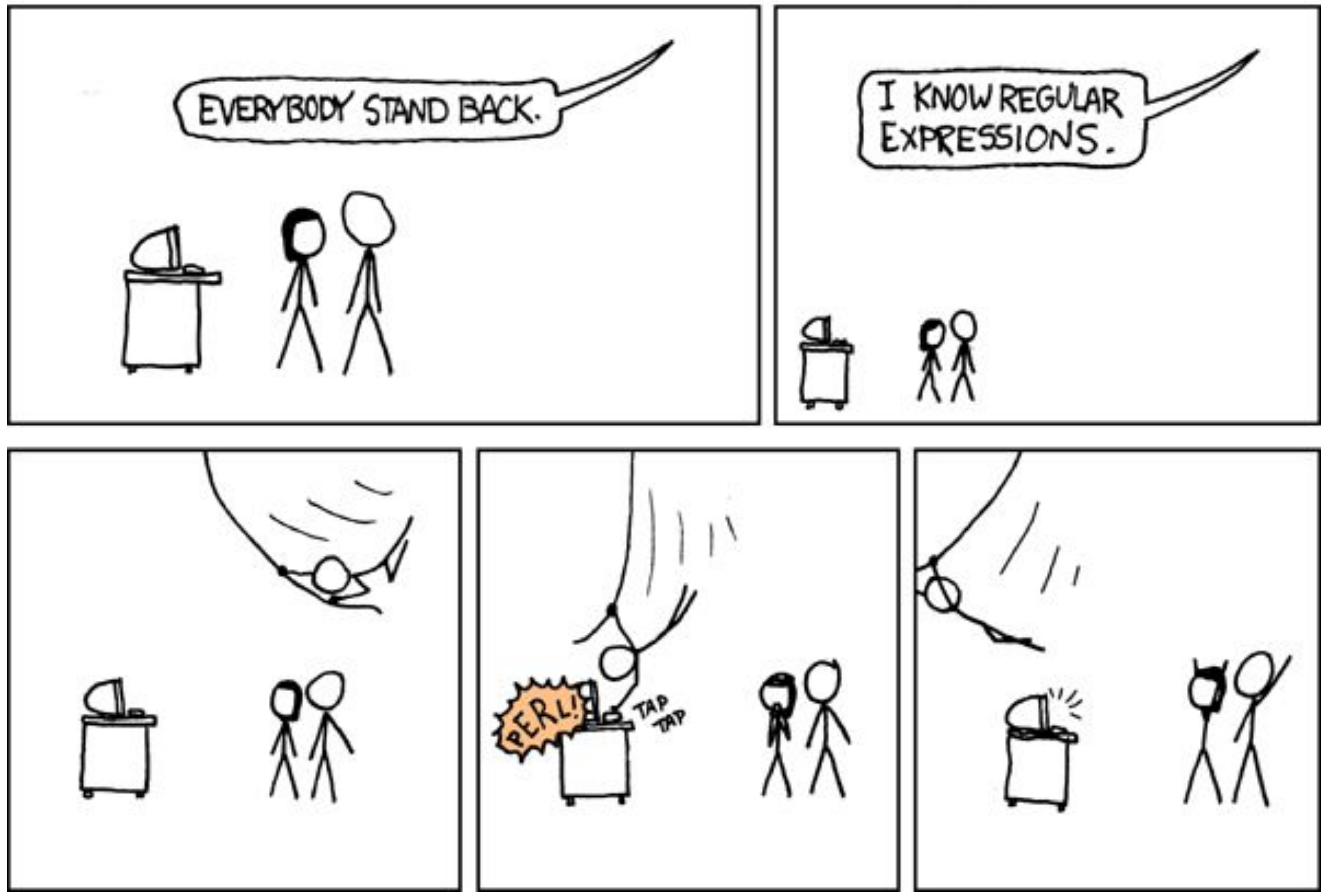
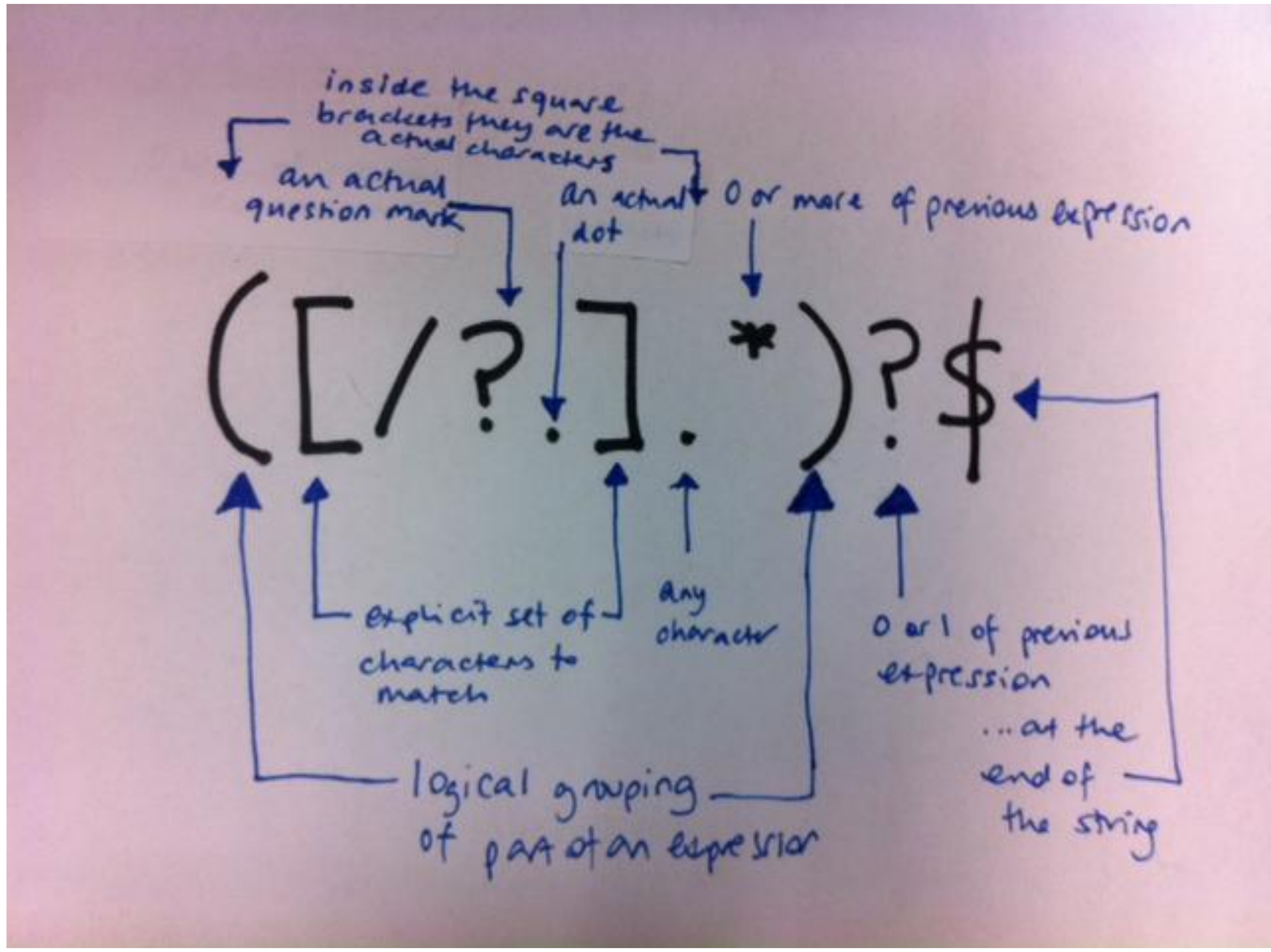
Intro & Regular Expressions

Spring 2023, Week 1
January 20, 2022




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

Outline

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

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

Syllabus on Bb Learn

BIO-599 (1211-4804) CONTEMPORARY DEVELOPMENTS (Spring 2021 M16) 007 Topic - PRACTICAL COMPUTING IN BIOLOGY
Course Content
Syllabus and Schedule
Assignments
Discussions
My Grades
Course Messages
Announcements
Tests/Quizzes
NAU Help
Syllabus and Schedule
Build Content
Assessments
Tools
Partner Content
banner placeholder
NAU
SYLLABUS & SCHEDULE
Syllabus & Schedule

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

New Content

(First 11 weeks)

- Lectures
- Demos
- In class work time (homework assignments)

Individual projects

(Last 4 weeks)

- Individual coding projects
- Topic of your choice
- 2 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
- Partial credit for revisions

Assignments submitted via Bb Learn

BIO-682 (1231-10755) QUANTITATIVE BIOLOGY (Spring 2023 M16) 001
Course Content
Syllabus and Schedule
Assignments
Demos - 2022 Archive
Lectures - 2022 Archive
Discussions
My Grades
Course Messages
Build Content
Assessments
Tools
Partner Content

Assignments

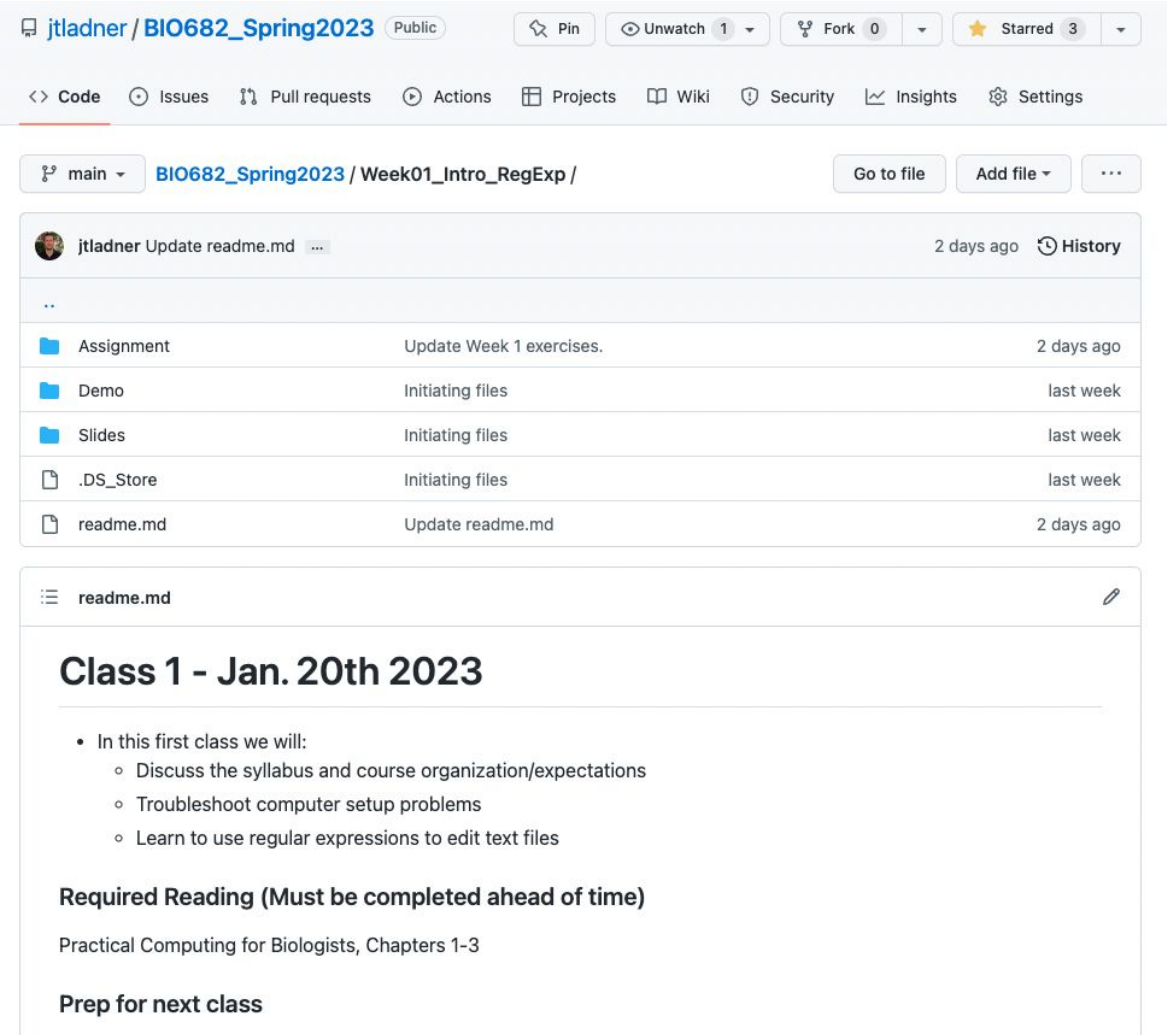
Week 01 - Regular Expressions

Attached Files: Assignment_01.docx (18.934 KB)
filenames.txt (390 B)
HastingsBirdList_2007.txt (26.383 KB)

1. Download the Assignment_01.docx and other assignment files (also available on GitHub).
2. Follow instrictions in GitHub repository.
3. Upload a) completed answer sheet, b) reformatted version of filenames.txt and c) reformatted version of HastingsBirdList_2007.txt.

GitHub

https://github.com/jtladner/BIO682_Spring2023



Assignment	Update Week 1 exercises.	2 days ago
Demo	Initiating files	last week
Slides	Initiating files	last week
.DS_Store	Initiating files	last week
readme.md	Update readme.md	2 days ago

readme.md

Class 1 - Jan. 20th 2023

- 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

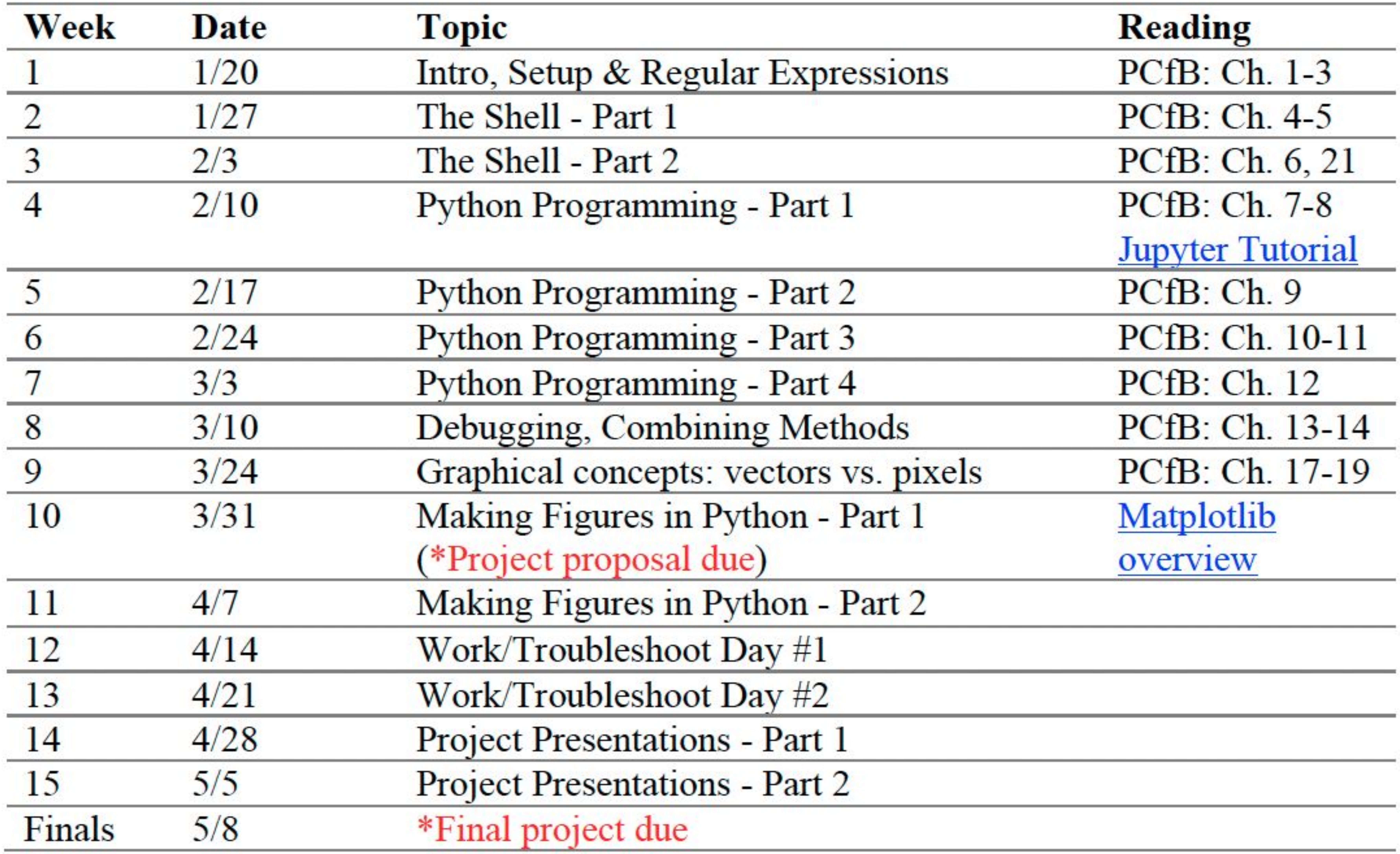
Prep for next class

“Pulling” GitHub updates

Grading

- Assignments (30%)
- Attendance/Participation (30%)
- Final Project/Presentation (40%)

Final project - deadlines



Week	Date	Topic	Reading
1	1/20	Intro, Setup & Regular Expressions	PCfB: Ch. 1-3
2	1/27	The Shell - Part 1	PCfB: Ch. 4-5
3	2/3	The Shell - Part 2	PCfB: Ch. 6, 21
4	2/10	Python Programming - Part 1	PCfB: Ch. 7-8 Jupyter Tutorial
5	2/17	Python Programming - Part 2	PCfB: Ch. 9
6	2/24	Python Programming - Part 3	PCfB: Ch. 10-11
7	3/3	Python Programming - Part 4	PCfB: Ch. 12
8	3/10	Debugging, Combining Methods	PCfB: Ch. 13-14
9	3/24	Graphical concepts: vectors vs. pixels	PCfB: Ch. 17-19
10	3/31	Making Figures in Python - Part 1 (*Project proposal due)	Matplotlib overview
11	4/7	Making Figures in Python - Part 2	
12	4/14	Work/Troubleshoot Day #1	
13	4/21	Work/Troubleshoot Day #2	
14	4/28	Project Presentations - Part 1	
15	5/5	Project Presentations - Part 2	
Finals	5/8	*Final project due	

Computer setup

- Text Editor
- Command line terminal
- GitHub Repository

https://github.com/jtladner/BIO682_Spring2023/tree/main/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)

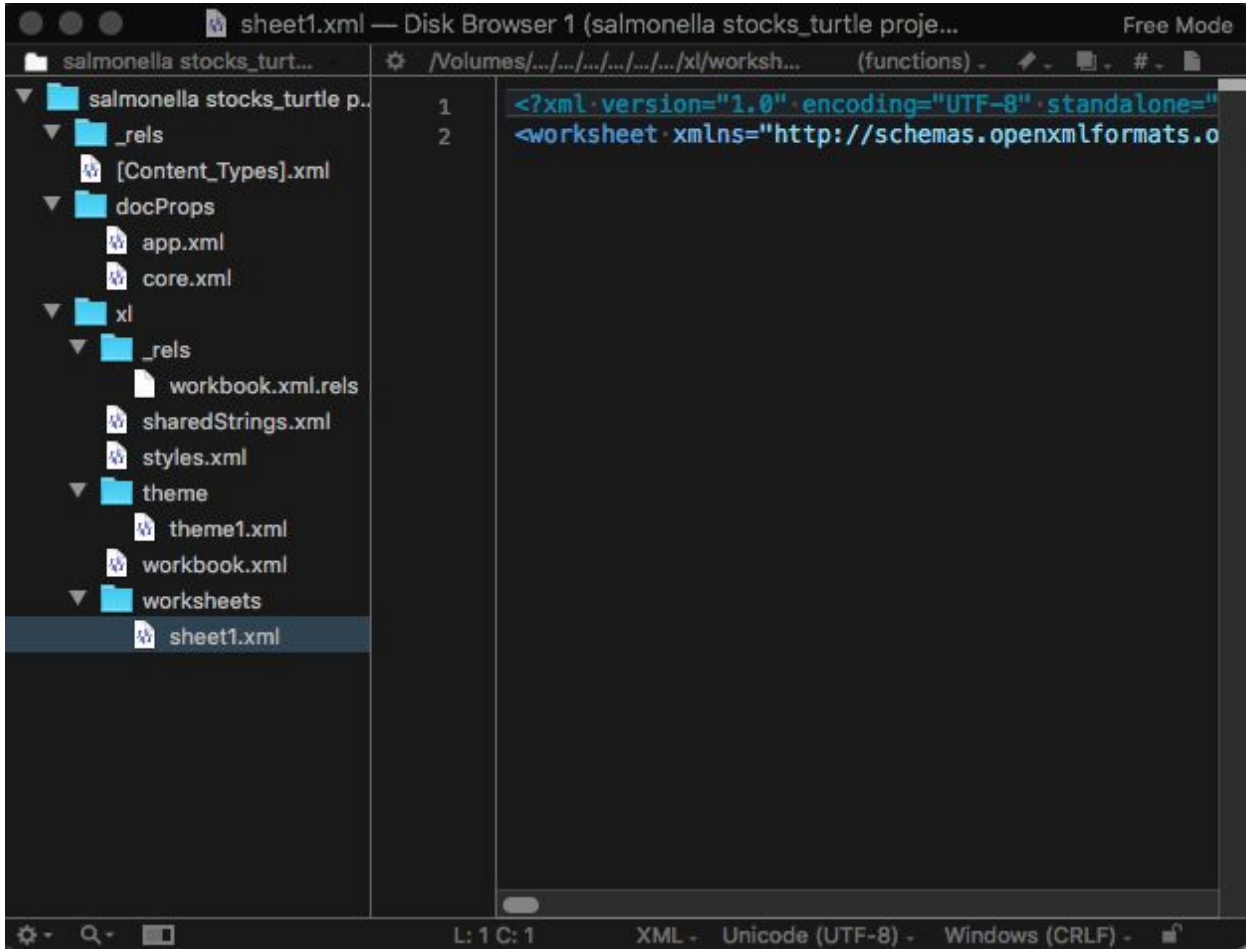
rich text (.rtf)

python script (.py)

tab-separated (.tsv)

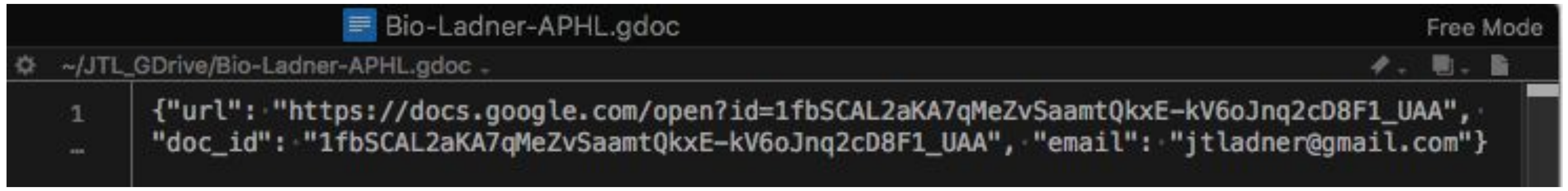
Viewing non-plain text in text editor

.xlsx/.docx



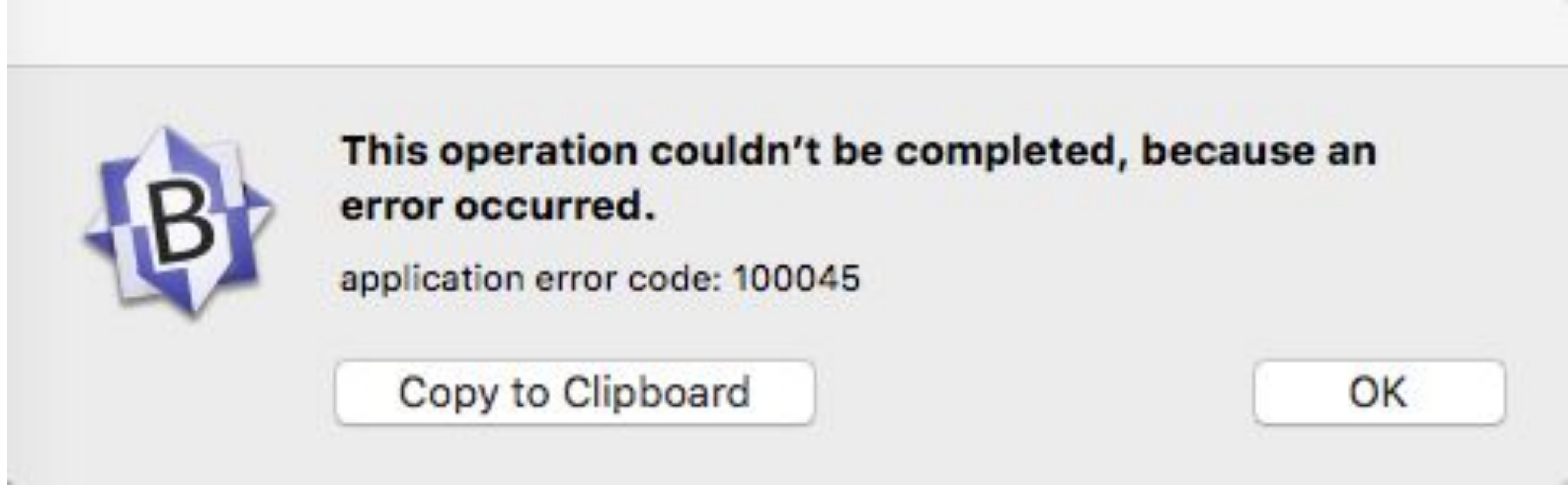
```
<?xml version="1.0" encoding="UTF-8" standalone="
<worksheet xmlns="http://schemas.openxmlformats.o
```

Google Doc



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

Google Sheet

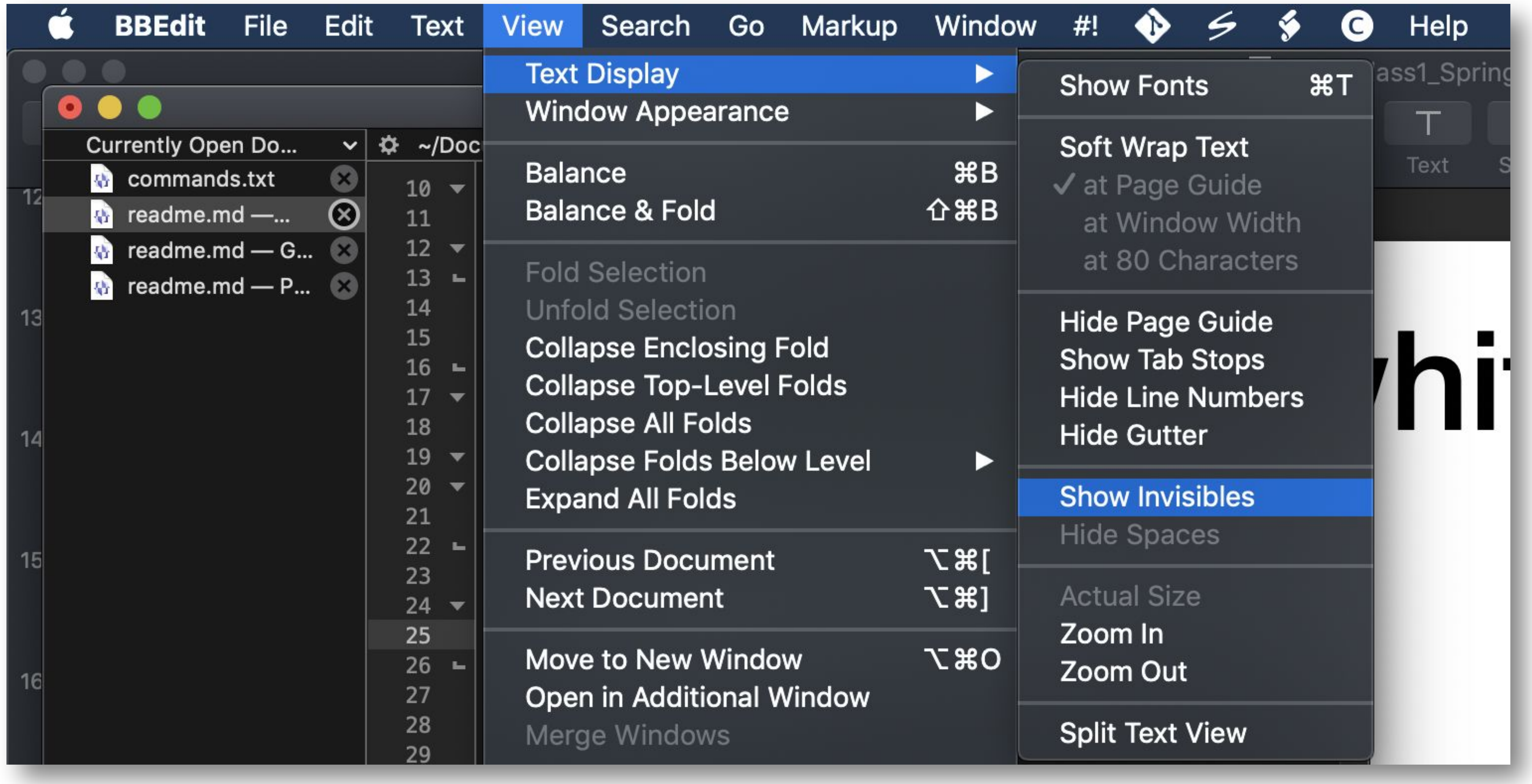

This operation couldn't be completed, because an error occurred.

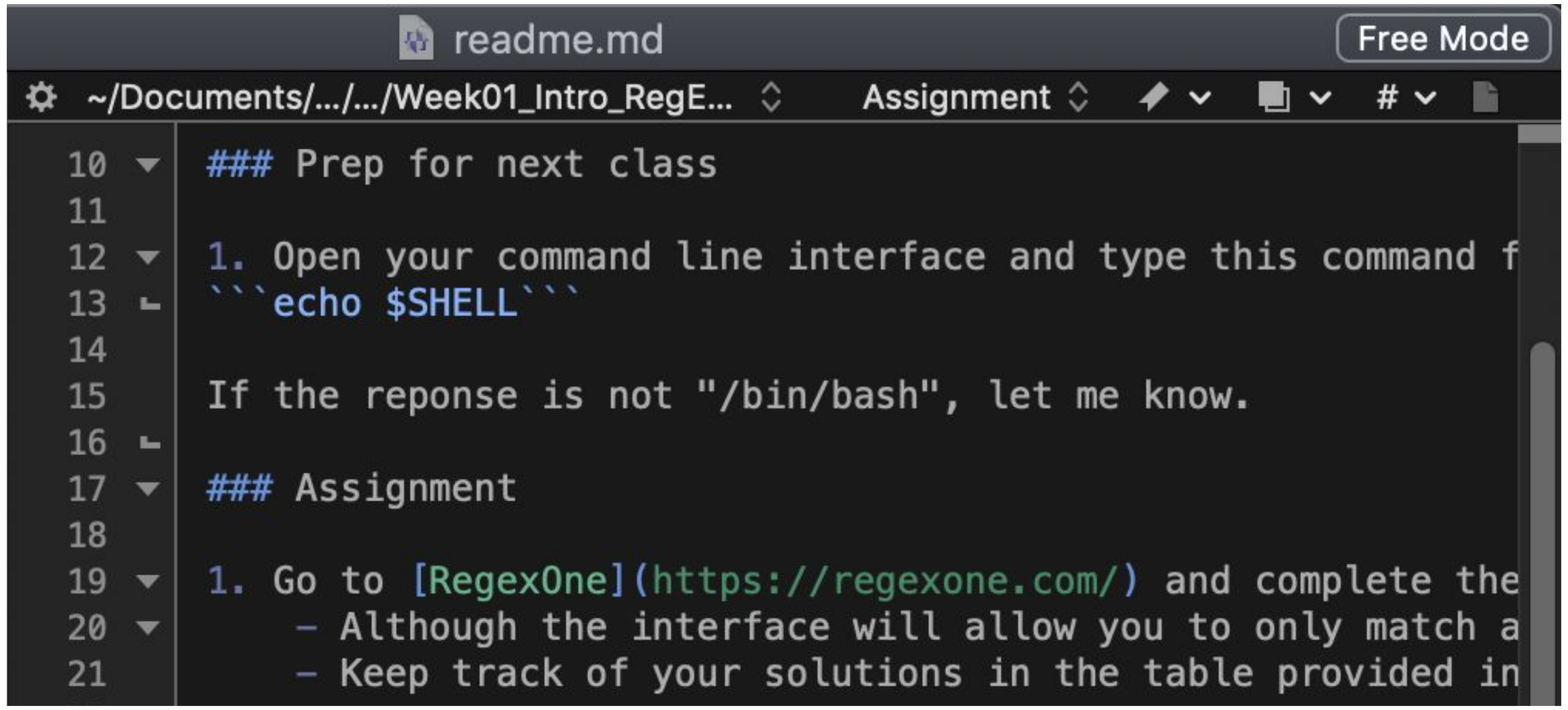
application error code: 100045

Whitespace

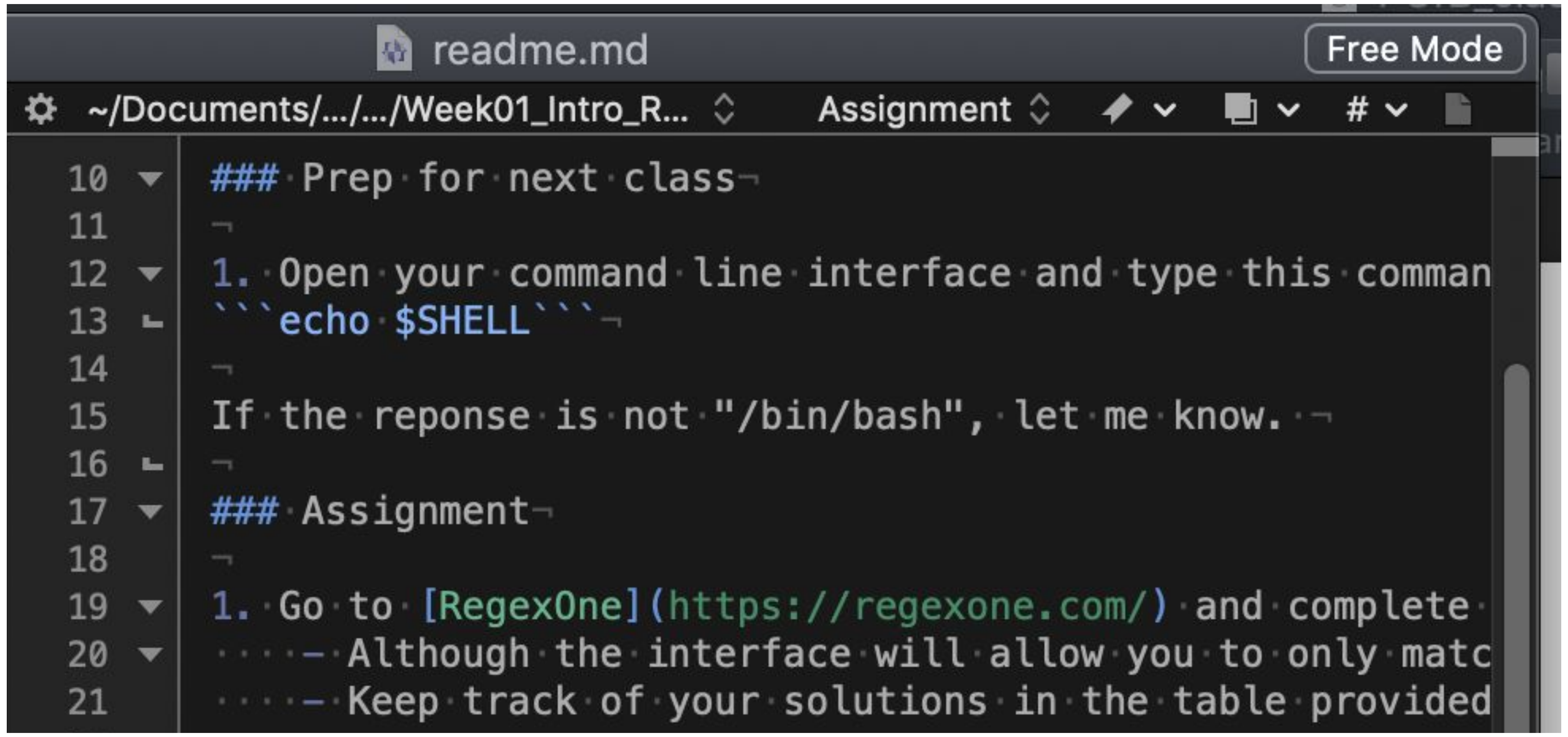
- Space
- Tab
- End of line

Visualizing white space (BBEdit)



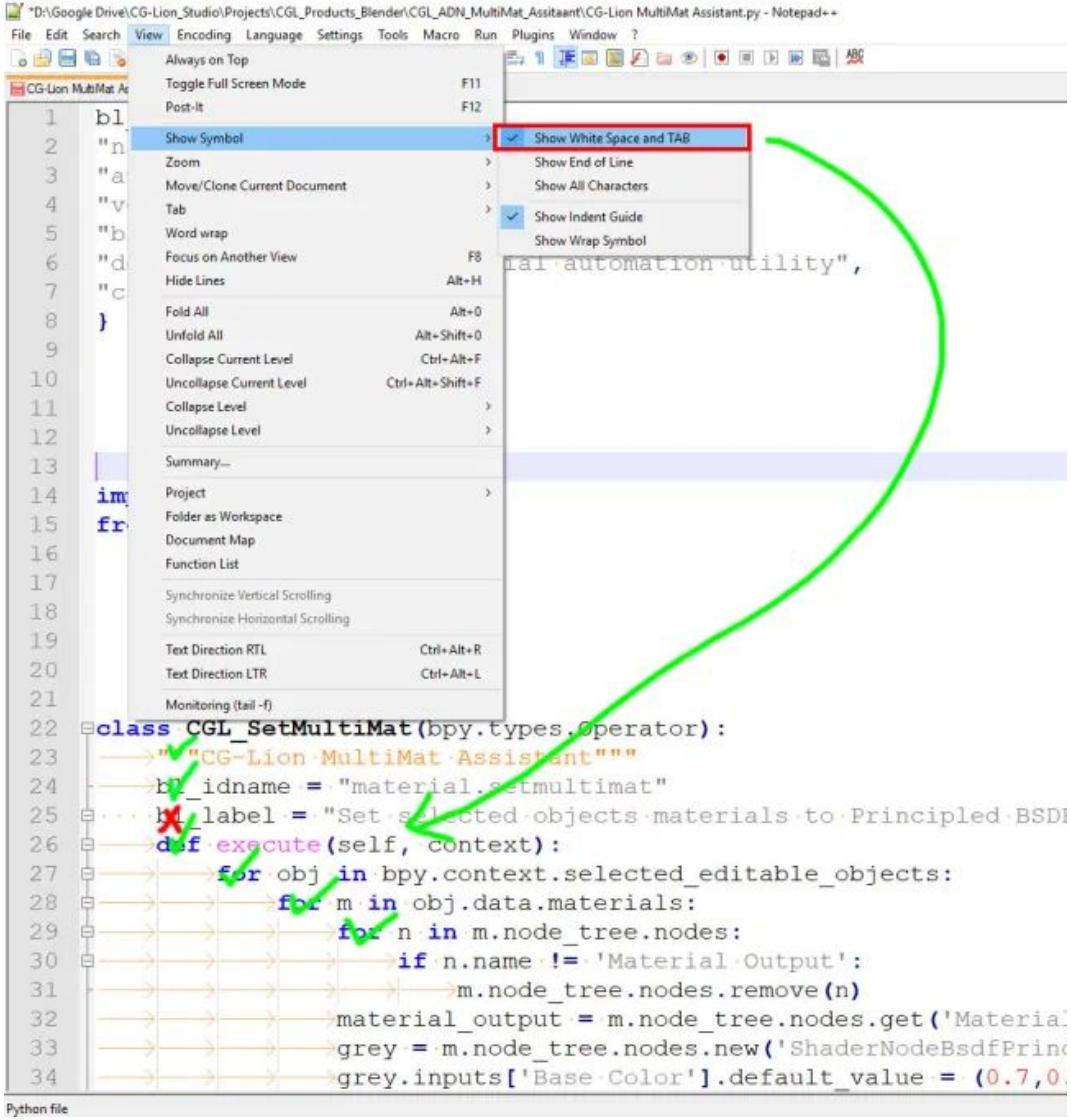


~~~
### Prep for next class

1. Open your command line interface and type this command f
```echo $SHELL```

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

### Assignment

1. Go to [RegexOne](https://regexone.com/) and complete the
    - Although the interface will allow you to only match a
    - Keep track of your solutions in the table provided in
~~~



~~~
###·Prep·for·next·class¬
¬
1.·Open·your·command·line·interface·and·type·this·comman
```echo·$SHELL```¬
¬
If·the·reponse·is·not·"/bin/bash",·let·me·know.·¬
¬
###·Assignment¬
¬
1.·Go·to·[RegexOne](https://regexone.com/)·and·complete·
····–·Although·the·interface·will·allow·you·to·only·matc
····–·Keep·track·of·your·solutions·in·the·table·provided
~~~

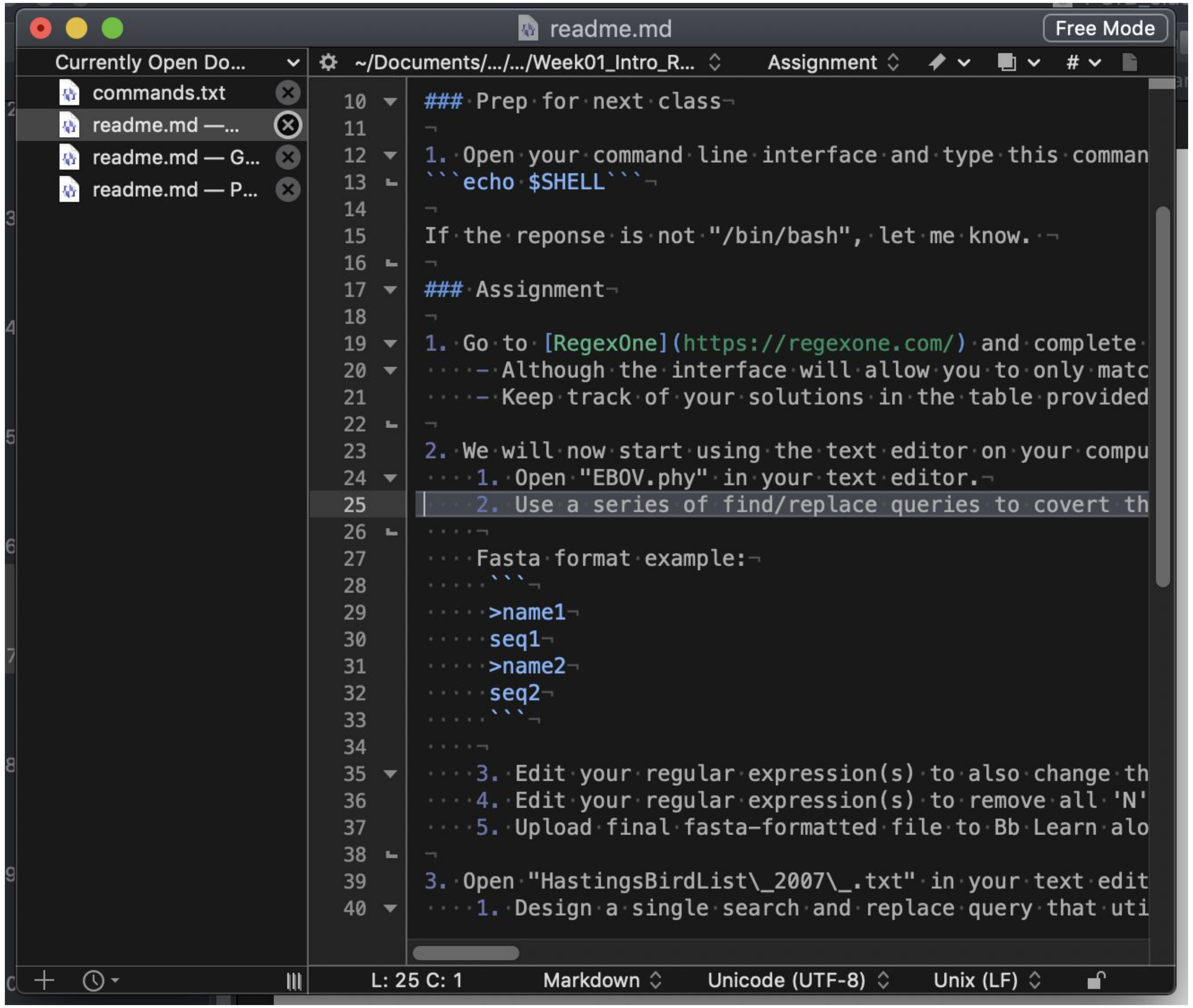
Visualizing white space (Notepad++)


*D:\Google Drive\CG-Lion_Studio\Projects\CGL_Products_Blender\CGL_ADN_MultiMat_Assitaant\CG-Lion MultiMat Assistant.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CG-Lion MultiMat A
Always on Top
Toggle Full Screen Mode F11
Post-It F12
Show Symbol
Zoom
Move/Clone Current Document
Tab
Word wrap
Focus on Another View F8
Hide Lines Alt+H
Fold All Alt+0
Unfold All Alt+Shift+0
Collapse Current Level Ctrl+Alt+F
Uncollapse Current Level Ctrl+Alt+Shift+F
Collapse Level
Uncollapse Level
Summary...
Project
Folder as Workspace
Document Map
Function List
Synchronize Vertical Scrolling
Synchronize Horizontal Scrolling
Text Direction RTL Ctrl+Alt+R
Text Direction LTR Ctrl+Alt+L
Monitoring (tail -f)
✓ Show White Space and TAB
Show End of Line
Show All Characters
✓ Show Indent Guide
Show Wrap Symbol
Python file

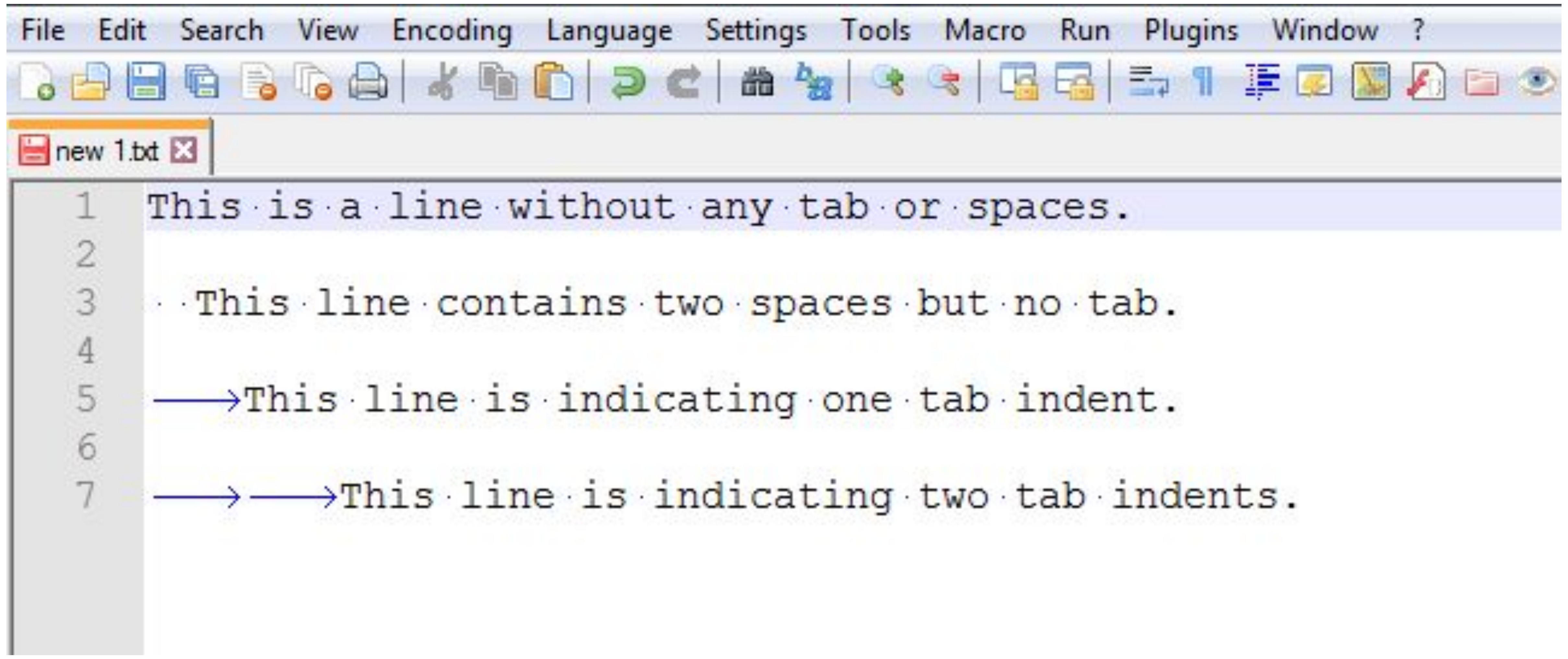
```
bl
"n
"a
"v
"b
"d                                  ial automation utility",
"c
}



im
fr







class CGL_SetMultiMat(bpy.types.Operator):
    """CG-Lion MultiMat Assistant"""
    bl_idname = "material.setmultimat"
    bl_label = "Set selected objects materials to Principled BSDI
    def execute(self, context):
        for obj in bpy.context.selected_editable_objects:
            for m in obj.data.materials:
                for n in m.node_tree.nodes:
                    if n.name != 'Material Output':
                        m.node_tree.nodes.remove(n)
                material_output = m.node_tree.nodes.get('Materia
                grey = m.node_tree.nodes.new('ShaderNodeBsdfPrinc
                grey.inputs['Base Color'].default_value = (0.7,0.
```

BBEdit

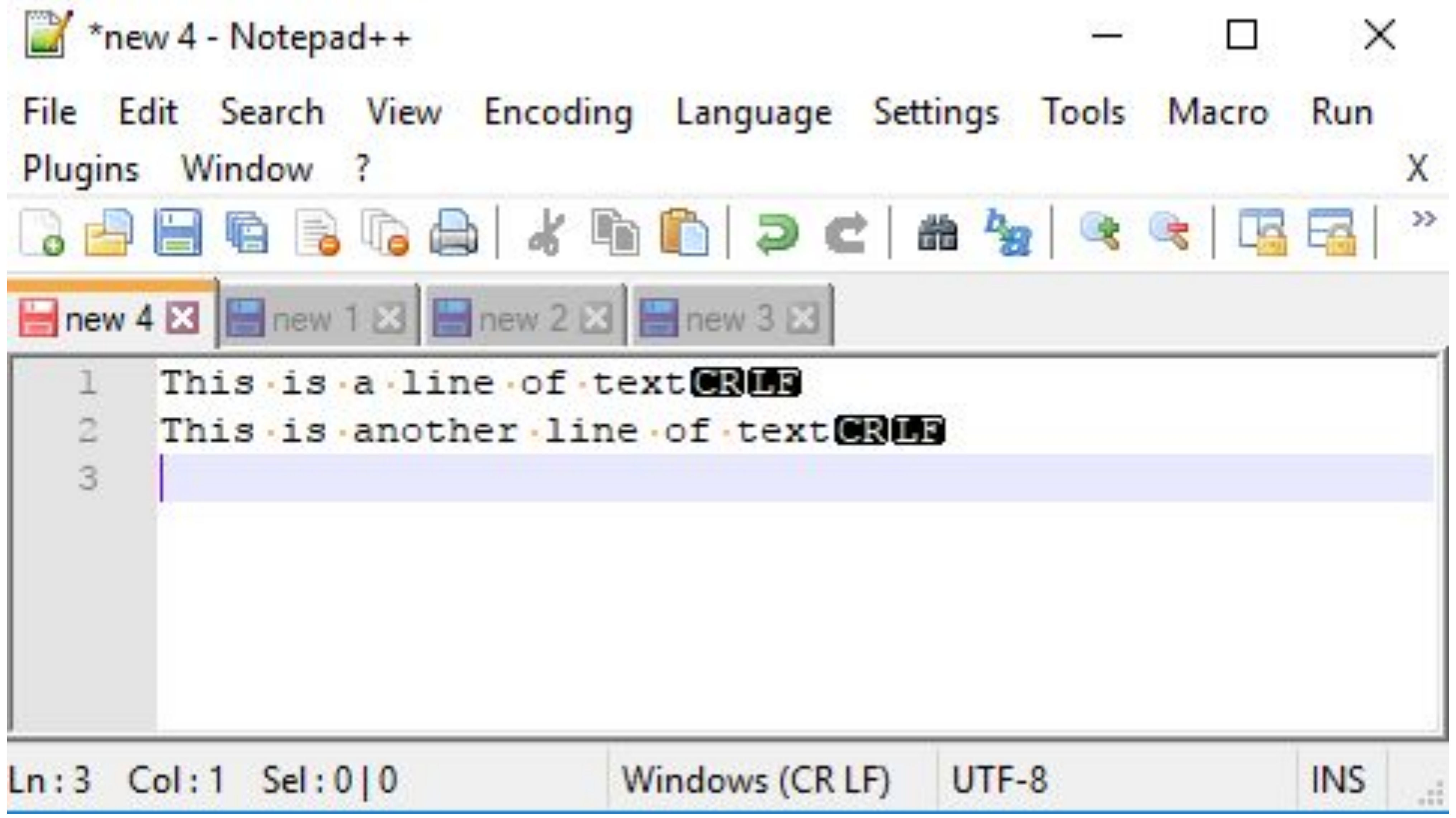
Notepad++



~~~
### Prep for next class

1. Open your command line interface and type this comman
```echo $SHELL```

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

### Assignment

1. Go to [RegexOne](https://regexone.com/) and complete
    - Although the interface will allow you to only matc
    - Keep track of your solutions in the table provided

2. We will now start using the text editor on your compu
    1. Open "EBOV.phy" in your text editor.
    2. Use a series of find/replace queries to covert th

    Fasta format example:
    ~~~
    >name1
    seq1
    >name2
    seq2
    ```

    3. Edit your regular expression(s) to also change th
    4. Edit your regular expression(s) to remove all 'N'
    5. Upload final fasta-formatted file to Bb Learn alo

3. Open "HastingsBirdList_2007_.txt" in your text edit
    1. Design a single search and replace query that uti
```



```
This·is·a·line·without·any·tab·or·spaces.

··This·line·contains·two·spaces·but·no·tab.

→This·line·is·indicating·one·tab·indent.

→→This·line·is·indicating·two·tab·indents.
```



```
This·is·a·line·of·textCRLF
This·is·another·line·of·textCRLF

```

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)
```

Anchors

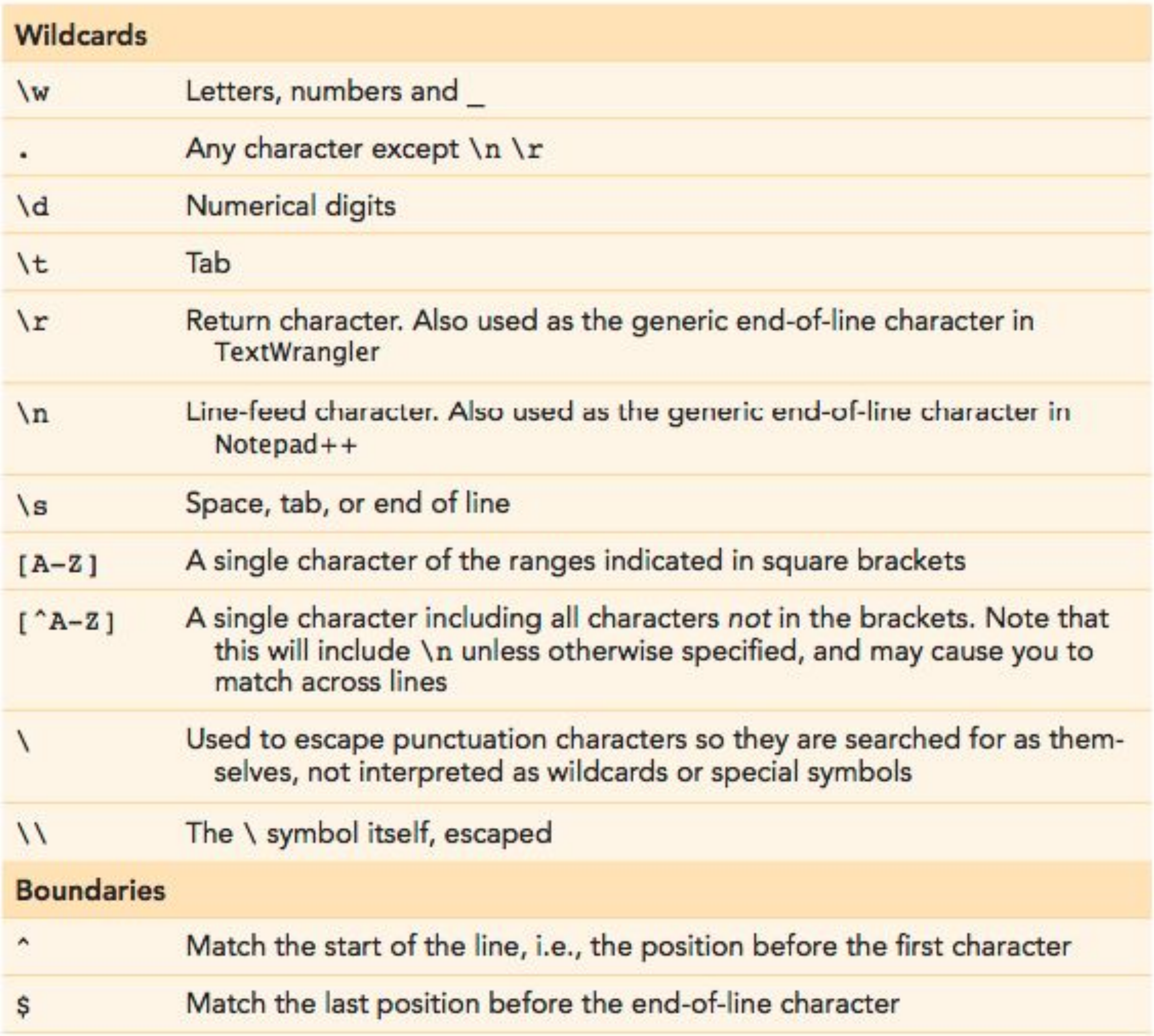
^

$

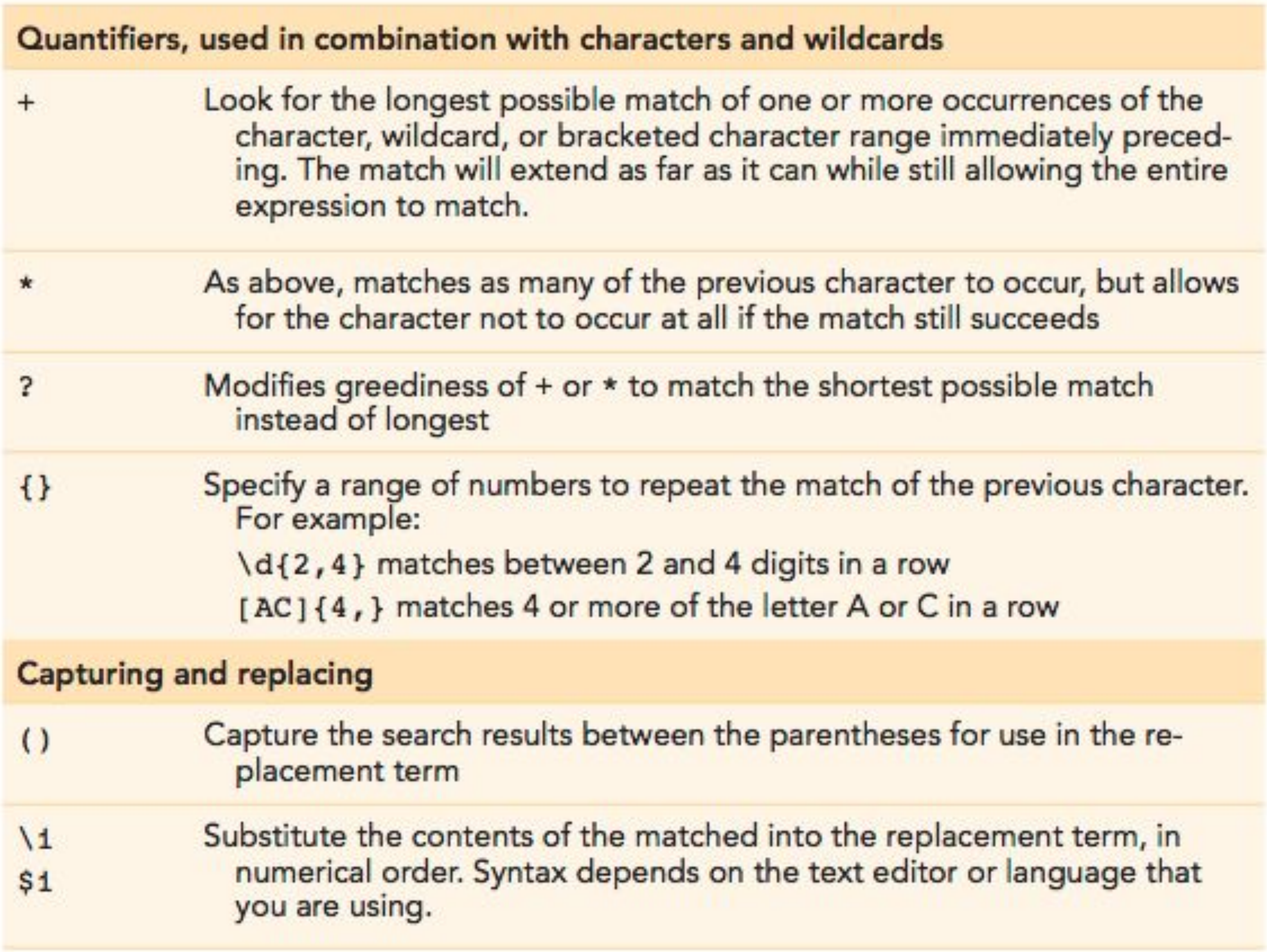
Tips

- 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



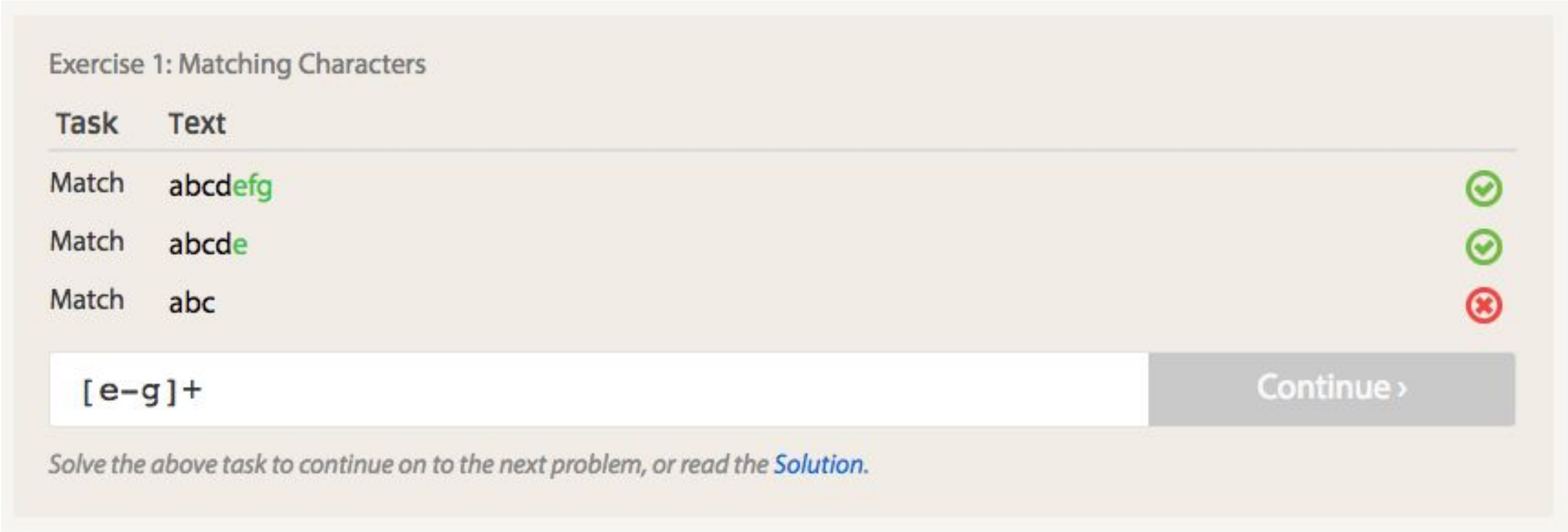
Quantifiers, used in combination with characters and wildcards	
+	Look for the longest possible match of one or more occurrences of the character, wildcard, or bracketed character range immediately preceding. The match will extend as far as it can while still allowing the entire expression to match.
*	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.

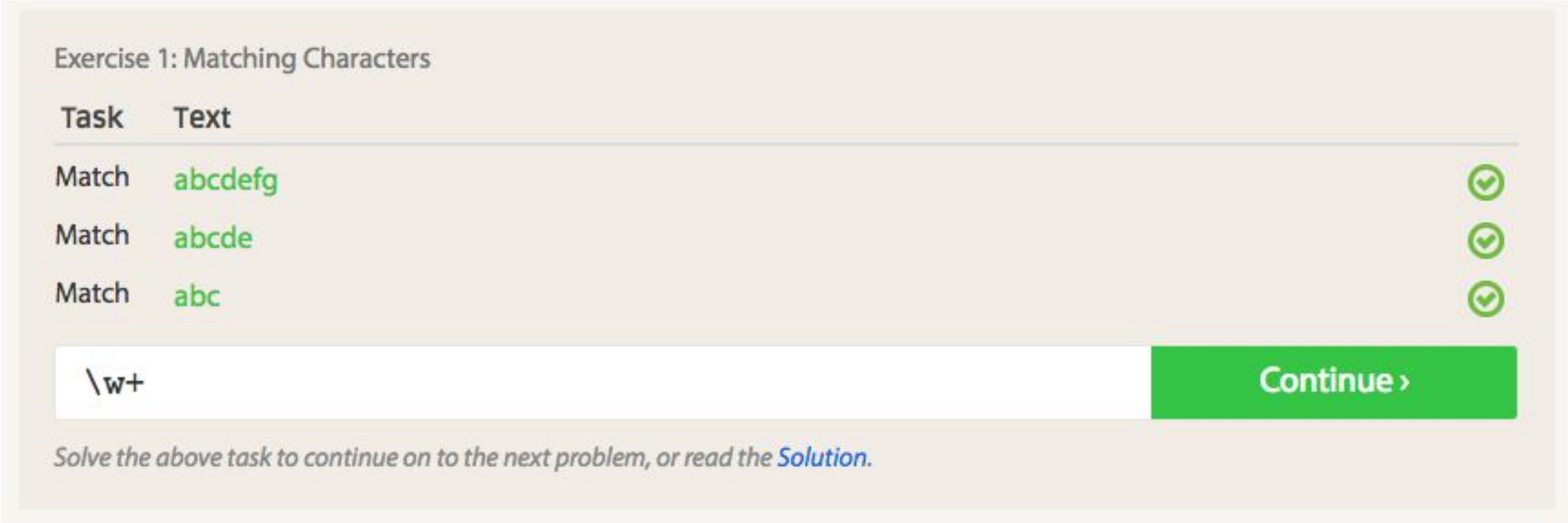


Exercise 1: Matching Characters

Task	Text	
Match	abcdefg	✓
Match	abcde	✓
Match	abc	✗

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

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



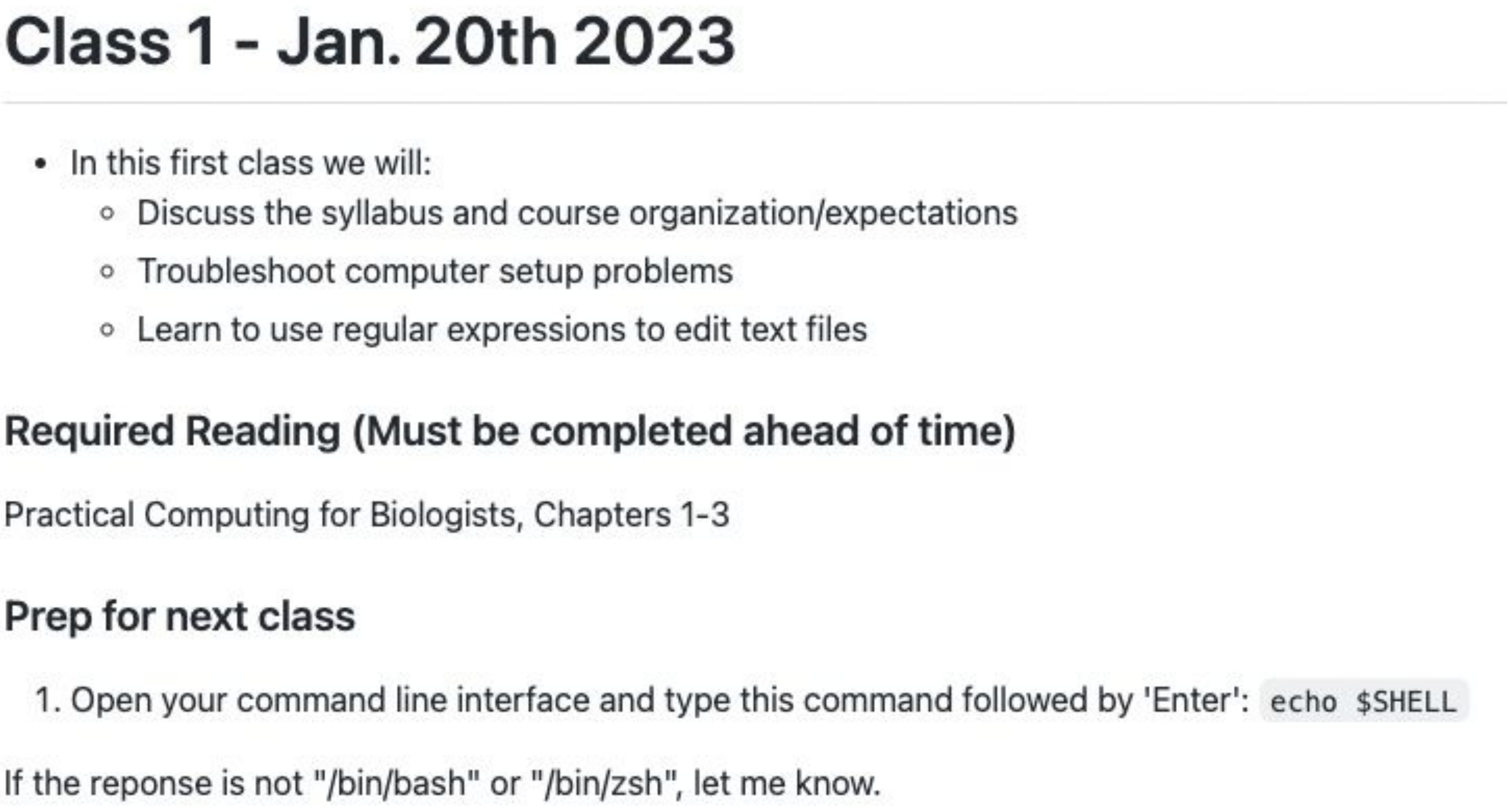
Exercise 1: Matching Characters

Task	Text	
Match	abcdefg	✓
Match	abcde	✓
Match	abc	✓

```
\w+
```

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

“Prep for next class”



Class 1 - Jan. 20th 2023

- 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

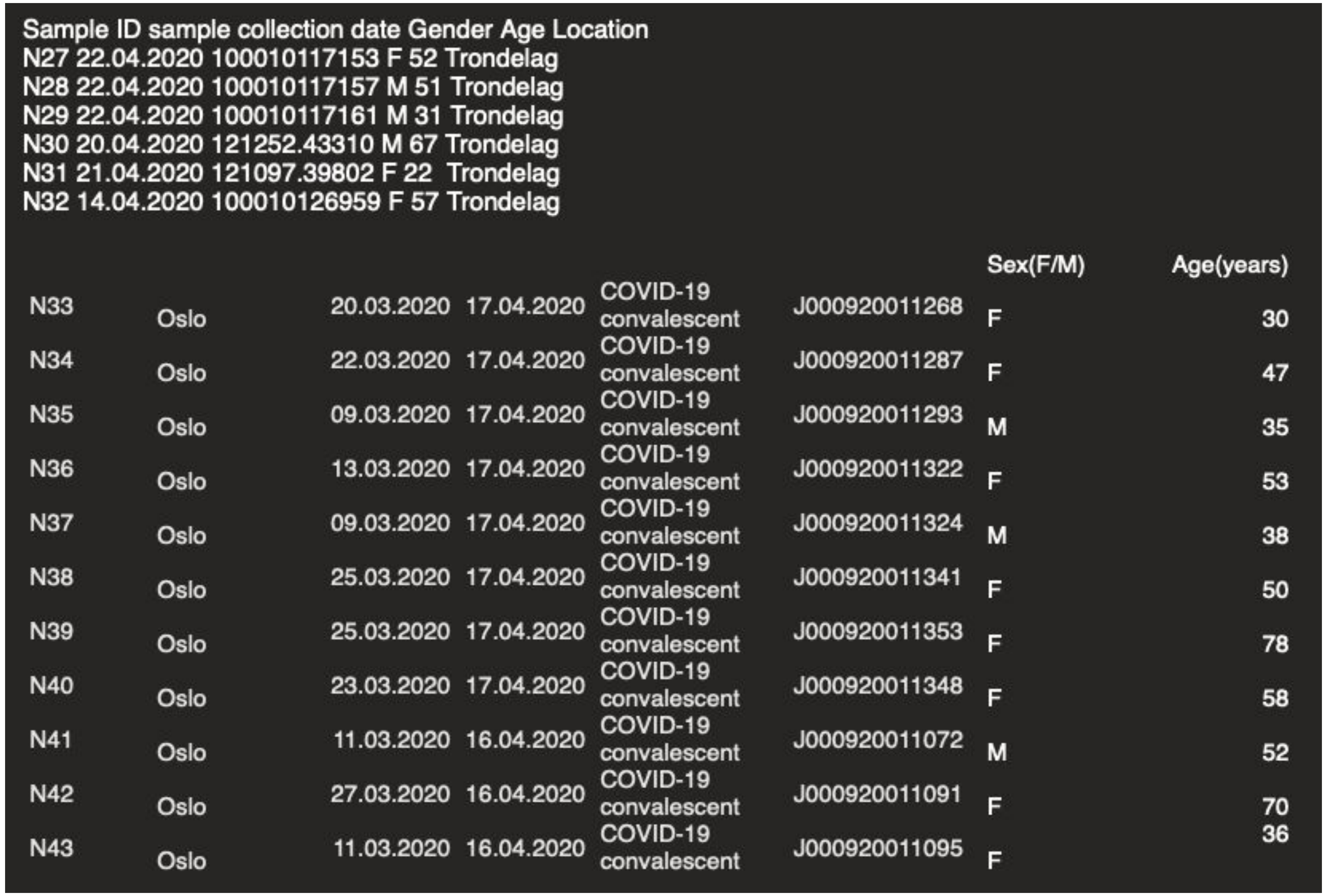
1. Open your command line interface and type this command followed by 'Enter': `echo $SHELL`

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

Text editor regex demos

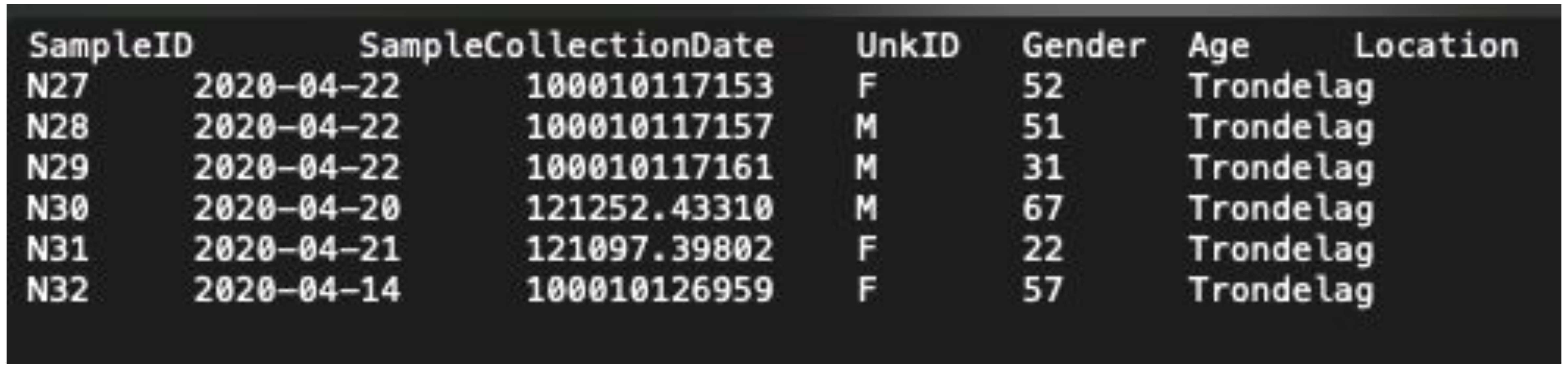
Start
(email)

End
(tsv)

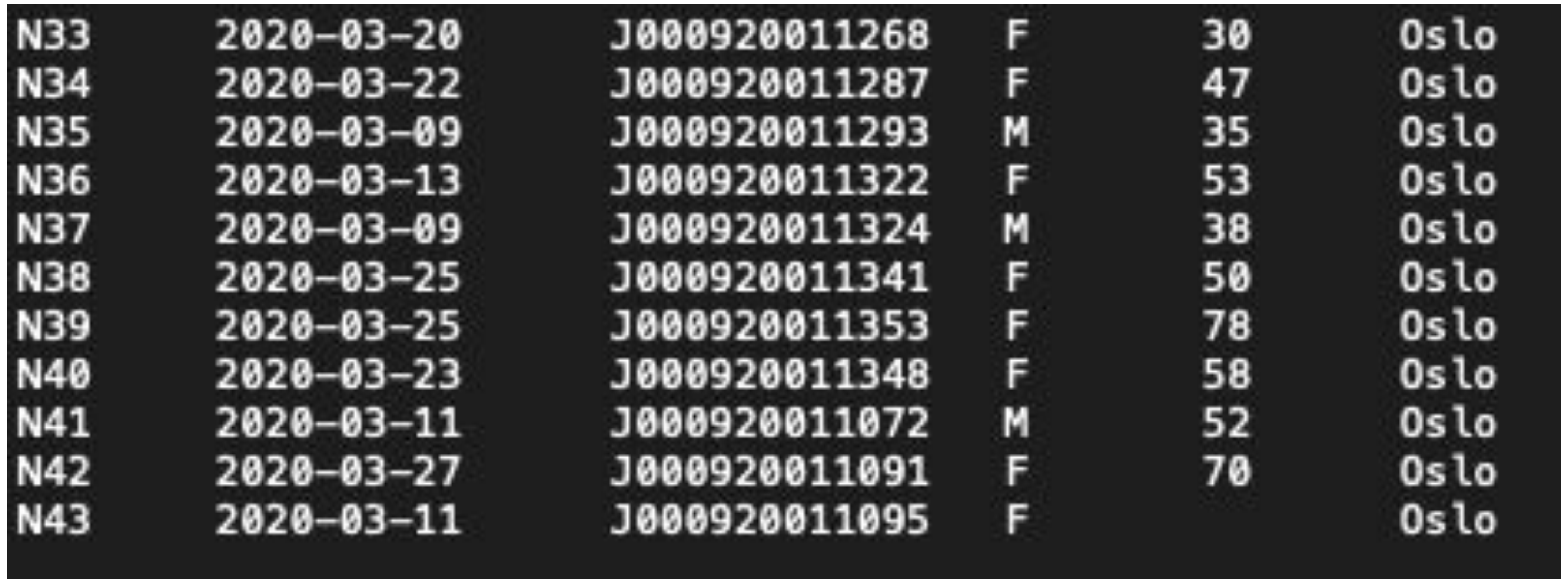


Sample ID sample collection date Gender Age Location
N27 22.04.2020 100010117153 F 52 Trondelag
N28 22.04.2020 100010117157 M 51 Trondelag
N29 22.04.2020 100010117161 M 31 Trondelag
N30 20.04.2020 121252.43310 M 67 Trondelag
N31 21.04.2020 121097.39802 F 22 Trondelag
N32 14.04.2020 100010126959 F 57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



```
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
```