CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson
Fall 2017
Swarthmore College
Outline Sept 22:

- Nesting (telephone.py)
- Lab 1 and research examples
- String formatting
- Next time:
  - range(..) with multiple parameters
  - ascii (ord and chr)

Notes

- Lab 2 due Saturday night
- Office hours today 3-5pm (room 260)
- Ninja session tonight 7-9pm (room 256)
Lab 1 examples
(not posted online)
Research examples
Computational biology examples

```python
# loop over the haplotypes
keep_haps = []
for i in range(N):
    hap = f.readline()
    if anc_der[i] == 'a': # if the haplotype is ancestral
        keep_haps.append(hap)

# loop through the files and add haplotypes
total_haps = []
for filename in file_lst:
    positions, keep_haps = parse_mbs_file(filename)
    total_haps = total_haps + keep_haps
```
Notecard feedback
Notecard feedback

1) **Something you understand well**: conditionals, types, input, print, math operations, variables

2) **Most confusing**: for loops, range, indexing, accumulators, random, importing libraries

3) **Change about the course**: more examples, go over labs, more optional practice, more time, smaller class
String formatting
String formatting

- **%d** – decimal (same behavior as %i it turns out!) use for integers, in newer python formatting %i is sometimes not supported

- **%f** – float, use %8.2f (for example) to make each float be 8 characters total (spaces used as padding on the left), rounded to 2 decimal places

- **%s** – string (same notation applies, but %8.2s will pad with spaces as well as truncate to 2 characters)
String formatting examples

```python
>>> string = "code"
>>> z = 35

>>> print("My %s is %d lines long." % (string, z))
My code is 35 lines long.

>>> p = 3.141592

>>> print("Pi is about %f" % p)
Pi is about 3.141592

>>> print("Pi is about %.2f" % p)
Pi is about 3.14

>>> print("Pi is about %.4f" % p)
Pi is about 3.1416

>>> num_lst = [234.575742, 14.46574, 8.231235, 19.4239823928398]

>>> for num in num_lst:
...     print("lining up numbers: %7.3f" % num)
...
lining up numbers: 234.576
lining up numbers: 14.466
lining up numbers: 8.231
lining up numbers: 19.424
```