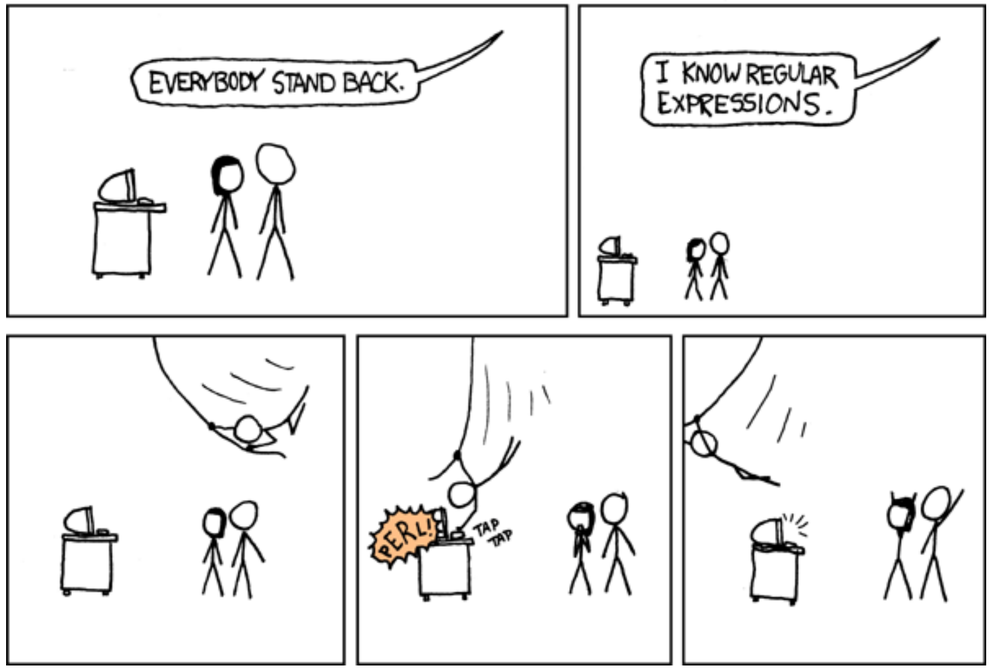
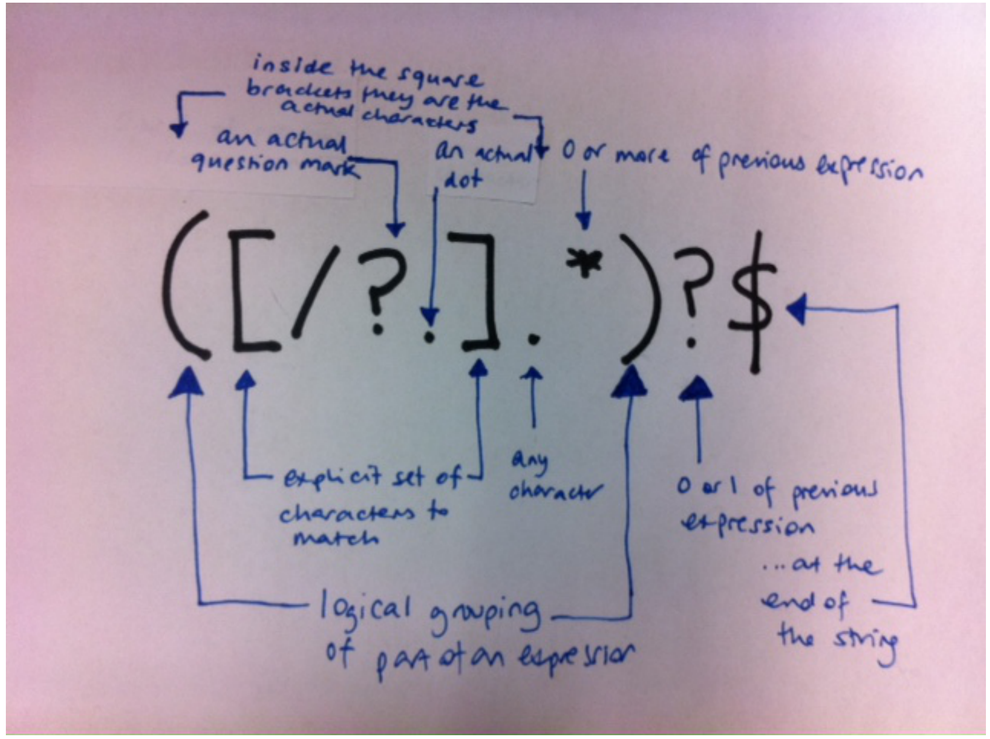
Intro & Regular Expressions

Spring 2022
PCfB Class 1
January 14, 2022





Outline

- Introductions
- Course organization
- Plain text files
- Regular expressions

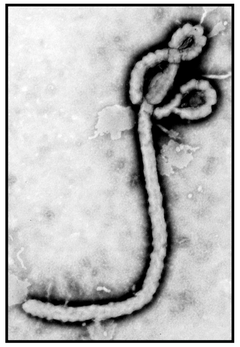
PhD - Evolutionary genetics

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**
4. **(Optional) Pronouns**

Course organization

What this course is:

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

What it isn't:

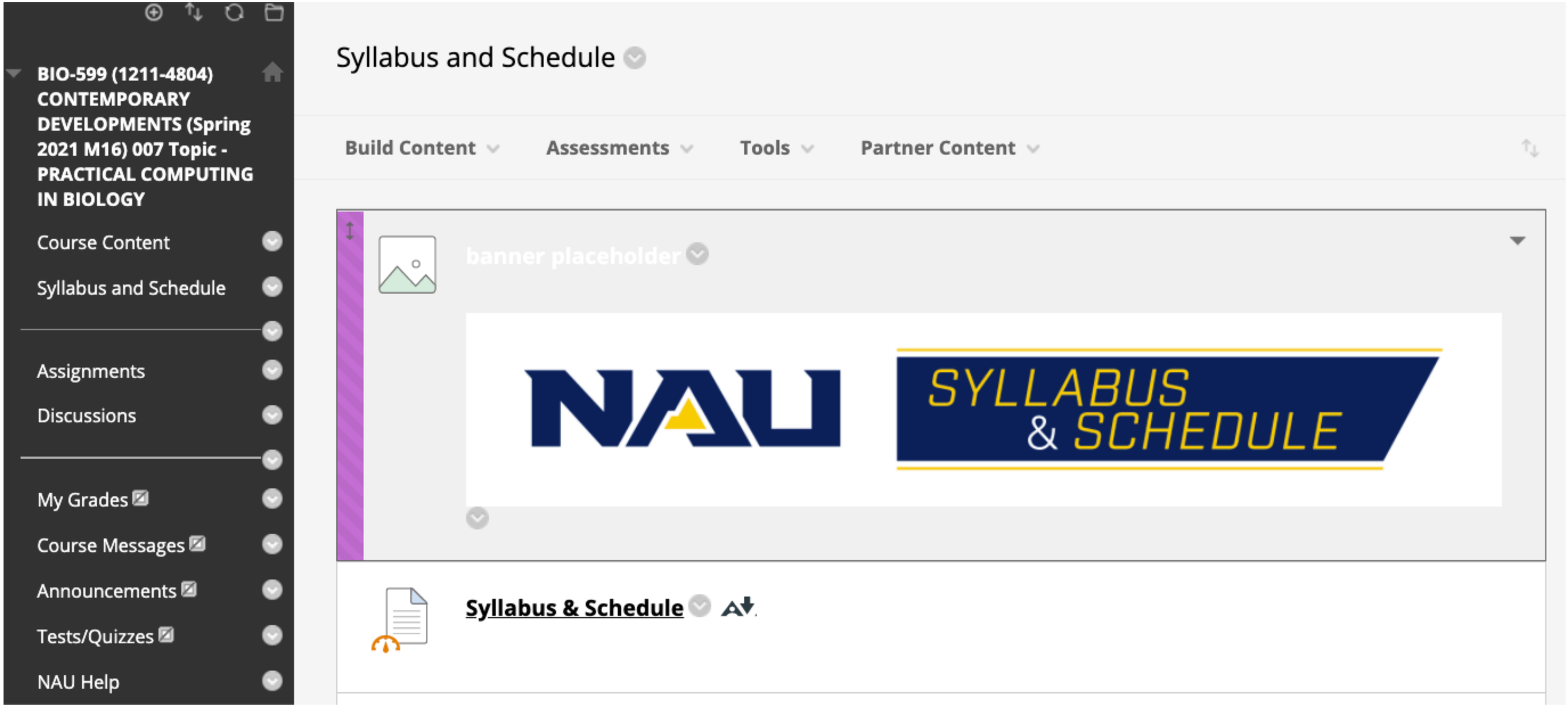
- A bioinformatics class

Most course materials on

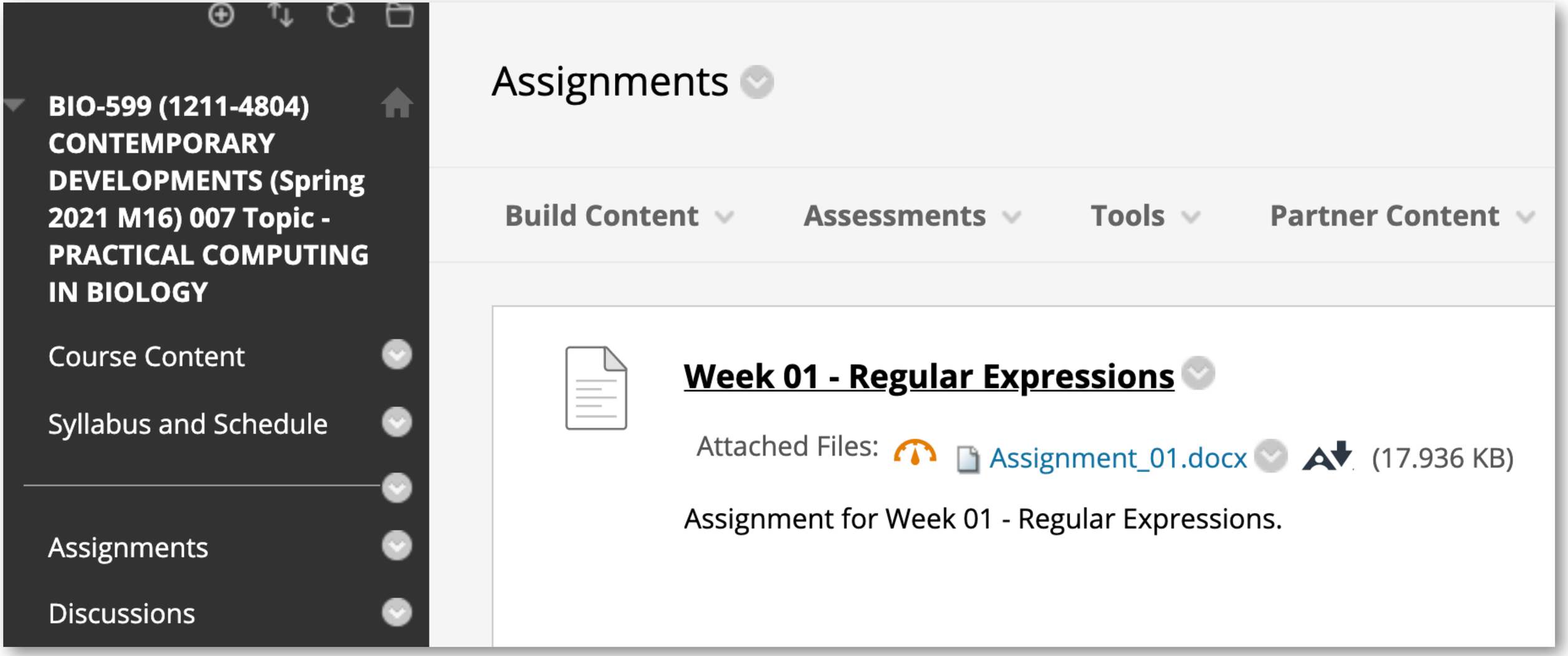
https://github.com/jtladner/PracticalComputing_Spring2022

“Pulling” GitHub updates

Syllabus on Bb Learn



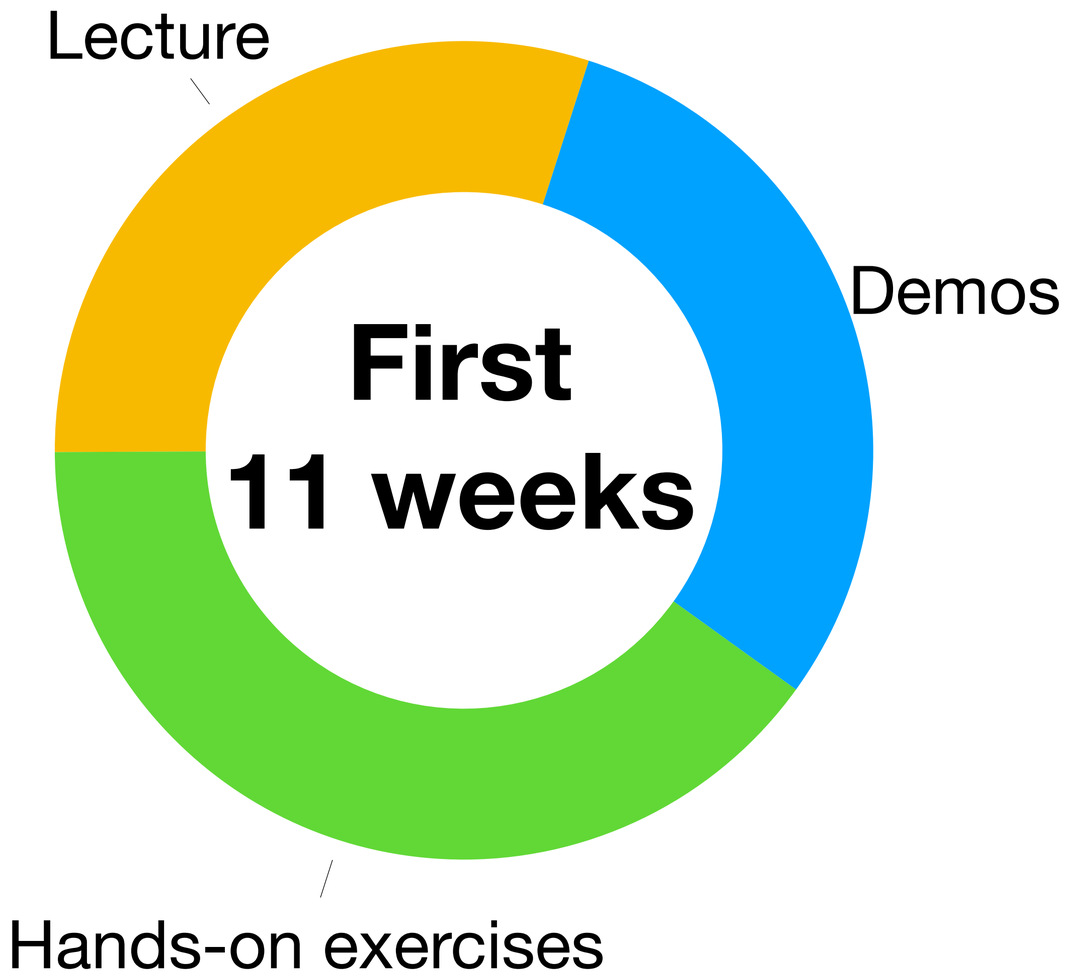
Assignments submitted via Bb Learn



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



Individual projects
(Last 5 weeks)

- Individual coding projects
- Topic of your choice
- 3 work weeks
- 2 weeks for presentations

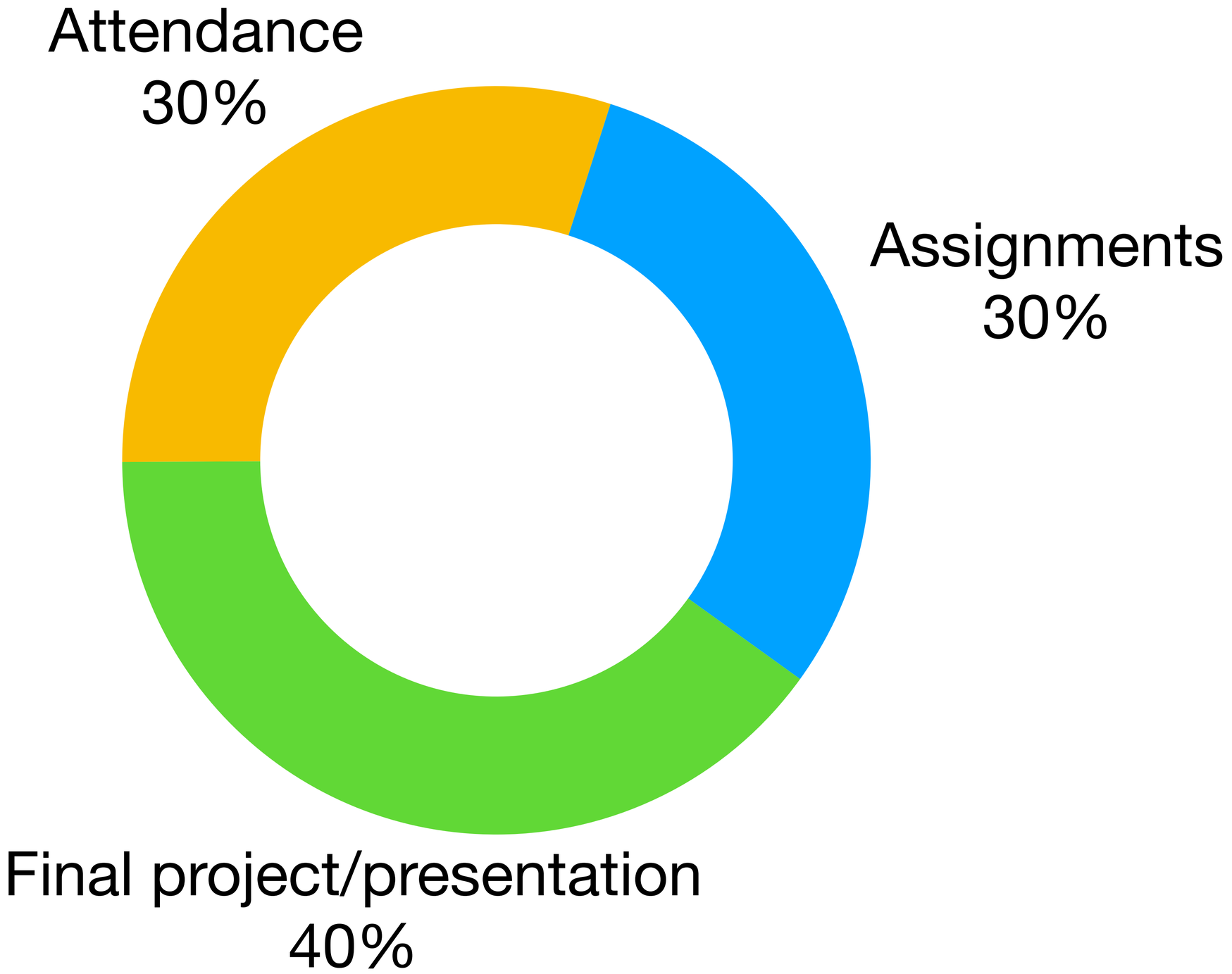
Assignments

- One assignment per week (weeks 1-11)
- Focus on hands-on time in class (may need to complete outside of class)
- Always due by 11:59 pm on Thursday
- Directions on GitHub, Answer sheet on Bb Learn
- Partial credit for revisions

Breakout groups for hands-on exercises

- Main room: questions/troubleshooting w/ professor
- Quiet workspace: for independent work
- Discussion spaces: for discussion among students

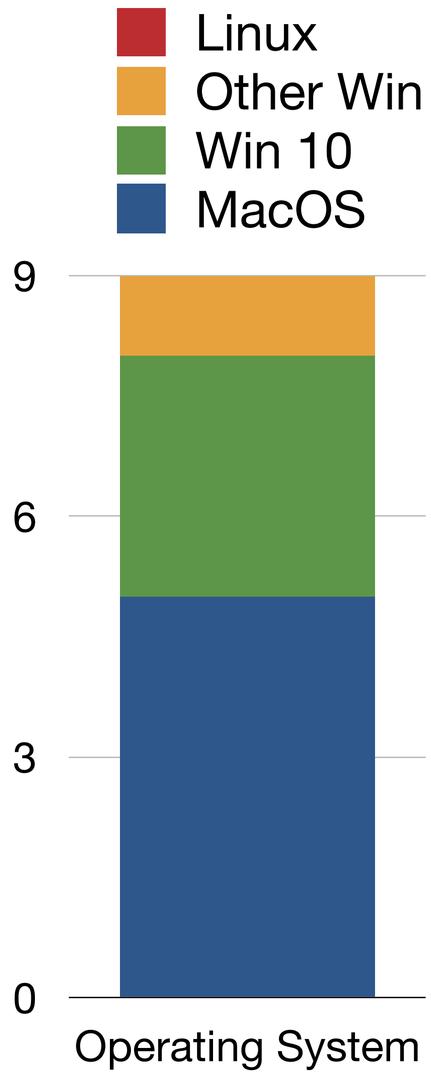
Grading

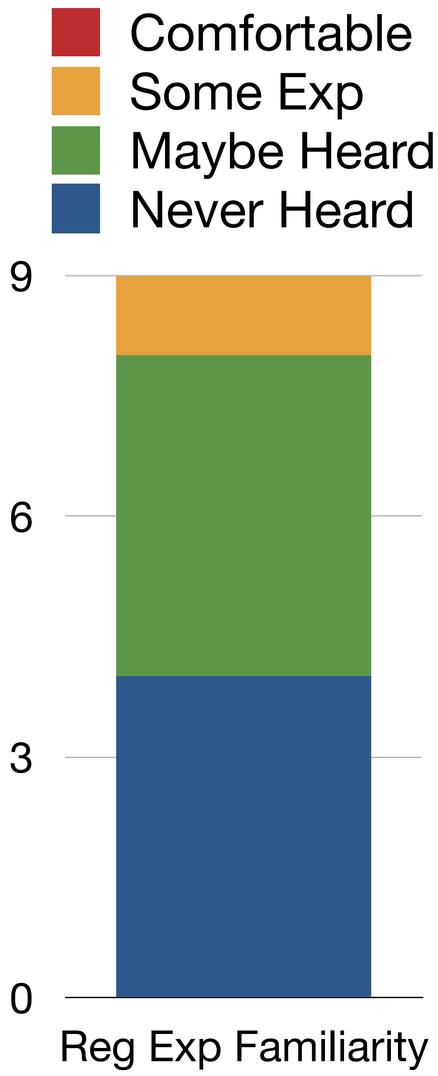

Attendance
30%
Assignments
30%
Final project/presentation
40%

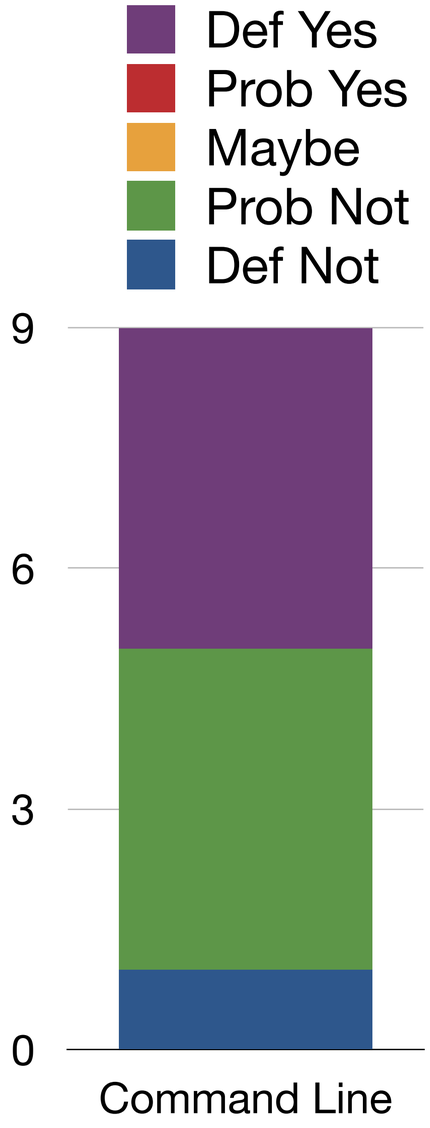
Final project - deadlines

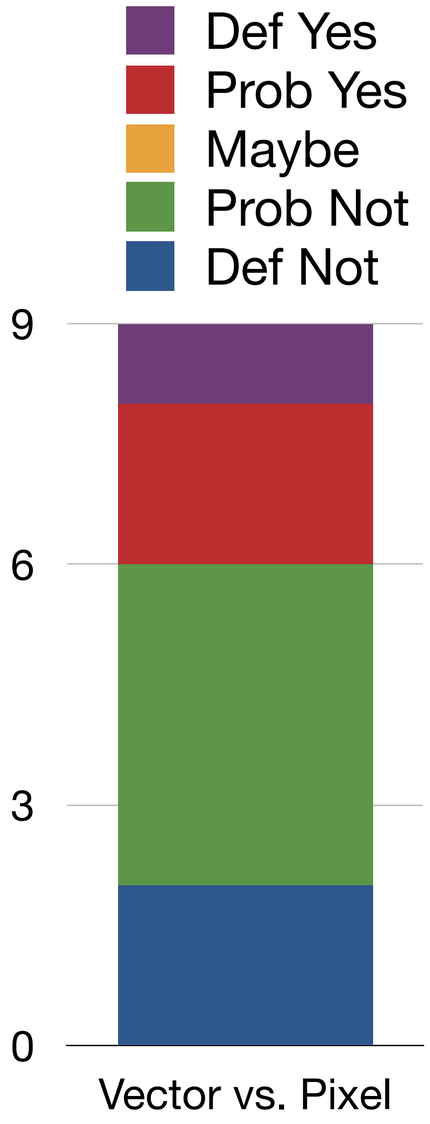
Week	Date	Topic	Reading
1	1/14	Intro, Setup & Regular Expressions	PCfB: Ch. 1-3
2	1/21	The Shell - Part 1	PCfB: Ch. 4-5
3	1/28	The Shell - Part 2	PCfB: Ch. 6, 21
4	2/4	Python Programming - Part 1	PCfB: Ch. 7-8 Jupyter Tutorial
5	2/11	Python Programming - Part 2	PCfB: Ch. 9
6	2/18	Python Programming - Part 3	PCfB: Ch. 10-11
7	2/25	Python Programming - Part 4	PCfB: Ch. 12
8	3/4	Debugging, Combining Methods	PCfB: Ch. 13-14
9	3/11	Graphical concepts: vectors vs. pixels	PCfB: Ch. 17-19
10	3/18	Making Figures in Python - Part 1	Matplotlib overview
11	3/25	Making Figures in Python - Part 2 (*Project proposal due)	
12	4/1	Work/Troubleshoot Day #1	
13	4/8	Work/Troubleshoot Day #2	
14	4/15	Work/Troubleshoot Day #3	
15	4/22	Project Presentations - Part 1	
16	4/29	Project Presentations - Part 2	
Finals	5/2	*Final project due	

Pre-class survey









Computer setup

- Text Editor
- GitHub Repository
- Command line terminal

https://github.com/jtladner/PracticalComputing_Spring2022/tree/master/Getting%20Started

Plain text files

Plain text file

- Pure sequence of character codes
- No formatting (e.g., text size, color, font, spacing)
- Human and machine readable
- Standardized

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)

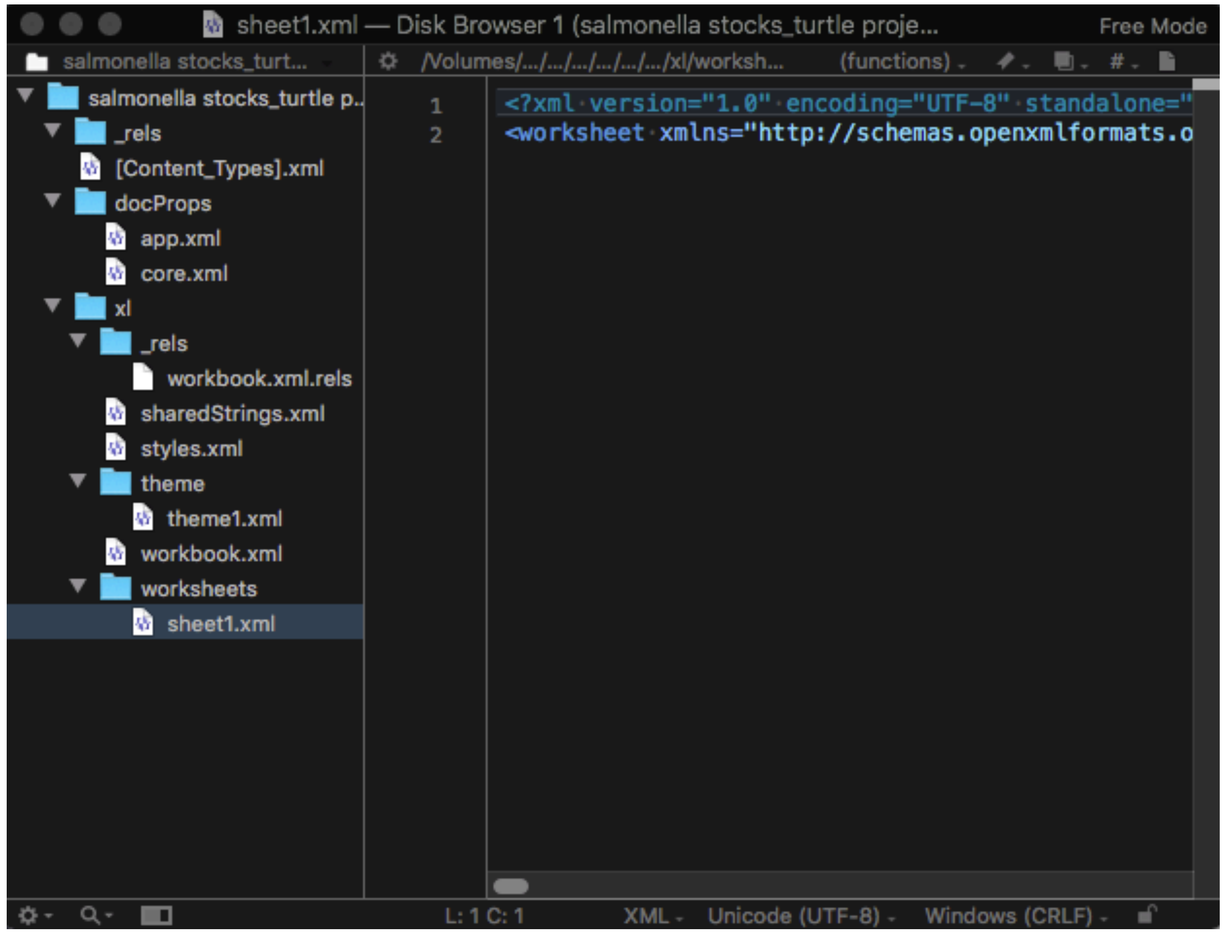
rich text (.rtf)

python script (.py)

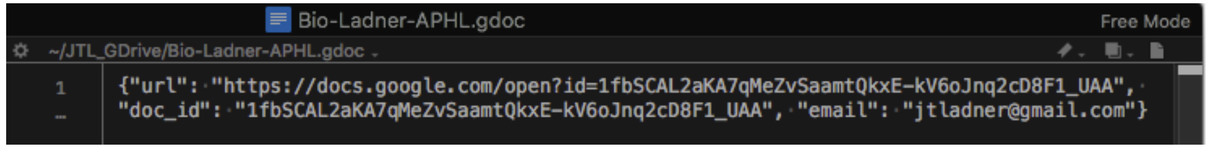
tab-separated (.tsv)

Viewing non-plain text in text editor

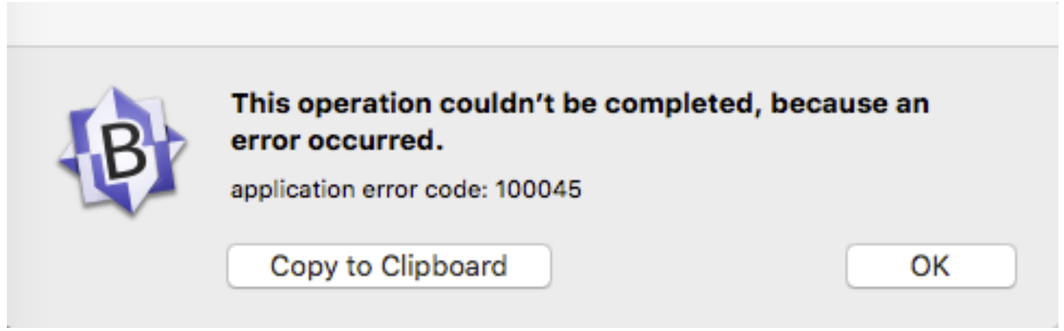
.xlsx/.docx



Google Doc



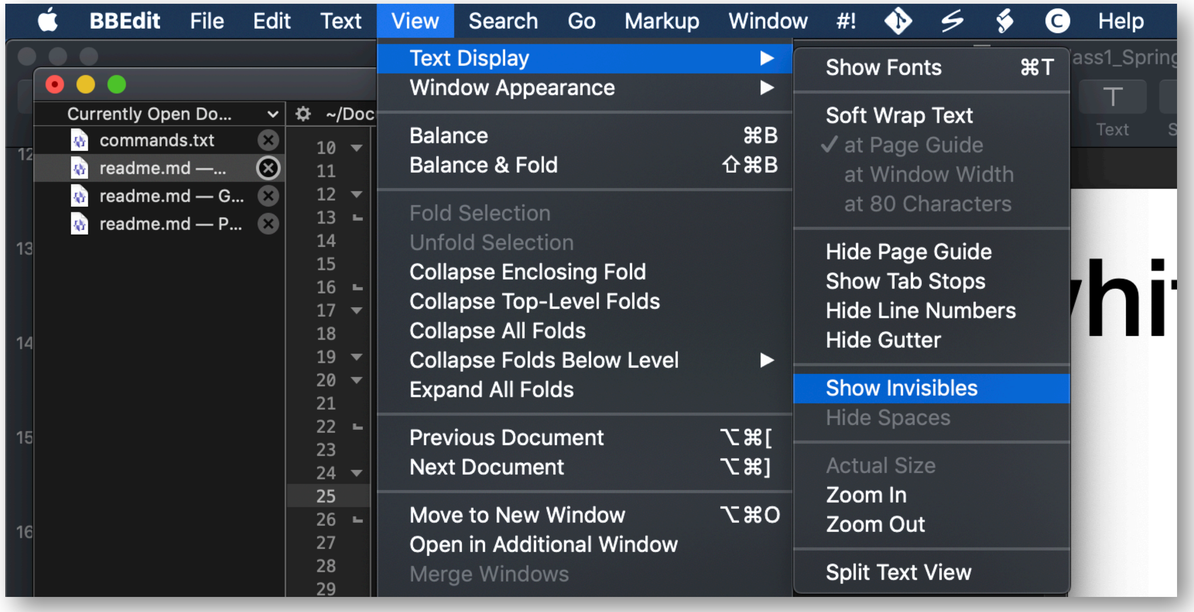
Google Sheet

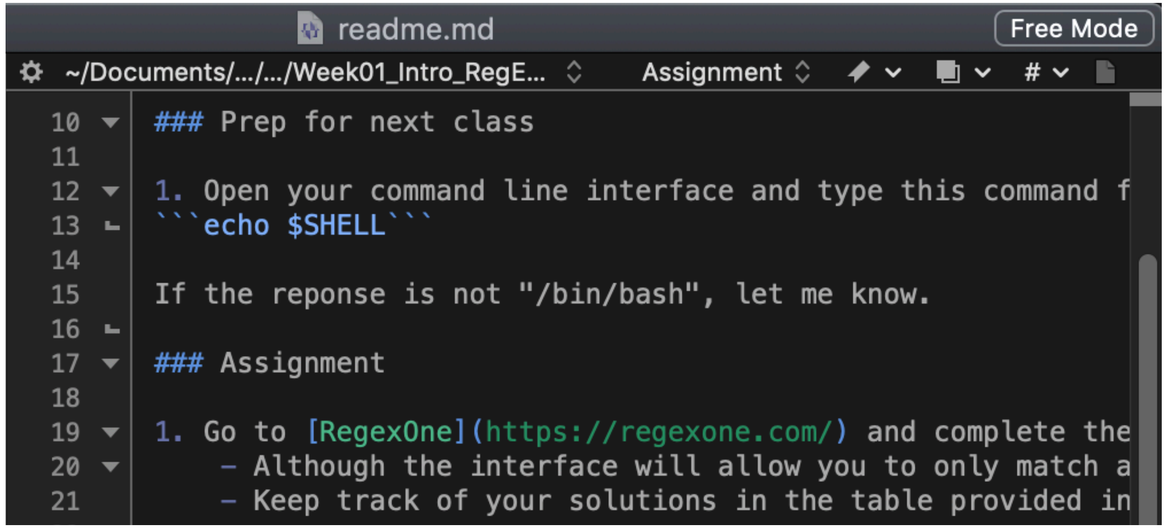


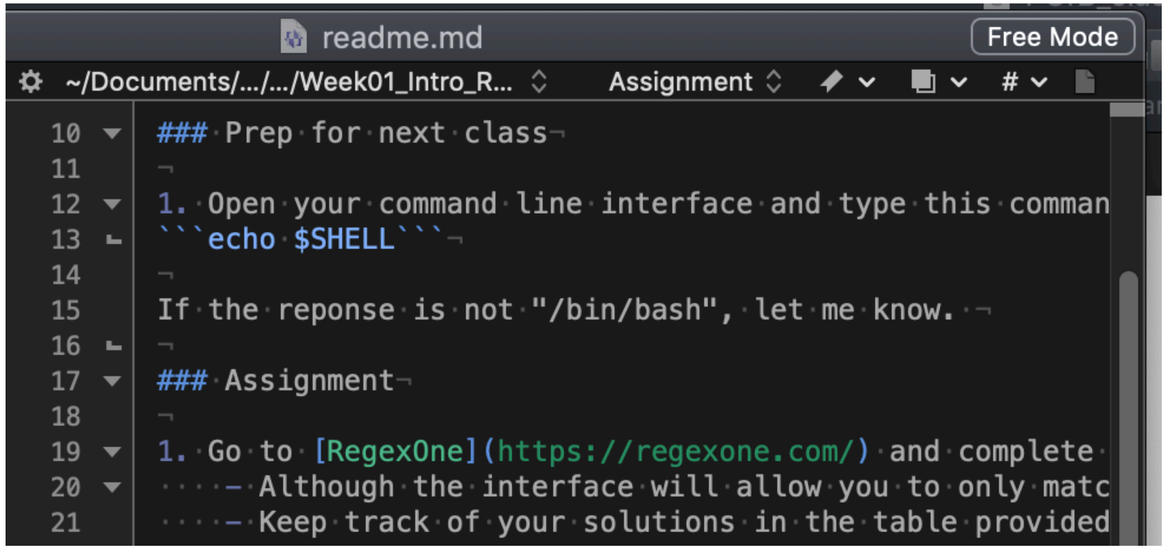
Whitespace

- Space
- Tab
- End of line

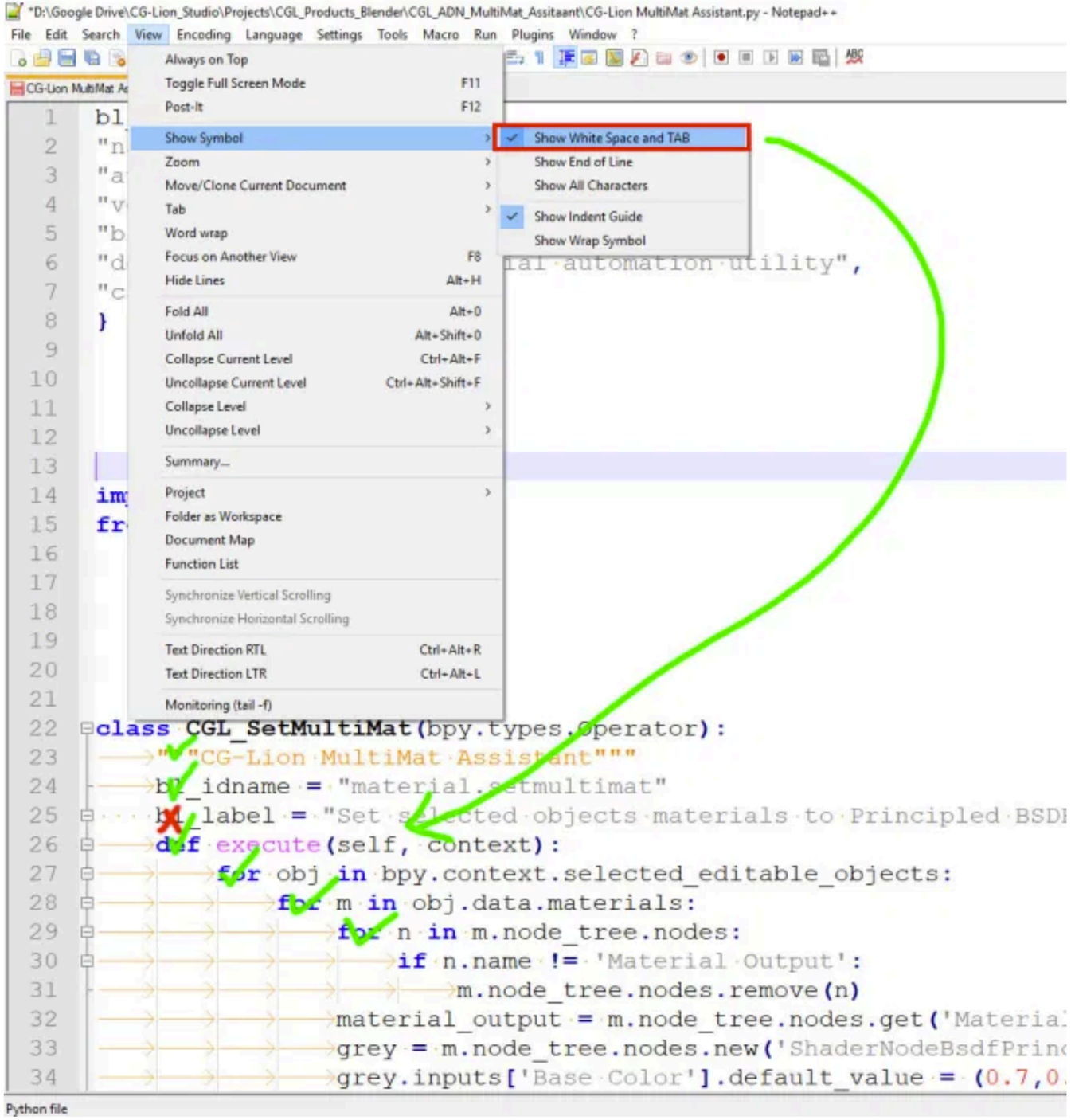
Visualizing white space (BBEdit)



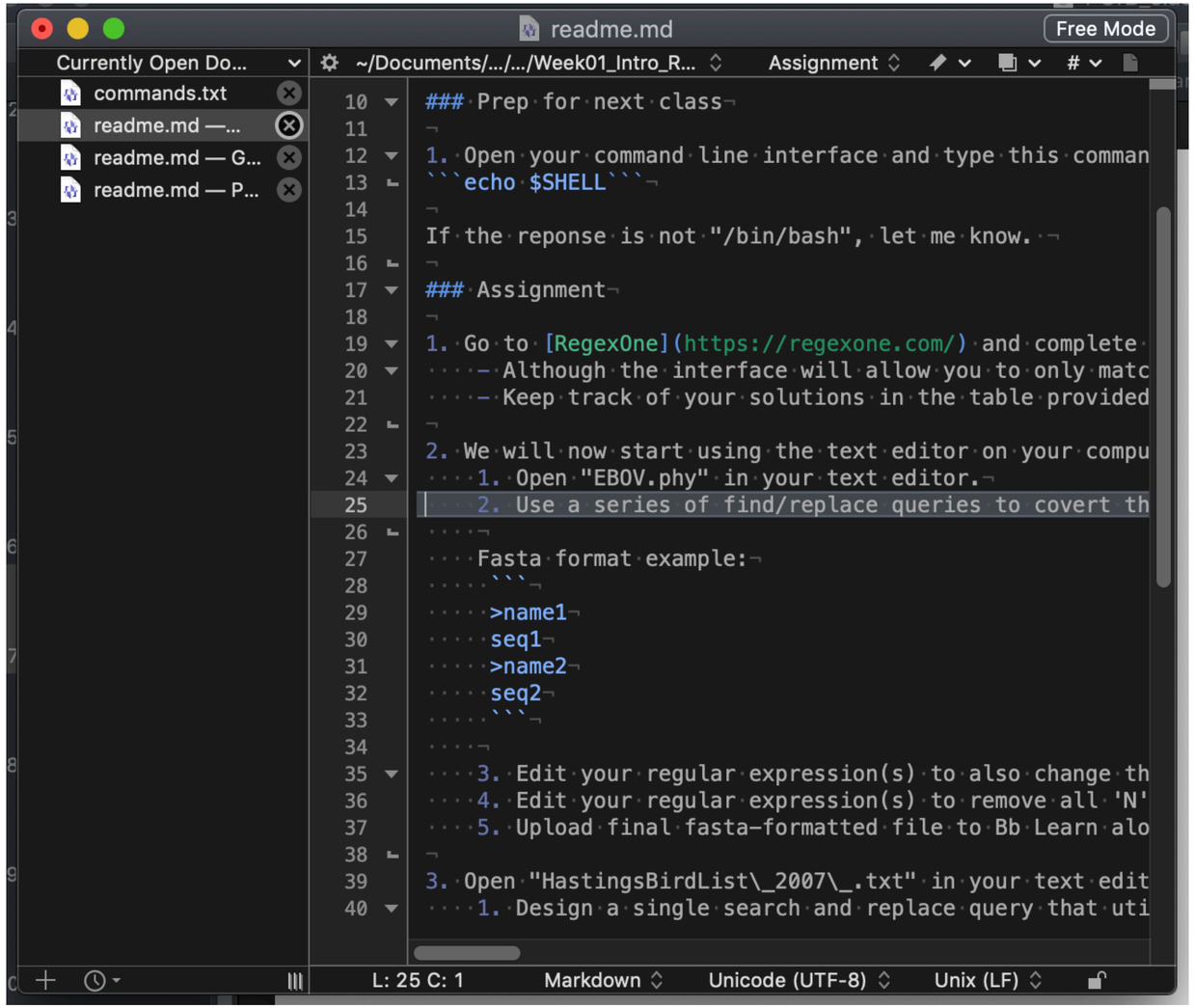




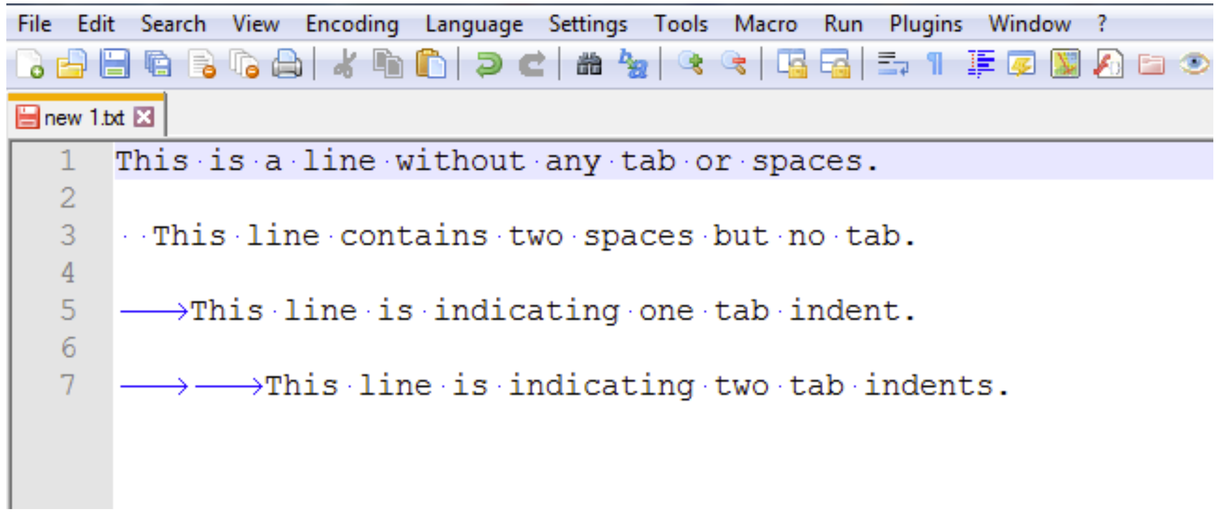
Visualizing white space (Notepad++)

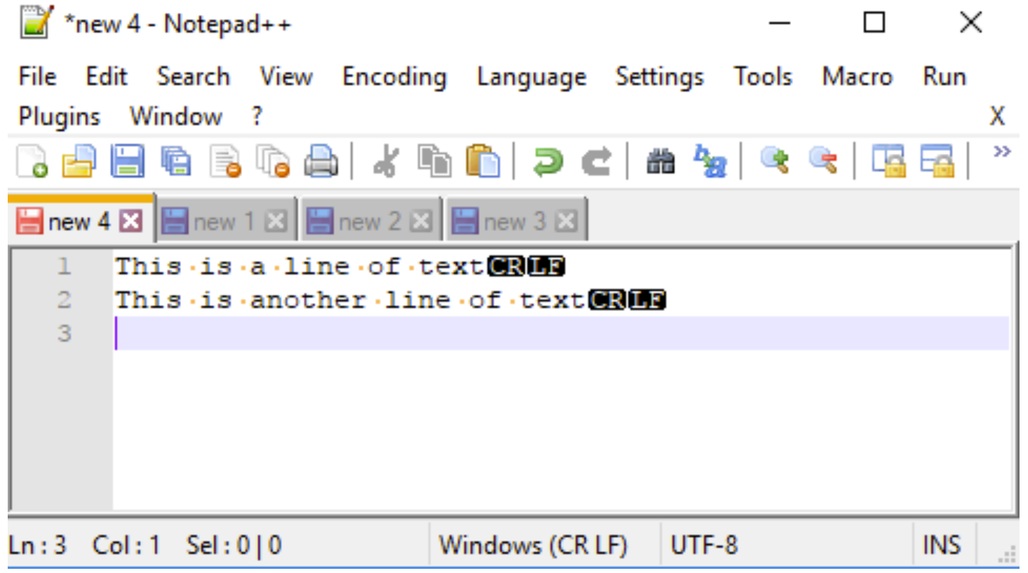


BBEdit



Notepad++





End of line characters differ by OS

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

Regular expressions

Regular expressions
(a.k.a. regex, regexp)

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

\d

\w

\s

.

```
\w+

\w*

\w?
```

[ABC]

[^ABC]

[A-C]

(ABC)

(AB)C

((AB)C)

^

$

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

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

Questions about the reading?

RegexOne

Learn Regular Expressions with simple, interactive exercises.

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.

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.

“Prep for next class”

Class 1 - Jan. 14th 2022

- 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

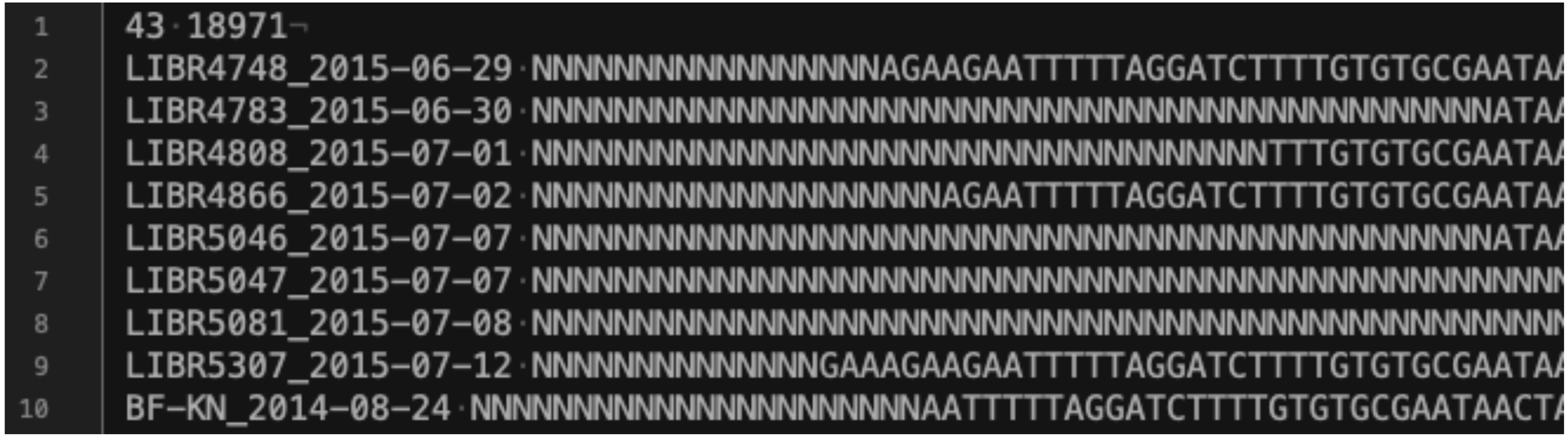
Required Reading (Must be completed ahead of time)

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" or "/bin/zsh", let me know.



```
43 18971
LIBR4748_2015-06-29 NNNNNNNNNNNNNNNNAGAAGAATTTTTAGGATCTTTTGTGTGCGAATAA
LIBR4783_2015-06-30 NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNATAA
LIBR4808_2015-07-01 NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNTTTGTGTGCGAATAA
LIBR4866_2015-07-02 NNNNNNNNNNNNNNNNNNNAGAATTTTTAGGATCTTTTGTGTGCGAATAA
LIBR5046_2015-07-07 NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNATAA
LIBR5047_2015-07-07 NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNM
LIBR5081_2015-07-08 NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNM
LIBR5307_2015-07-12 NNNNNNNNNNNNNNGAAAGAAGAATTTTTAGGATCTTTTGTGTGCGAATAA
BF-KN_2014-08-24 NNNNNNNNNNNNNNNNNNNNNAATTTTTAGGATCTTTTGTGTGCGAATAACTA
```

```
>LIBR4748
AGAAGAATTTTTAGGATCTTTTGTGTGCGAATAACTATGAGGAAGATTAATAATTTTCCTCTCATTGAAATTTATATCGGAATT
>LIBR4783
ATAACTATGAGGAAGATTAATAATTTTCCTCTCATTGAAATTTATATCGGAATTTAAATTGAAATTGTTACTGTAATCATACCT
>LIBR4808
TTTGTGTGCGAATAACTATGAGGAAGATTAATAATTTTCCTCTCATTGAAATTTATATCGGAATTTAAATTGAAATTGTTACTG
>LIBR4866
AGAATTTTTAGGATCTTTTGTGTGCGAATAACTATGAGGAAGATTAATAATTTTCCTCTCATTGAAATTTATATCGGAATTTAA
>LIBR5046
ATAACTATGAGGAAGATTAATAATTTTCCTCTCATTGAAATTTATATCGGAATTTAAATTGAAATTGTTACTGTAATCATACCT
>LIBR5047
GGAAGATTAATAATTTTCCTCTCATTGAAATTTATATCGGAATTTAAATTGAAATTGTTACTGTAATCATACCTGGTTTGTTTC
>LIBR5081
ACCTGGTTTGTTTCAGAGCCATATCACCAAGATAGAGAACAACCTAGGTCTCCGGAGGGGGCAAGGGCATCAGTGTGCTCAGTT
>LIBR5307
GAAAGAAGAATTTTTAGGATCTTTTGTGTGCGAATAACTATGAGGAAGATTAATAATTTTCCTCTCATTGAAATTTATATCGGA
>BF-KN
AATTTTTAGGATCTTTTGTGTGCGAATAACTATGAGGAAGATTAATAATTTTCCTCTCATTGAAATTTATATCGGAATTTAAAT
```

Text editor regex demos

Start

(email)

Sample ID sample collection date Gender Age Location
N27 22.04.2020 100010117153 F 52 Trondelag
N28 22.04.2020 100010117157 M 51 Trondelag
N29 22.04.2020 100010117161 M 31 Trondelag
N30 20.04.2020 121252.43310 M 67 Trondelag
N31 21.04.2020 121097.39802 F 22 Trondelag
N32 14.04.2020 100010126959 F 57 Trondelag

						Sex(F/M)	Age(years)
N33	Oslo	20.03.2020	17.04.2020	COVID-19 convalescent	J000920011268	F	30
N34	Oslo	22.03.2020	17.04.2020	COVID-19 convalescent	J000920011287	F	47
N35	Oslo	09.03.2020	17.04.2020	COVID-19 convalescent	J000920011293	M	35
N36	Oslo	13.03.2020	17.04.2020	COVID-19 convalescent	J000920011322	F	53
N37	Oslo	09.03.2020	17.04.2020	COVID-19 convalescent	J000920011324	M	38
N38	Oslo	25.03.2020	17.04.2020	COVID-19 convalescent	J000920011341	F	50
N39	Oslo	25.03.2020	17.04.2020	COVID-19 convalescent	J000920011353	F	78
N40	Oslo	23.03.2020	17.04.2020	COVID-19 convalescent	J000920011348	F	58
N41	Oslo	11.03.2020	16.04.2020	COVID-19 convalescent	J000920011072	M	52
N42	Oslo	27.03.2020	16.04.2020	COVID-19 convalescent	J000920011091	F	70
N43	Oslo	11.03.2020	16.04.2020	COVID-19 convalescent	J000920011095	F	36

End

(tsv)

SampleID	SampleCollectionDate	UnkID	Gender	Age	Location
N27	2020-04-22	100010117153	F	52	Trondelag
N28	2020-04-22	100010117157	M	51	Trondelag
N29	2020-04-22	100010117161	M	31	Trondelag
N30	2020-04-20	121252.43310	M	67	Trondelag
N31	2020-04-21	121097.39802	F	22	Trondelag
N32	2020-04-14	100010126959	F	57	Trondelag

N33	2020-03-20	J000920011268	F	30	Oslo
N34	2020-03-22	J000920011287	F	47	Oslo
N35	2020-03-09	J000920011293	M	35	Oslo
N36	2020-03-13	J000920011322	F	53	Oslo
N37	2020-03-09	J000920011324	M	38	Oslo
N38	2020-03-25	J000920011341	F	50	Oslo
N39	2020-03-25	J000920011353	F	78	Oslo
N40	2020-03-23	J000920011348	F	58	Oslo
N41	2020-03-11	J000920011072	M	52	Oslo
N42	2020-03-27	J000920011091	F	70	Oslo
N43	2020-03-11	J000920011095	F		Oslo