CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson

Fall 2018

Swarthmore College

Outline Sept 12:

Sit at a new computer!

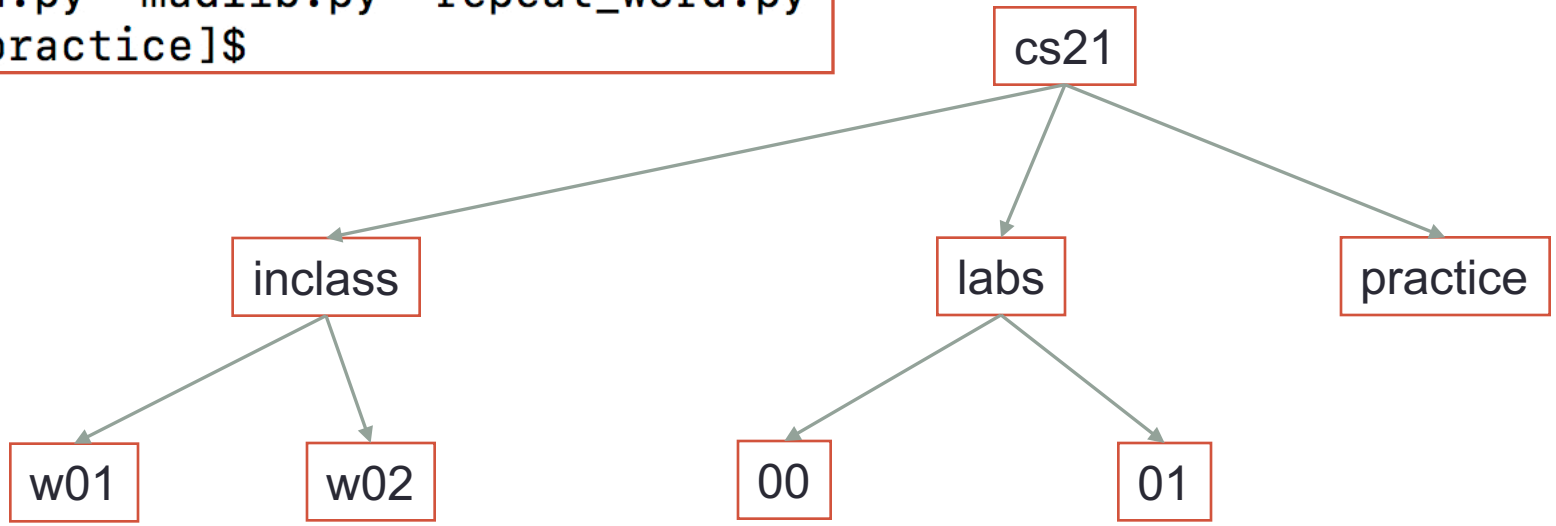
- Recap strings + finish **xo.py**
- Sequences and sequence operations
- For loops
  - Print sequence of numbers, sequence of characters
  - **square.py**, **blastoff.py**, **indexing.py**
- Fri: accumulator pattern (**miles_loop.py**)

Reminders

- Lab 1 due Saturday night
- Office hours 3-5pm Friday (+ when my door is open)
- Quiz 1: next Friday (9/21)
- Let me know about quiz conflicts

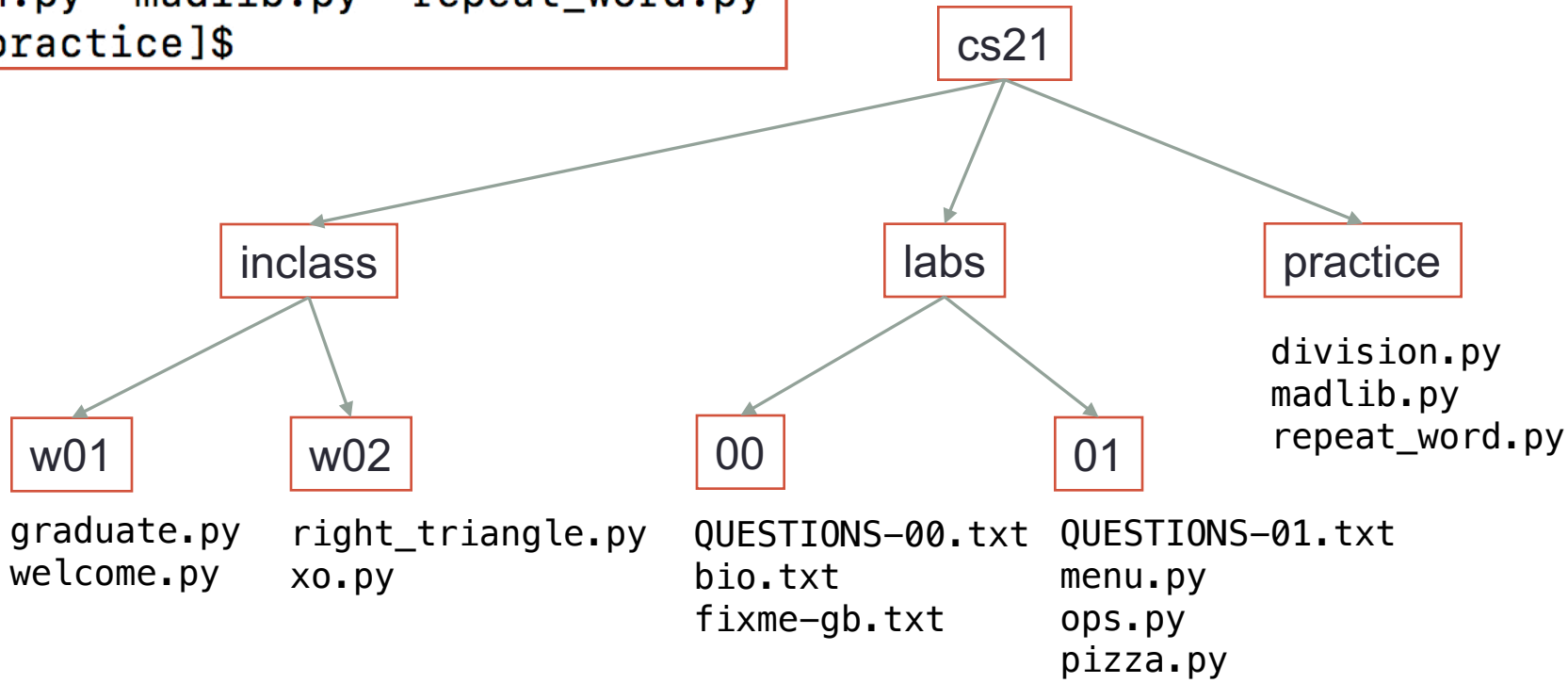
Practice directory

```
butter[~]$ ls
Desktop/  Documents/  cs21/  f17cs21/
butter[~]$ cd cs21/
butter[cs21]$ ls
inclass/  labs/  practice/
butter[cs21]$ cd practice/
butter[practice]$ ls
division.py  madlib.py  repeat_word.py
butter[practice]$
```



Practice directory

```
butter[~]$ ls
Desktop/  Documents/  cs21/  f17cs21/
butter[~]$ cd cs21/
butter[cs21]$ ls
inclass/  labs/  practice/
butter[cs21]$ cd practice/
butter[practice]$ ls
division.py  madlib.py  repeat_word.py
butter[practice]$
```



Workflow so far…

- **up** and **down** arrows for cycling through previous commands
- **Alt-tab** for switching between terminal and atom
- **tab** to autocomplete a file name or command

- ATOM
- **Ctrl-s** for save
- **Ctrl-n** for new file
- **Ctrl-z** undo
- **Ctrl-w** close window

More keyboard shortcuts here

https://www.cs.swarthmore.edu/courses/CS21Labs/f18/docs/using-atom.html

- Python style guide: https://www.python.org/dev/peps/pep-0008/

Studying Science at Swarthmore

- 6:30 PM: 30 min plus Q & A
- **Thursday, September 20**
- Science Center 199
- Presented by Amy Cheng Vollmer,
  Dept. of Biology
- Advice and recommendations compiled from her interviews of many faculty in the Division of Natural Sciences and Engineering
- Recommended by SAMs & geared toward students of all class years
- Sponsored by the Dean of Students Division and Department of Biology

Recap strings

String operations

- Length: **len(“swarthmore”)** **10**
- Empty string: **“”** length?
- Concatenation: **“a” + “b”** **“ab”**
- Repetition: **“a”*5** **“aaaaa”**
- Strings are ***sequences*** of ***characters***

Program to try with a partner

1) Ask user for integer n and print 2n x's.

```
Enter a positive integer: 3
xxxxxx
```

2) On the next line print 2n o's.

```
Enter a positive integer: 3
xxxxxx
oooooo
```

3) Finally print "xo" n times.

```
Enter a positive integer: 3
xxxxxx
oooooo
xoxoxo
```

Sequences and loops demo

Motivation: pretty print

Sequences

- strings (chars)

0	1	2	3	4	5	6	7	8	9
S	w	a	r	th	m	o	r	e	

- range (integers)
- lists (anything)

```
for i in range(10):
    print(i)
```

integer

~~i=0~~

~~i=1~~

~~i=2~~

i=3

⋮

i=9

Sequences (strings, lists, ranges)

- All sequences have a way to get the *length*: **len(<seq>)**
- All sequences support *indexing*: **<seq>[<int>]**
- All sequences have an *"empty"* version:
  - Strings: **""**
  - Lists: **[]**
  - Ranges: **range(0)**
- *Concatenation* and *repetition*: only for strings and lists

Types so far

- int
- float
- str
- list
- range

For loop exercises

1) square.py
2) blastoff.py
3) indexing.py
4) reverse.py (in practice directory)

Tips

Don't forget :!

```
def main():
    → y = ...
    print(.....)
```

balanced parentheses

Note: Python puts blue lines under balanced parentheses!

If your terminal says

>>>

YOU ARE IN THE INTERPRETER!

```
main()
```

align with def main():

Reminders from the Ninjas!