

CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson

Fall 2017

Swarthmore College

Outline Oct 13:

Right now: find a *new* partner and work on the stack diagram handout, then work on max_circle.py

- Return Lab 4
- Practice stack diagrams (pick up a handout)
- List accumulators and mutation in graphics
- Go over **max_circle.py** program
- Try/except in python: **try_except.py**

Notes

- **Lab 5** due **tomorrow** night
- **Office Hours 3-5pm today!!**
- **I'll be around over break if you want to meet**

Screenshots and Videos: please email!

- Windows Videos: <https://www.hongkiat.com/blog/win-screen-recording-softwares/>
- Mac Videos: Quicktime
- Linux screenshot:

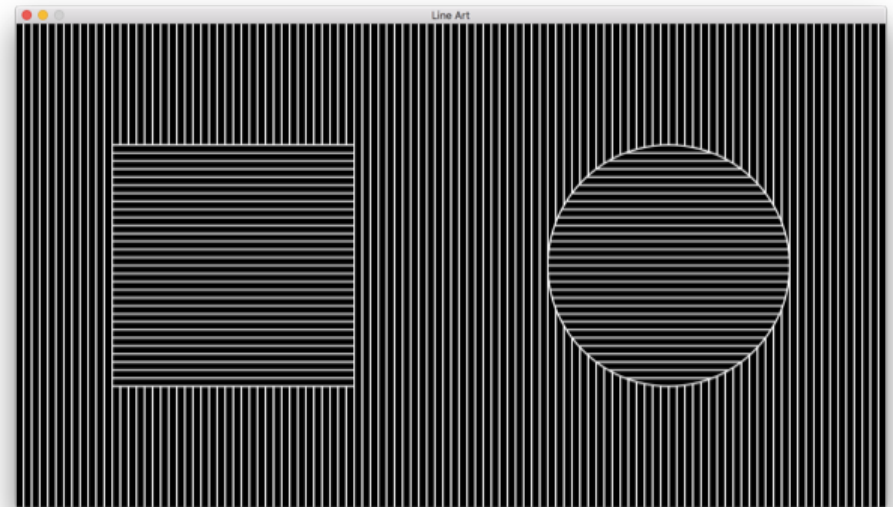
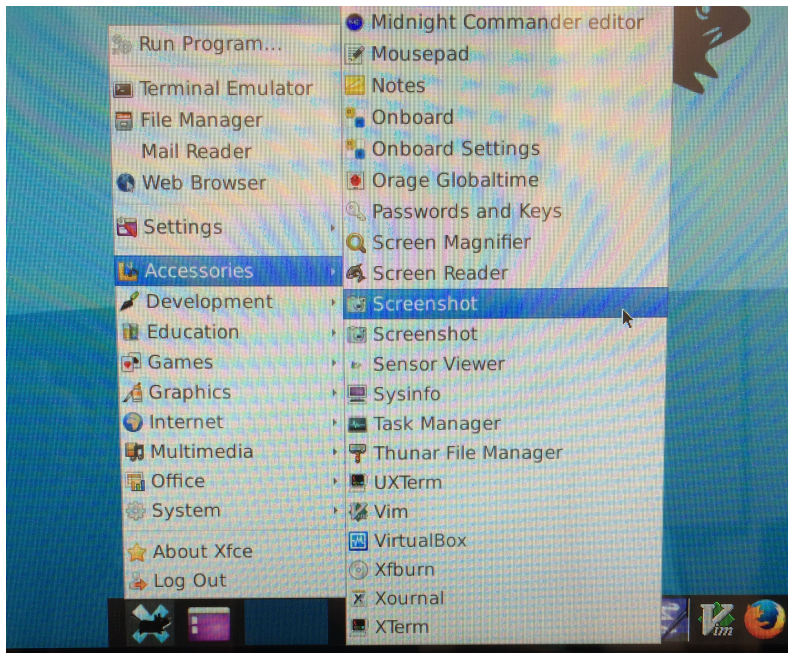


Image by Rich Wicentowski

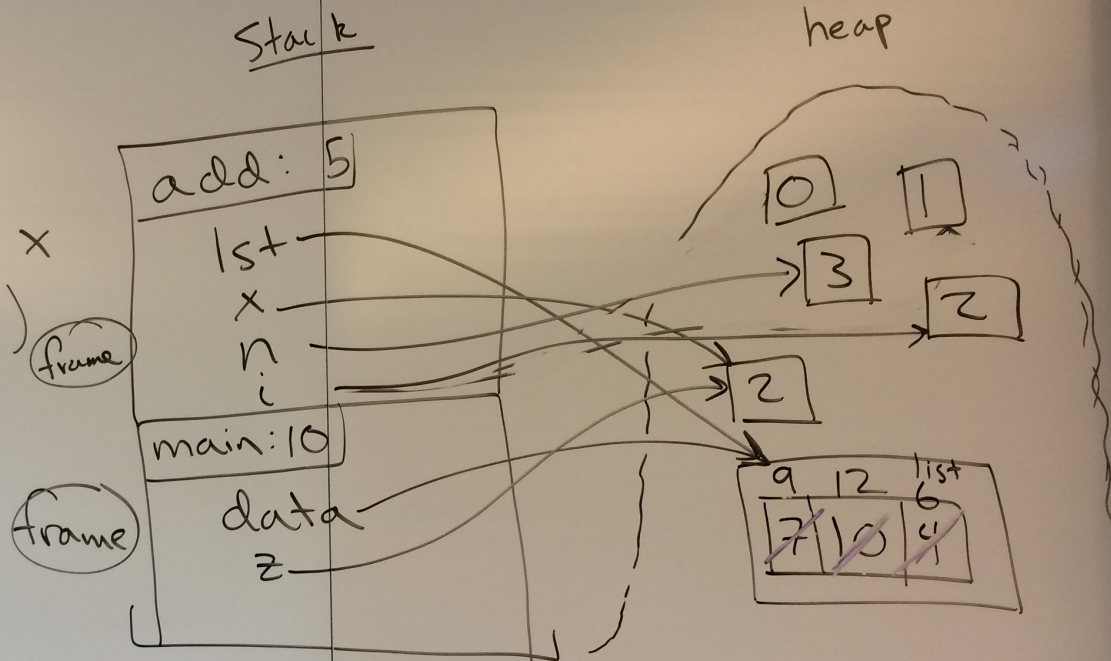
Stack Diagram Practice

(see Handout 2)

```

1 def add(lst, x):
2     n = len(lst)
3     for i in range(n):
4         LHSlst[i] = lst[i] + x
5         print("done adding")
6
7 def main():
8     data = [7, 10, 4]
9     z = 2
10    add(data, z)
11    print(data)
12
13 main()

```



Stack Handout Solution

Programs for today

- `inclass/week06/stack_handout.py`
- `inclass/week06/max_circle.py`
- `inclass/week06/try_except.py`
- `practice/snow_challenge.py`

Try/Except (start)

Try/Except

- Whenever an error message is produced, our code terminates and we can't execute any further code
- Try/except is a way to navigate/circumvent errors
- Example: if a user should enter an integer but they don't, allow them to try again
- A beginning of this example is shown below (add a while loop after break):

```
try:
    n = int(input("Enter an integer: "))
except:
    print("not a valid number!")

print("continuing code...")
```