Practical Bioinformatics

Mark Voorhies

4/19/2011

Review

Topic	Learning Python	Dive Into Python
Types (string, int, float, list)	2	3
Assignment (=, +=)	2	3
Slicing and indexing	4	3.2
References and copying	2.6	3.2
if statement	3	2.5
for loops	3	6
Defining functions	4	2
import and reload	5	2

```
def mean(x):
    s = 0.0
    for i in x:
        s += i
    return s/len(x)
```

Gotchas

- Case matters (True, not TRUE or true)

Gotchas

- Case matters (True, not TRUE or true)
- Strings are quoted, names of things are not.
  - mystring = "mystring"

Gotchas

- Case matters (True, not TRUE or true)
- Strings are quoted, names of things are not.
  - mystring = "mystring"
- Square brackets, [and], are for lists and indexing:
  - ratios = [.25, .5, 1.0, 2.0, 4.0]
  - ratios[0]
  - ratios[:3]

Gotchas

- Statements that precede code blocks (if, def, for, while, ...) end with a colon.

```
def mean(x):
    s = 0.0
    for i in x:
        s += i
    return s/len(x)
```

Gotchas

- Statements that precede code blocks (if, def, for, while, ...) end with a colon.

```
def mean(x):
    s = 0.0
    for i in x:
        s += i
    return s/len(x)
```

- In interactive sessions (, IDLE), blocks need to be followed by a blank line.

Setting IDLE's working directory

OS X

- Open a terminal
- cd path/to/working/directory
- python -m idlelib.idle

Windows

- Use (file) explorer to find the path to your working directory and copy it to the clipboard.
- Right-click your IDLE menu item and choose “properties”.
- Paste your working directory into the “Start in” field, making sure that it is quoted.

Loading and re-loading your functions

```
# Us rt t rst t y u a a u
#   (A us rt u t t a s
#    succ ss u y)
import my_module

my_module.my_function(42)
```

Mean

```
def mean(x):
    s = 0.0
    for i in x:
        s += i
    return s/len(x)
```

Mean

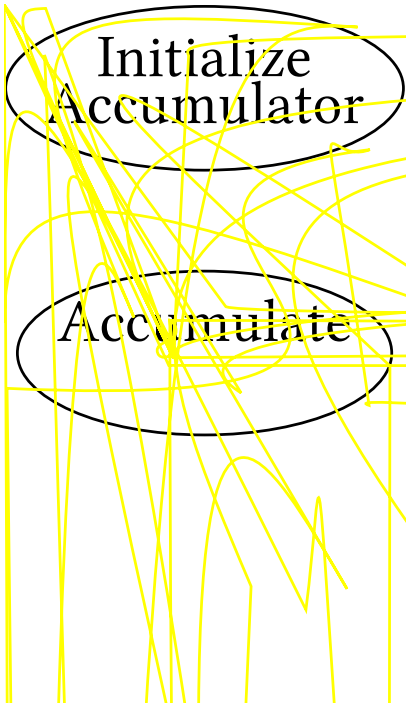
```
def mean(x):
    s = 0.0
    for i in x:
        s += i
    return s/len(x)

def mean(x):
    return sum(x)/float(len(x))
```

Mean

```
def mean(x):
    s = 0.0
    for i in x:
        s += i
    return s/len(x)

def mean(x):
    return sum(x)/float(len(x))
```



Standard Deviation

$$\sigma_x = \sqrt{\frac{\sum_i^N (x_i - \bar{x})^2}{N-1}}$$

Standard Deviation

$$\sigma_x = \sqrt{\frac{\sum_i^N (x_i - \bar{x})^2}{N-1}}$$

```
def stdev(x):
    m = mean(x)
    s = 0.0
    for i in x:
        s += (i - m)**2
    from math import sqrt
    return sqrt(s/(len(x) - 1))
```

Pearson's Correlation Coe cient

```
def pearson(x,y):
    mx = mean(x)
    my = mean(y)
    sxy = 0.0
    ssx = 0.0
    ssy = 0.0
    for i in range(len(x)):
        dx = x[i] - mx
        dy = y[i] - my
        sxy += dx*dy
        ssx += dx**2
        ssy += dy**2
    from math import sqrt
    return sxy/sqrt(ssx*ssy)
```

$$r(x,y) = \frac{\sum_i (x_i - x)(y_i - y)}{\sqrt{\sum_i (x_i - x)^2}\sqrt{\sum_i (y_i - y)^2}}$$

Subject, verb that noun!

return_value = object.function(parameter, ...)
"Object, do *u ct* to *ara t r*"

- file = open("myfile.txt")
- file.read()
- file.readlines()
- for line in file:
- string.split() and string.join()
- file.write()

Binary files are like genomic DNA

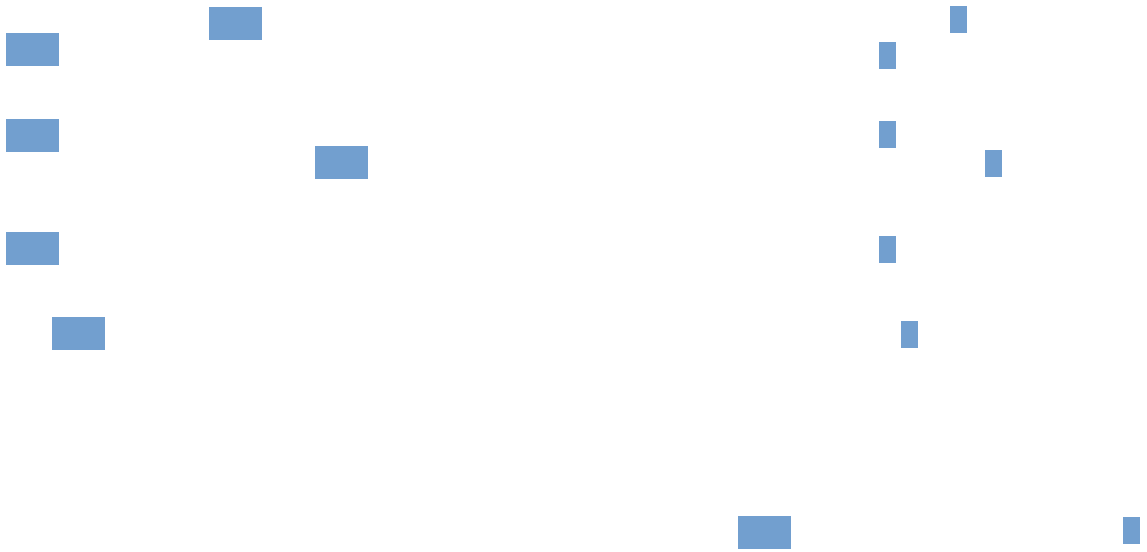
hexdump -C computers.png

```
fp = open("computers.png")
fp.read(50)
fp.close()
```

Text files are like ORFs

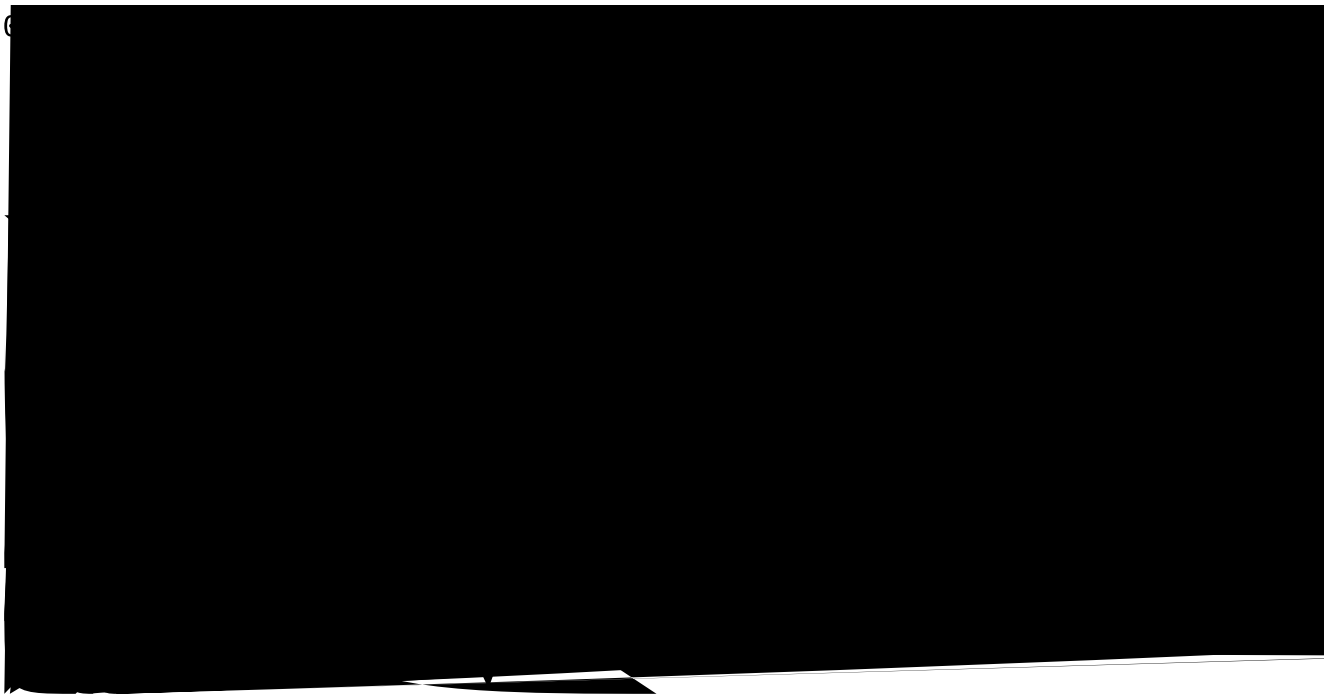
hexdump -C 3_4_2010.txt

```
00000000  4d 65 6
```



OS X *sometimes* uses CR newlines

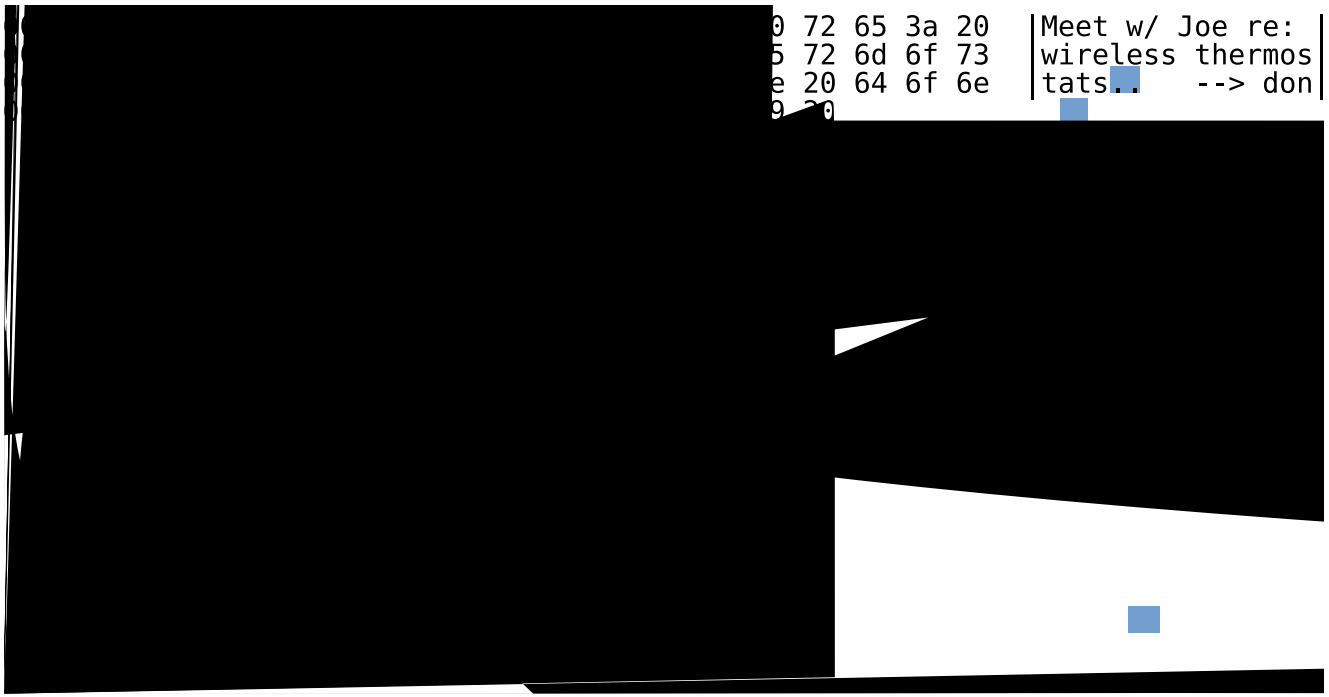
hexdump -C macfile.txt



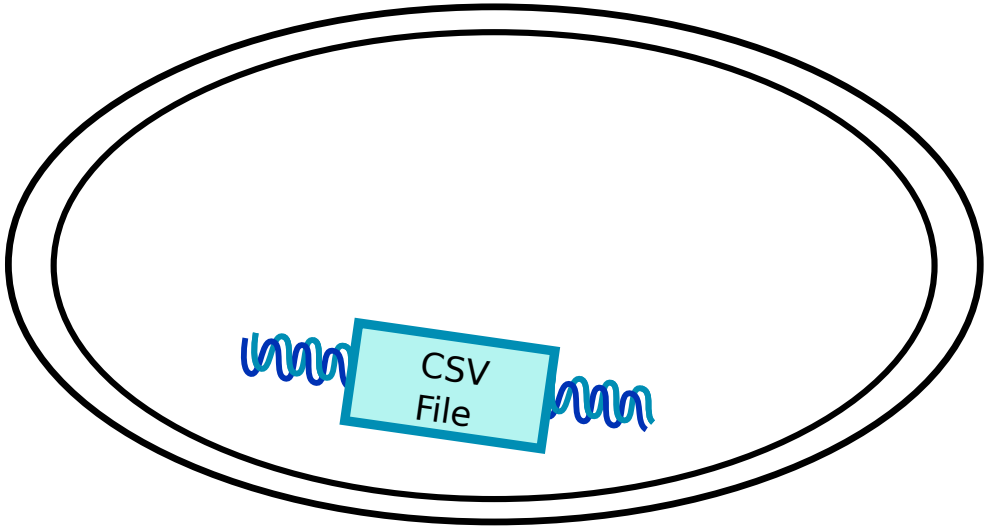
tr '\r' '\n' < macfile.txt > unixfile.txt

Windows uses CRLF newlines

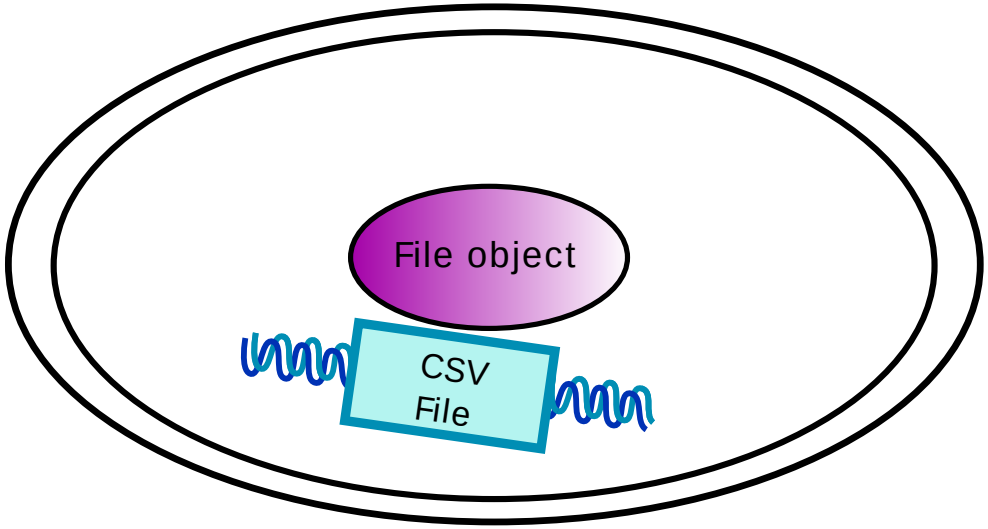
hexdump -C dosfile.txt



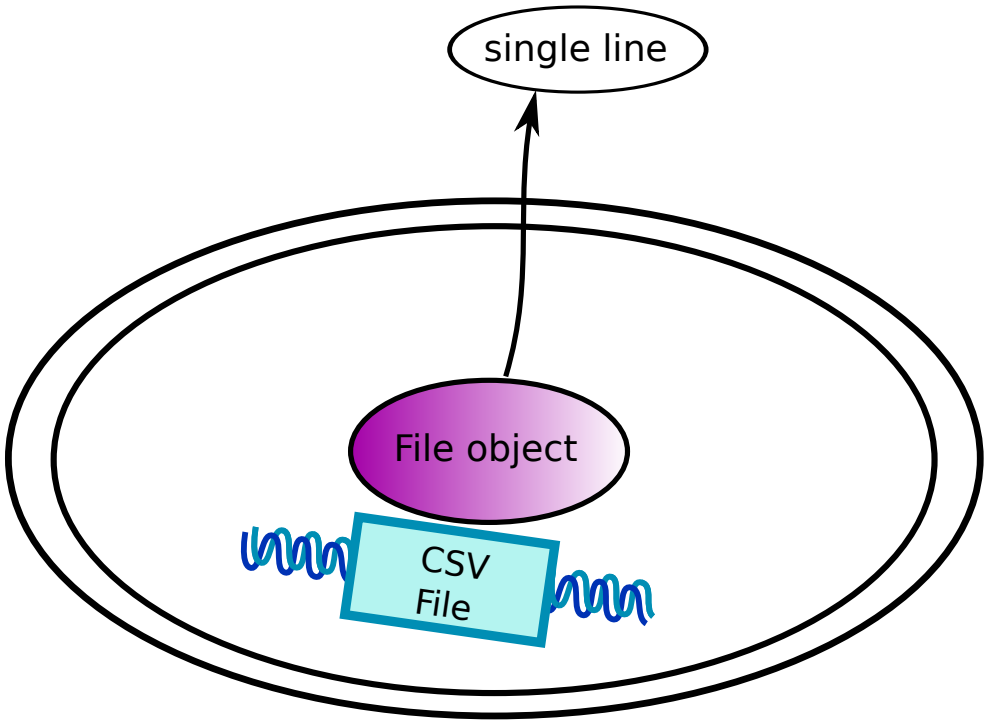
supp2data.csv



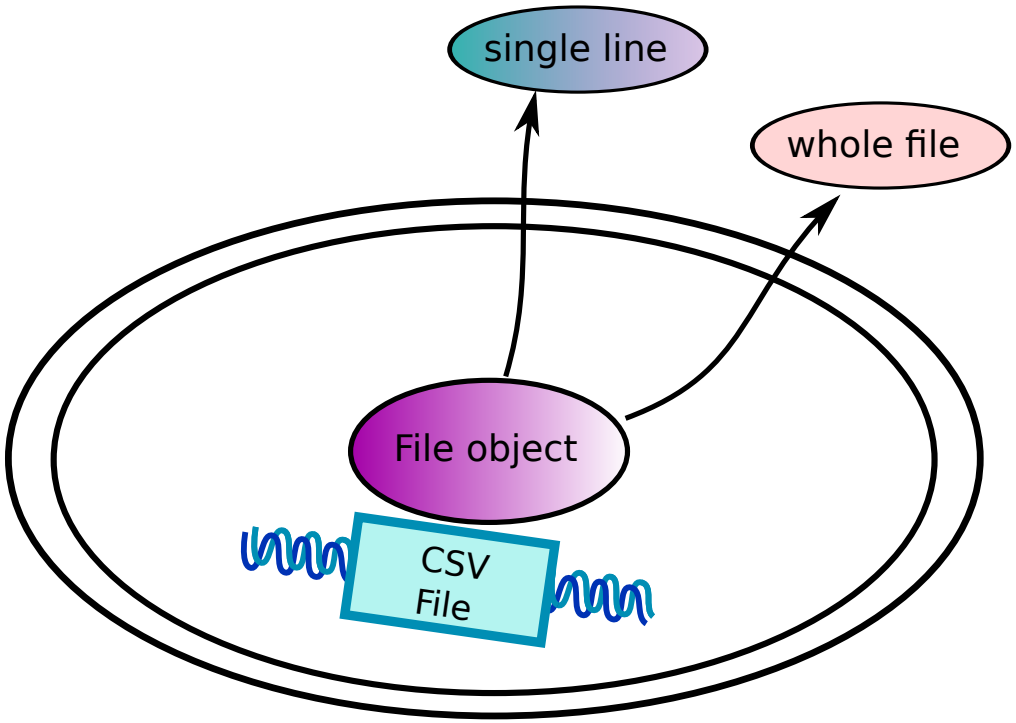
open(\supp2data.csv")



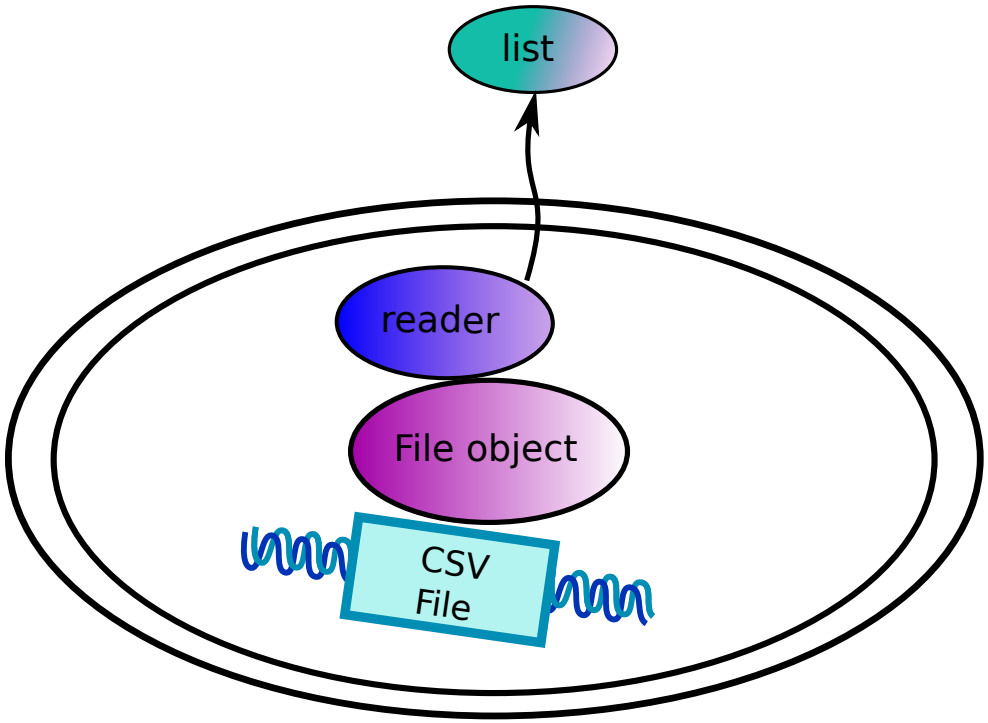
open(\supp2data.csv").next()



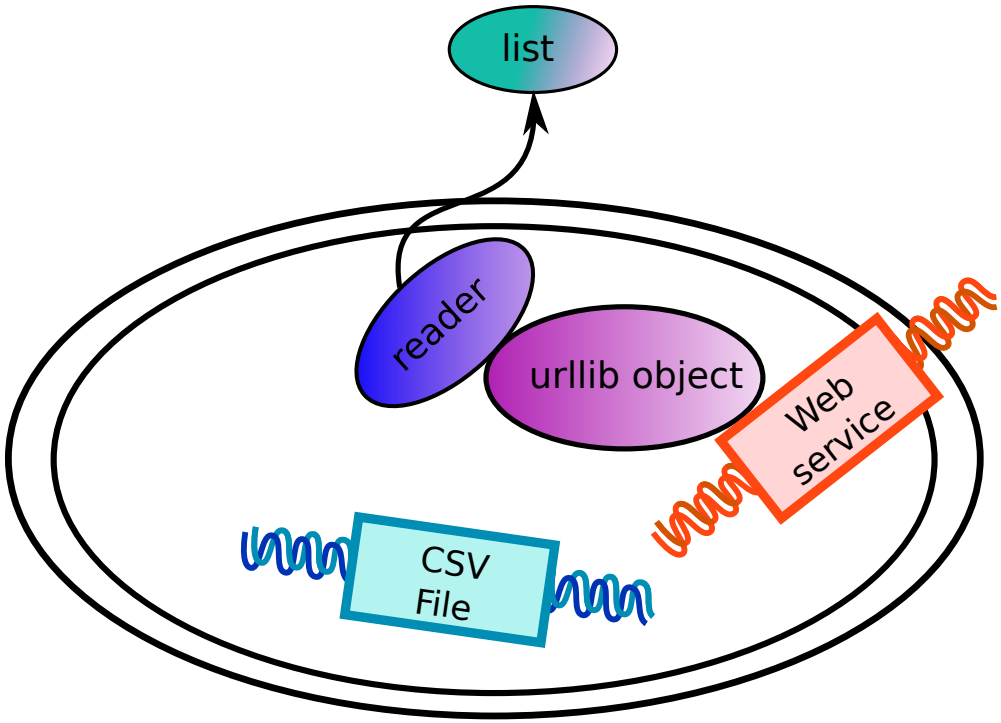
open(\supp2data.csv").read()



csv.reader(open(\supp2data.csv")).next()



csv.reader(urlopen(\http://example.com/csv")).next()



Homework

1. Read PNAS 95:14863
2. Experiment with reading the first few bytes of different files on your computer. Can you distinguish binary files from text files?
3. Create a simple data table in your favorite spreadsheet program and save it in a text format (e.g., save as CSV or tab-delimited text from Excel[1]). Practice reading the data from Python.

[1] Note for Mac users: Excel will offer you Macintosh and DOS/Windows text formats. Choose *DOS/Windows*; otherwise, Python will think that the entire file is a single line.