Problem Solving

Environment, Mapping, Etc.
hw08 due 4/17
exam4 4/18–4/20
hw09 due 4/21 (not posted yet)
The Python Environment
Python exists as a component of a complex operating system.

I assume that you are using a Unix-style command line (Linux, macOS, or Bash for Windows).
The command line or shell is the program that opens first.

- (This is where you would type `jupyter-notebook`.)
- The prompt is normally `$.`
$ ls  # show the folder contents
$ cat file  # show contents of file
$ mv file newfile  # rename a file
$ cp file newfile  # copy a file
$ which python  # tells you where Python is
$ python  # or ipython

(You don’t need to type the $.)
Unfortunately, the same functionality is rather scattered in Python: check `sys`, `os`, and `shutil` for details.
The prompt reminds you where you are. This is the Python prompt:

```python
>>> (You don’t need to type the »>.)
```
>>> import seaborn
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named seaborn

What if something you need isn’t installed yet?
$ pip install seaborn
Normally, this is all you need to do to install a package.
Mapping
items = [1,2,3,4,5]
cubed = []
for x in items:
    cubed.append(x**3)
Sometimes a loop is a little heavy when you just need to transform a list of data by a function.

- **map** allows you to map values to results.
- **map** accepts 1) a function; 2) a list of values to pass to the function.
- Like range, **map** needs to be cast as a list.
def cube( x ):
    return x**3
map( cube,[ 1,2,3,4,5 ] )
Multiple arguments can be handled gracefully:

```javascript
map( pow, [ 1, 2, 3, 4, 5 ], [ 1, 2, 3, 4, 5 ] )
```
- map is always available.
- map is probably slightly faster than a for loop.
- map can always be replaced with a for loop (so like `itertools.product`, you can use it but don’t have to know it).
Filtering

```python
data = []
for i in [ -1, -0.5, 0, 0.5, 1 ]:
    if i < 0:
        data.append( i )
```
- **filter** allows you to extract each element of a sequence for which the function returns **True**.
- **Corollary:** have to have a Boolean function for your data.
def ltz(x):
    return x<0

list( filter( ltz, [ -1,-0.5,0,0.5,1 ] ) )
def isalpha(x):
    return x.isalpha()

list(filter(isalpha, [ '1','m','Pat','2a' ] ))

This can be a bit awkward sometimes though. Use what fits your problem.
If you are interested in delving deeper into Python, here are some topics to investigate:

- List comprehensions
- `class` (objects)
- `pandas` for data analysis
- `lambda` (anonymous) functions

Take a look at CSE Python workshops (cse.illinois.edu).
Exam Preparation
What to expect:

- Coding (brute force)
- Coding (optimization)
- Coding or MC on try/except
- A few MC on newton or fmin, timing/profiling, and exceptions
Your friend Vanessa can’t remember her Facebook password and wants your help figuring it out. She remembers the password is exactly 8 characters long. She also remembers that her username is either "vanessa" or "VanessaC" or "Vanessa95". Assume someone else has already written a function \texttt{login} that takes a two string arguments representing a username and password combination. \texttt{login} returns \texttt{True} if the input username and password are valid credentials for Facebook and \texttt{False} otherwise. Compose a function \texttt{guess_password()} which performs a brute force search and return the correct username and password for Vanessa’s account as a tuple of two strings.

```python
def guess_password():
    alphabet="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
    alphabet+="0123456789!@#$%^&*()-_=+,<.>/?~‘"
```
Write a Python program to simulate a population of rabbits for 50 years. Your simulation should update annually ($\Delta t = 1 \text{ a}$). The initial population of rabbits is 15. Each year, 5% of the population of the previous year dies off, and each year exactly 5 new rabbits are born. In your simulation, it should be impossible for “partial rabbits” to exist. For example, the population should never be 13.7—we’re not interested in 70% of a rabbit. You should always round the population down to a whole number.
Next steps
Next steps

- hw08 due 4/17
- exam4 4/18–4/20
- hw09 due 4/21 (not posted yet)