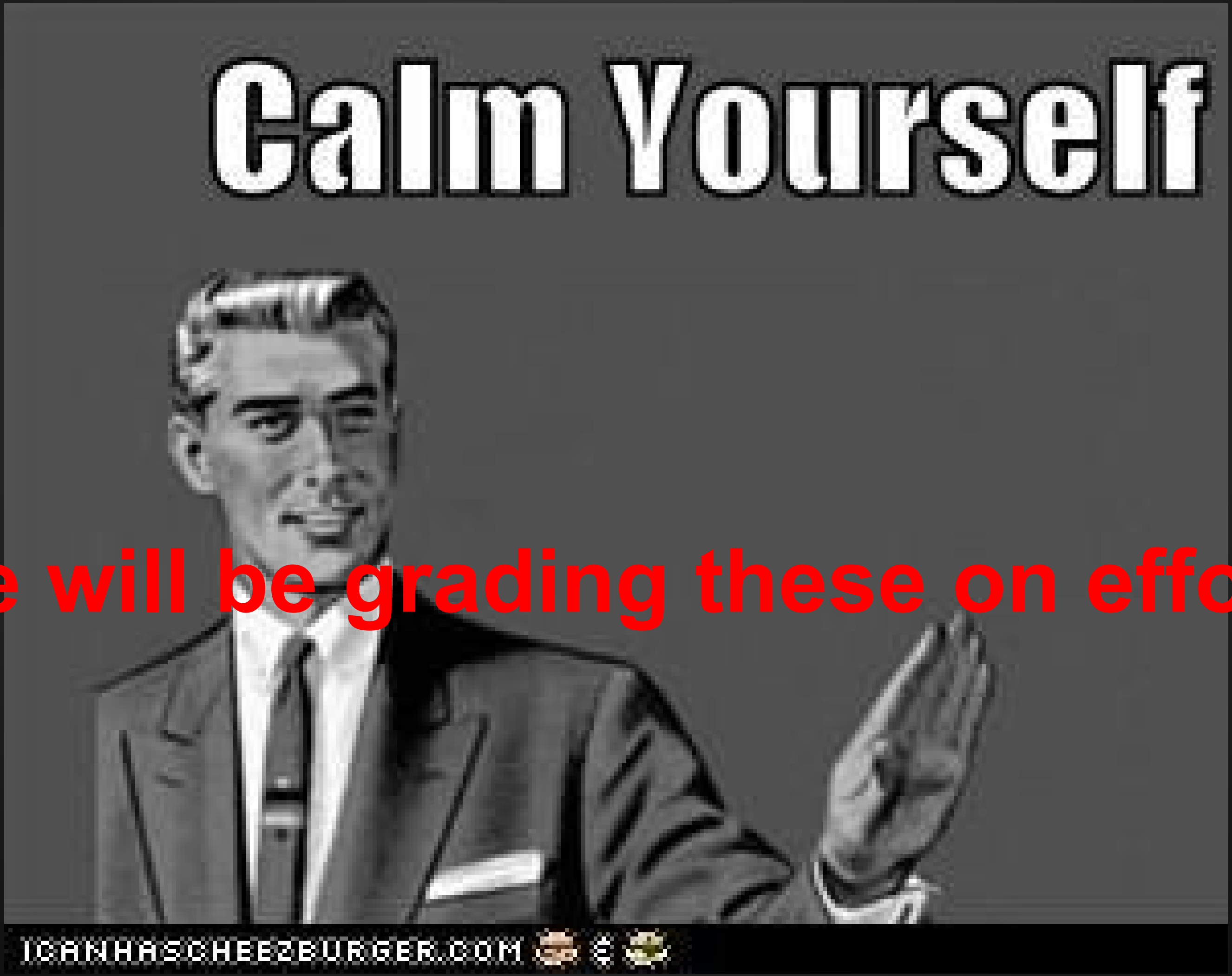
Loops

"We're losing track of the vastness of the potential for computer science. We really have to revive the beautiful intellectual joy of it, as opposed to the business potential"
-- Jaron Lanier

Recursion Lab & Homework



We will be grading these on effort!

Common mistakes from last week

- Variables and functions can only be assigned to 1 thing at a time!
- Defining it again, overrides the first definition

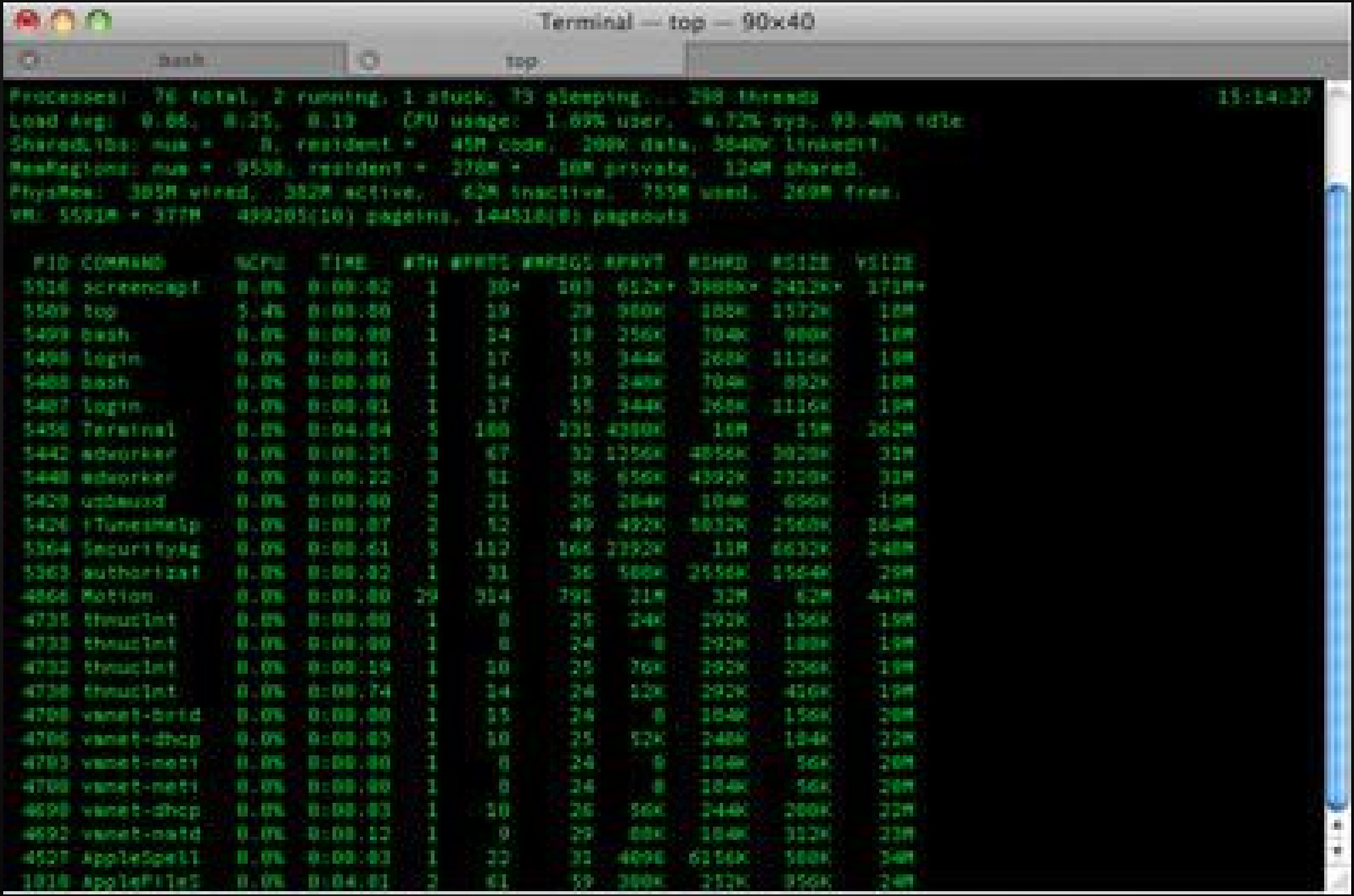
```
def foo():
    print 'this is foo'

#prints "this is foo'
foo()

def foo():
    print 'different foo'

#prints 'different foo'
foo()
```

Terminal



```
Processes:  76 total, 2 running, 1 stuck, 73 sleeping... 298 threads                   15:14:27
Load Avg:  0.06,  0.25,  0.19    CPU usage:  1.69% user,  4.72% sys, 93.40% idle
SharedLibs: num =    8, resident =   45M code,  200K data, 3540K linkedit.
MemRegions: num =  9530, resident =  278M +   10M private,  124M shared.
PhysMem:  305M wired,  382M active,   62M inactive,  755M used,  269M free.
VM: 5591M + 377M   499205(10) pageins, 144518(0) pageouts

  PID COMMAND      %CPU   TIME    #TH #PRTS #MREGS #RPRVT  RSHRD  RSIZE  VSIZE
 5516 screencapt   0.0%  0:00.02    1   30+    103  612K+ 3988K+ 2412K+  171M+
 5509 top          5.4%  0:00.50    1    19     29   980K   188K  1572K    18M
 5499 bash         0.0%  0:00.00    1    14     19   256K   704K   900K    18M
 5498 login        0.0%  0:00.01    1    17     55   344K   268K  1116K    19M
 5488 bash         0.0%  0:00.00    1    14     19   248K   704K   892K    18M
 5487 login        0.0%  0:00.01    1    17     55   344K   268K  1116K    19M
 5456 Terminal     0.0%  0:04.84    5   100    231  4380K    16M    15M   262M
 5442 mdworker     0.0%  0:00.21    3    67     32 1256K  4856K  3028K    31M
 5440 mdworker     0.0%  0:00.22    3    51     36   656K  4392K  2320K    31M
 5429 usbmuxd      0.0%  0:00.00    2    21     26   284K   104K   696K    19M
 5426 iTunesHelp   0.0%  0:00.07    2    52     49   492K  5832K  2560K   164M
 5364 SecurityAg   0.0%  0:00.41    3   112    166  2392K   11M  6632K   248M
 5363 authorizat   0.0%  0:00.02    1    31     36   588K  2556K  1564K    29M
 4866 Motion       0.0%  0:03.80   29   314    791    21M    32M    62M   447M
 4735 thnuclnt     0.0%  0:00.00    1     8     25    24K   292K   136K    19M
 4733 thnuclnt     0.0%  0:00.00    1     8     24      0   292K   100K    19M
 4732 thnuclnt     0.0%  0:00.19    1    10     25    76K   292K   236K    19M
 4730 thnuclnt     0.0%  0:00.74    1    14     24    12K   292K   416K    19M
 4708 vmnet-brid   0.0%  0:00.00    1    15     24      0   184K   156K    20M
 4706 vmnet-dhcp   0.0%  0:00.03    1    10     25    52K   240K   184K    22M
 4703 vmnet-neti   0.0%  0:00.00    1     8     24      0   184K    56K    20M
 4700 vmnet-neti   0.0%  0:00.00    1     8     24      0   184K    56K    20M
 4698 vmnet-dhcp   0.0%  0:00.03    1    10     26    56K   244K   200K    22M
 4692 vmnet-natd   0.0%  0:00.12    1     9     29    88K   184K   312K    23M
 4527 AppleSpell   0.0%  0:00.03    1    33     31   4096  6156K   580K    34M
 1018 AppleFileS   0.0%  0:04.01    2    61     59   300K   252K   856K    24M
```

Terminal

- ls - list everything in the current folder
- cd FOLDER - enter FOLDER
- cd .. - go up 1 folder
- mkdir FOLDER - create FOLDER
- rm FILE - delete FILE
- rm -rf FOLDER - delete FOLDER and it's contents
  - THESE FILES SKIP THE TRASH
  - THIS IS IRREVERSIBLE
- touch FILE - create empty FILE
- python FILE - run FILE in Python
- python - start Python interpreter
- Ctrl-C - exit a running program
- Ctrl-D - exit Python interpreter

Random Numbers

```
import random

print random.random()#random float between 0.0 and 1.0
#does not include 1.0!

print random.randint(1,100)#random integer between 1 and 100

print random.random() * 5#random float between 0.0 and 5.0

print (random.random() * 5) + 5
#random float between 5.0 and 10.0
```

What are loops?

What are loops?

Loops allow code to be executed multiple times

```
def triangle(size):
    winston.forward(size)
    winston.left(120)
    winston.forward(size)
    winston.left(120)
    winston.forward(size)
    winston.left(120)
```

```
def triangle(size):
    #loop this code 3 times
        winston.forward(size)
        winston.left(120)
```

What are loops?

Loops allow code to be executed multiple times

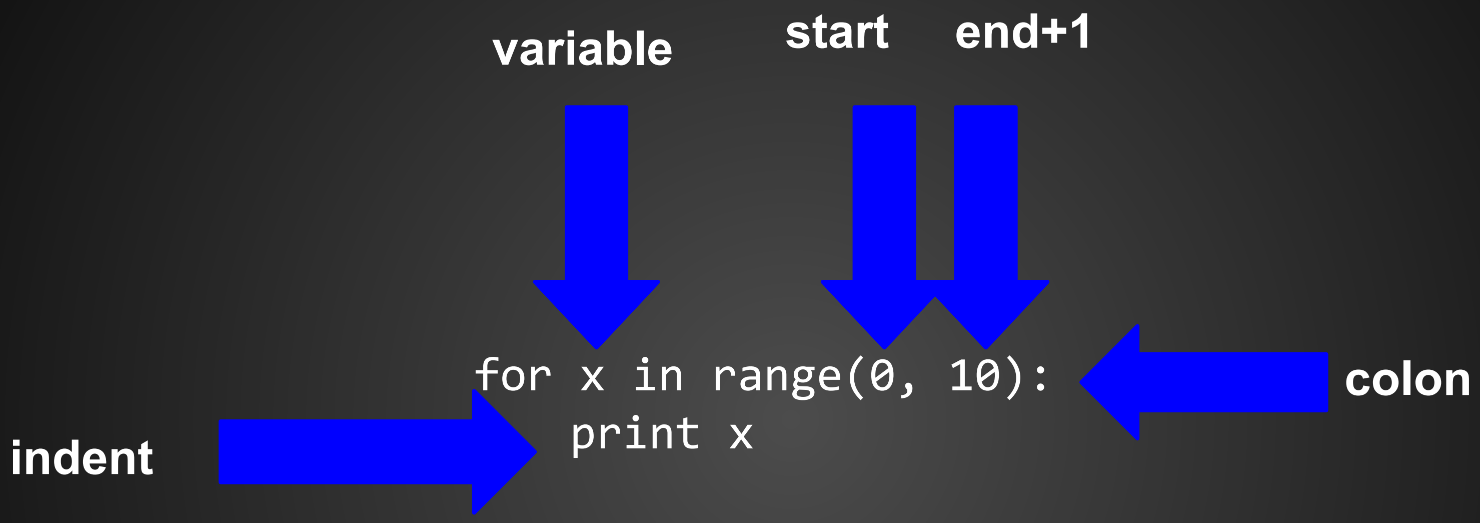
```
def dodecagon(size):
    winston.forward(size)
    winston.left(30)
    winston.forward(size)
    winston.left(30)
    winston.forward(size)
    winston.left(30)
    winston.forward(size)
    winston.left(30)
    winston.forward(size)
    winston.left(30)
    winston.forward(size)
    winston.left(30)
    winston.forward(size)
    winston.left(30)
```

```
def dodecagon(size):
    #loop this code 12 times
        winston.forward(size)
        winston.left(30)
```

Types of loops

- For loops
    - Generally used for a known number of repetitions
- While loops
    - Generally used when number of repetitions is unknown/not known at the start

For loops



```
for x in range(0, 10):
    print x
```

REMEMBER TO INDENT

While loops

condition

```
while x < 7:
    print x
    x = x + 1
```

colon

indent

REMEMBER TO INDENT

Why is this useful?

- Do things with less code

```
print 1
print 2
print 3
print 4
print 5
print 6
print 7
print 8
print 9
print 10
...
print 100
```

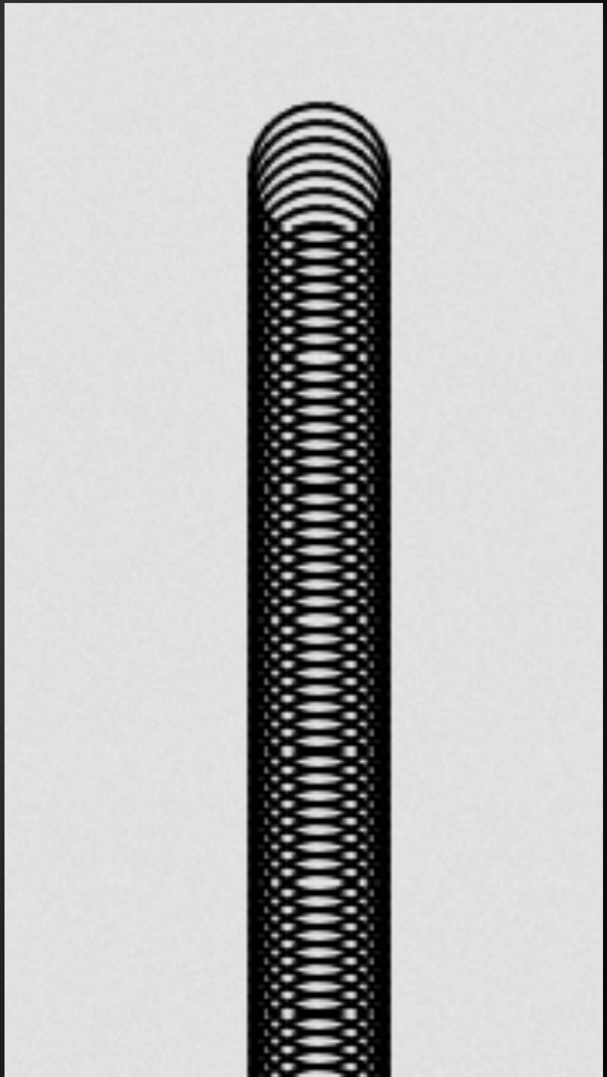
```
for n in range(1,101):
    print n
```

Why is this useful?

- That's it

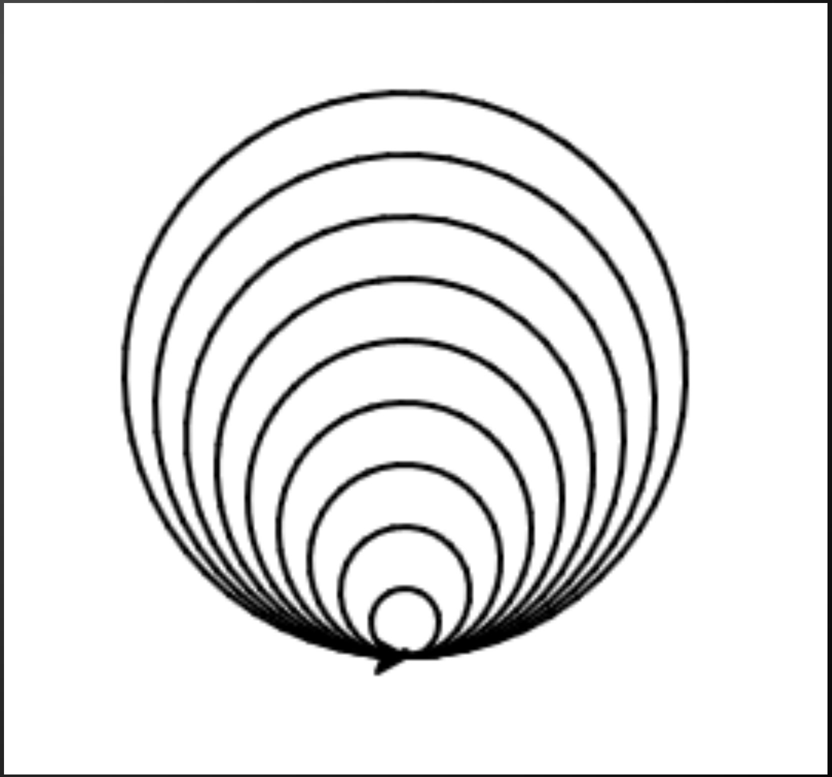
Drawing things

```
def tube():
   for i in range(1,100):
      winston.circle(20)
      winston.penup()
      winston.right(90)
      winston.forward(5)
      winston.left(90)
      winston.pendown()
```



Drawing things

```
def foo():
   for i in range(1,100,10):
      winston.circle(i)
```



Smarter Chatbot

```
answer = ''
while answer != 'please?':
   answer = raw_input("Where's your manners?")
print 'Thank you!'
```

Questions?

Another Pop Quiz!