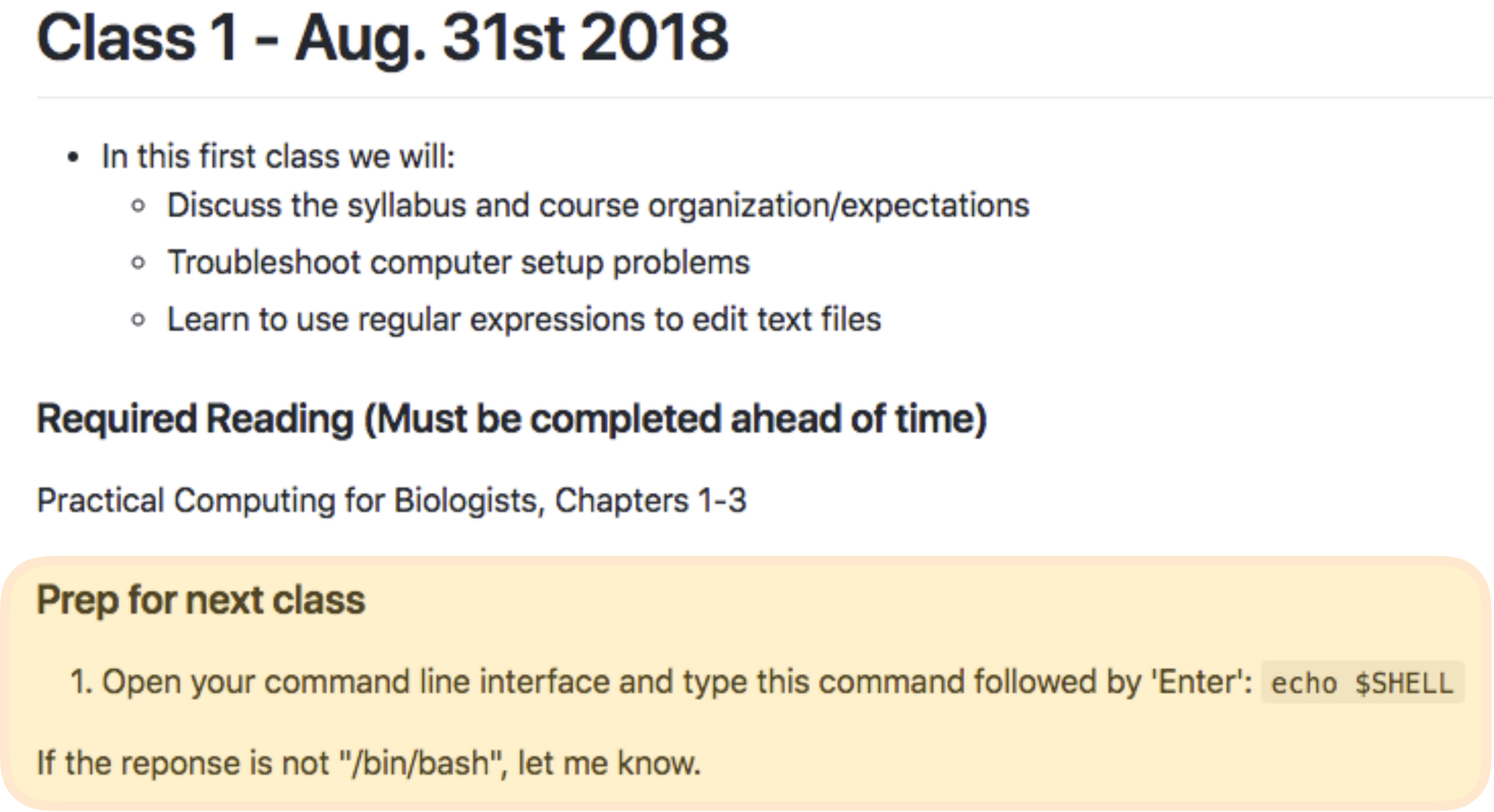
- Powerful search and replace toolkit
- Understood by many text editors, programming languages and even search engines
- Power comes from wildcard operators

Questions about the reading?

Tips

- Try PCfB methodology
    - copy target text into search dialog
    - replace text with wildcards, piece by piece
- Be as specific as possible
- Build in redundancies

“Prep for next class”



Class 1 - Aug. 31st 2018

- In this first class we will:
  - Discuss the syllabus and course organization/expectations
  - Troubleshoot computer setup problems
  - Learn to use regular expressions to edit text files

Required Reading (Must be completed ahead of time)

Practical Computing for Biologists, Chapters 1-3

Prep for next class

1. Open your command line interface and type this command followed by 'Enter': `echo $SHELL`

If the reponse is not "/bin/bash", let me know.

Exercises

(https://github.com/jtladner/Courses/tree/master/PracticalComputing/Fall_2018/Class1_Intro_RegExp)

RegexOne

Learn Regular Expressions with simple, interactive exercises.

Exercise 1: Matching Characters

Task	Text	
Match	abcdefg	✓
Match	abcde	✓
Match	abc	✓

```
\w+
```

Continue ›

Solve the above task to continue on to the next problem, or read the Solution.

Exercise 1: Matching Characters

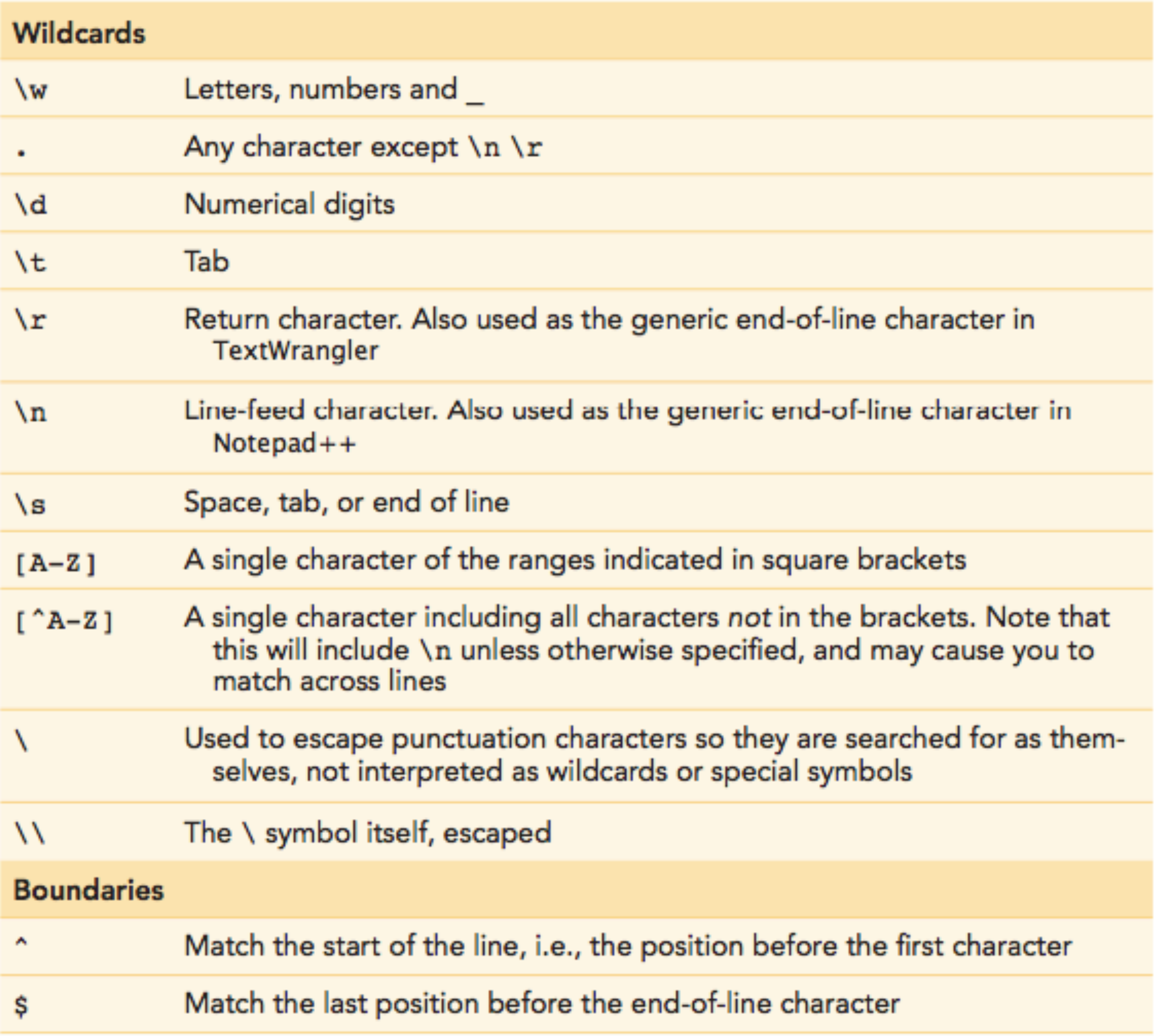
Task	Text	
Match	abcdefg	✓
Match	abcde	✓
Match	abc	✗

```
[e-g]+
```

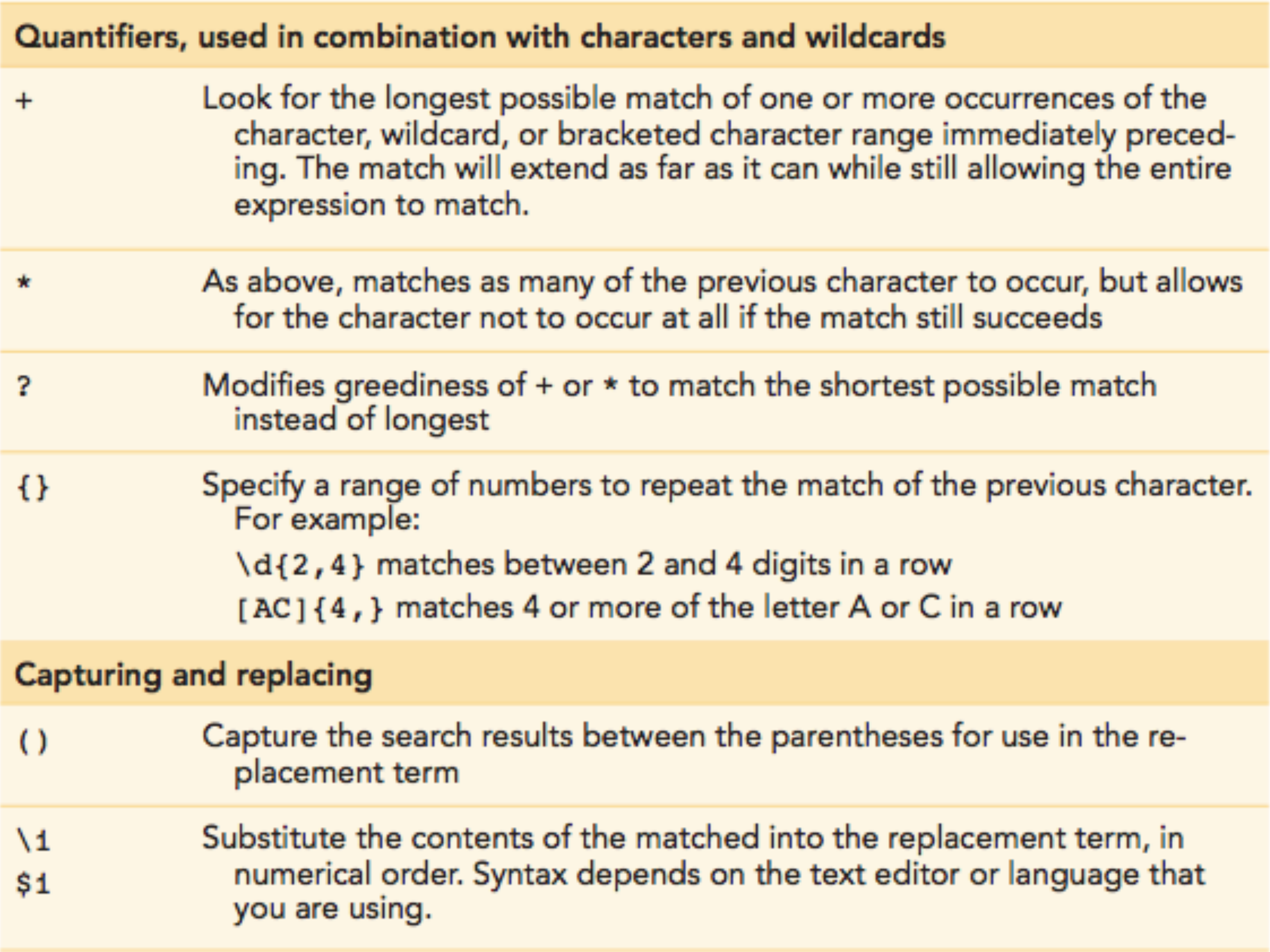
Continue ›

Solve the above task to continue on to the next problem, or read the Solution.

Regexp reference tables



Wildcards	
`\w`	Letters, numbers and _
`.`	Any character except `\n` `\r`
`\d`	Numerical digits
`\t`	Tab
`\r`	Return character. Also used as the generic end-of-line character in TextWrangler
`\n`	Line-feed character. Also used as the generic end-of-line character in Notepad++
`\s`	Space, tab, or end of line
`[A-Z]`	A single character of the ranges indicated in square brackets
`[^A-Z]`	A single character including all characters *not* in the brackets. Note that this will include `\n` unless otherwise specified, and may cause you to match across lines
`\`	Used to escape punctuation characters so they are searched for as themselves, not interpreted as wildcards or special symbols
`\\`	The \ symbol itself, escaped
Boundaries	
`^`	Match the start of the line, i.e., the position before the first character
`$`	Match the last position before the end-of-line character



Quantifiers, used in combination with characters and wildcards	
`+`	Look for the longest possible match of one or more occurrences of the character, wildcard, or bracketed character range immediately preceding. The match will extend as far as it can while still allowing the entire expression to match.
`*`	As above, matches as many of the previous character to occur, but allows for the character not to occur at all if the match still succeeds
`?`	Modifies greediness of + or * to match the shortest possible match instead of longest
`{}`	Specify a range of numbers to repeat the match of the previous character. For example: `\d{2,4}` matches between 2 and 4 digits in a row `[AC]{4,}` matches 4 or more of the letter A or C in a row
Capturing and replacing	
`()`	Capture the search results between the parentheses for use in the replacement term
`\1` `$1`	Substitute the contents of the matched into the replacement term, in numerical order. Syntax depends on the text editor or language that you are using.

http://practicalcomputing.org/files/PCfB_Appendices.pdf