

General Notes

Containers

How to make tuples and lists and dicts and sets:

```
Length 0
list:  []
tuple: ()
dict:  {}
set:   set()

Length 1
      [10]  [10,]
      10,   (10,)
      {'raymond': 'red'}
      {10}
```

Lists versus iterators

List ~ inventory ~ uses much mem

Iterator ~ jit manufacturing ~ memory friendly -> fast

list	iterator
range()	xrange()
zip()	izip()
--	enumerate()
--	reversed()
sorted()	--
d.items()	d.iteritems()

Convert to other types

```
tuple(s)
set(s)
dict.fromkeys(s)
```

Comprehension

LIST COMP:

```
[i**2 for i in range(10)]
```

SET COMP:

```
{i**2 for i in range(-5, 10)}
```

DICT COMP:

```
{i**2 : i for i in range(10)}
```

GEN EXP:

```
sum(i**2 for i in range(10))
```

With-Statement

How to use the with-statement for file closing and lock releasing:

```
# The old way that STILL works
f = open('notes/stocks.txt')
try:
    data = f.read()
finally:
    f.close()

# The new way
with open('notes/stocks.txt') as f:
    data = f.read()

# The old way that STILL works
lock = threading.Lock()
lock.acquire()
try:
    crit1()
    crit2()
finally:
    lock.release()

# The new way
with lock:
    crit1()
    crit2()
```

Coding style

LBYL -- look before you leap:

```
def average(seq):
    n = len(seq)
    if n == 0:
        return 1.0
    else:
        return sum(seq) / n
```

EAFP -- easier to ask forgiveness than permission:

```
def average(seq):
    n = len(seq)
    try:
        return sum(seq) / n
    except ZeroDivisionError:
        return 1.0
```

Regex notation

raymond	Does an exact match	
profit loss	Match either profit or loss	
photo_...jpg	Dots match any character	photo_32.jpg
photo[a-g].jpg	Brackets match a character group	photoa.jpg photob.jpg
[aeiou]	Matches any single vowel	
[^aeiou]	Matches any single non-vowel	
\d	[0-9] any digit	
\D	[^0-9] any non-digit	
\w	[A-Za-z0-9_] alphanumeric text with an underscore	
\W	non-alphanumeric without an underscore	
\s	space characters	
\S	non-space characters	
\b	word boundaries	
a?	Match zero or one	a{0,1}
a+	Match one or more	a{1,}
a*	Match zero or more	a{0,}
a{m,n}	Match m to n	
match	Must match right at the beginning	--> match object
search	May match anywhere	--> match object
findall	Finds them all	--> list of matches

Recommendations:

1. Always start with `findall()`
2. Start with a small regex that matches, then grow it.
3. Always use raws strings for regexes

Regular expression groups:

```
mo = re.search(r'(\d+)/(\d+)/(\d+)', data)
mo.group()      entire match
mo.group(0)     entire match
mo.group(1)     month
mo.group(2)     day
mo.group(3)     year

month, day, year = re.search(r'(\d+)/(\d+)/(\d+)', s).groups()
```

Logging

To setup a logger:

```
import logging
logging.basicConfig(filename='example.log', level=logging.DEBUG)
```

To make log entries:

```
logging.debug(msg)
logging.info(msg)
```

```
logging.error(msg)
logging.critical(msg)
logging.warn(msg)
```

Auto-documentation

How to generate documentation automatically:

```
$ python -m pydoc -w vcard
```

Directories

Popular commands in the OS module:

```
os.getcwd()          # show the current working directory
os.listdir('.')       # list the files in a given directory
os.chdir(somedir)     # change to another directory
```

How to use subprocess

```
>>> import subprocess
>>> s = subprocess.getoutput('netstat -a')
>>> [line.split()[4] for line in s.splitlines() if 'tcp4' in line]
```

REST APIs

QR codes:

```
https://developers.google.com/chart/infographics/docs/qr_codes
http://chart.apis.google.com/chart?cht=qr&chs=300x300&chl=Hello%20World --> hello.png
```

Twitter:

```
https://dev.twitter.com/docs/api/1/get/search
http://search.twitter.com/search.json?q=@raymondh&rpp=10
```

Debugger

To run the debugger from the command-line (not the interactive prompt):

```
$ python -m pdb tweet_bug.py
```

The most common debugger commands:

```
c    - continue until an exception or normal finish
n    - next (run to the next line)
l    - list the source and show where you are
b    - set a breakpoint
p    - print a variable
pp   - pretty print a variable
q    - quit
```