Practical Bioinformatics

Mark Voorhies

5/21/2013

Gotchas

- Enclosing symbols (”” ,” ,(),[],...) need to be matched

Gotchas

- Enclosing symbols ("",'',(),[],...) need to be matched
- Pay attention to IPython's syntax highlighting to match your quotes.

Gotchas

- Enclosing symbols (""",",(),[],...) need to be matched
- Pay attention to IPython's syntax highlighting to match your quotes.
- IPython highlights parens for the current cursor position in green

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.

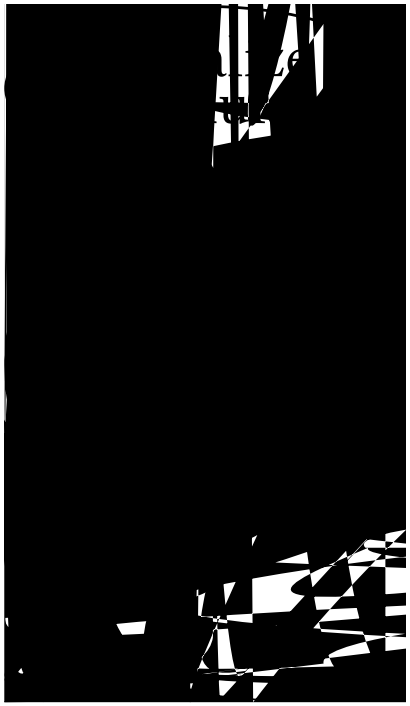
```
def mean(x):
    s = 0.0
    for i in x:
        s += i
    return s/len(x)
```

- Loop variables retain their state after the loop is finished (so if you want to reuse the variable, you need to reinitialize it).

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))
```



Loading and re-loading your functions

```
# Use import the first time you load a module
#   (And keep using import until it loads
#    successfully)
import my_module

my_module.my_function(42)

# Once a module has been loaded, use reload to
#   force python to read your new code
reload(my_module)
```

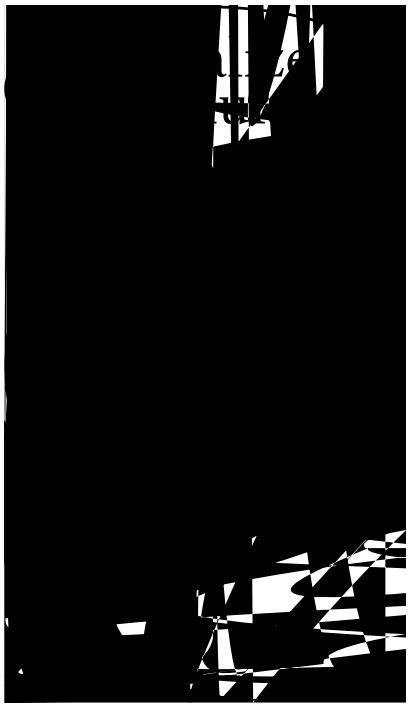
Standard Deviation

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

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 Coefficient

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

Pearson's Correlation Coefficient

```
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)
```

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

[T]he relational graphic { in its barest form, the scatterplot and its variants { is the greatest of all graphical designs. It links at least two variables, encouraging and even imploring the viewer to assess the possible causal relationship between the plotted variables.

–Edward Tufte

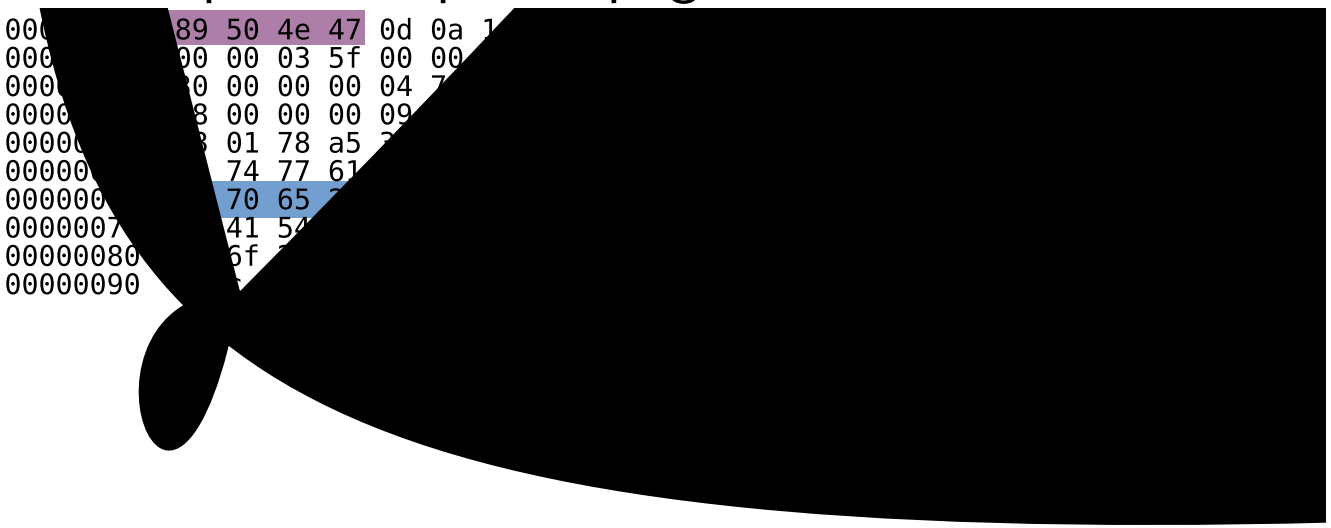
Subject, verb that noun!

return_value = object.function(parameter, ...)
"Object, do *function* to *parameter*"

- 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

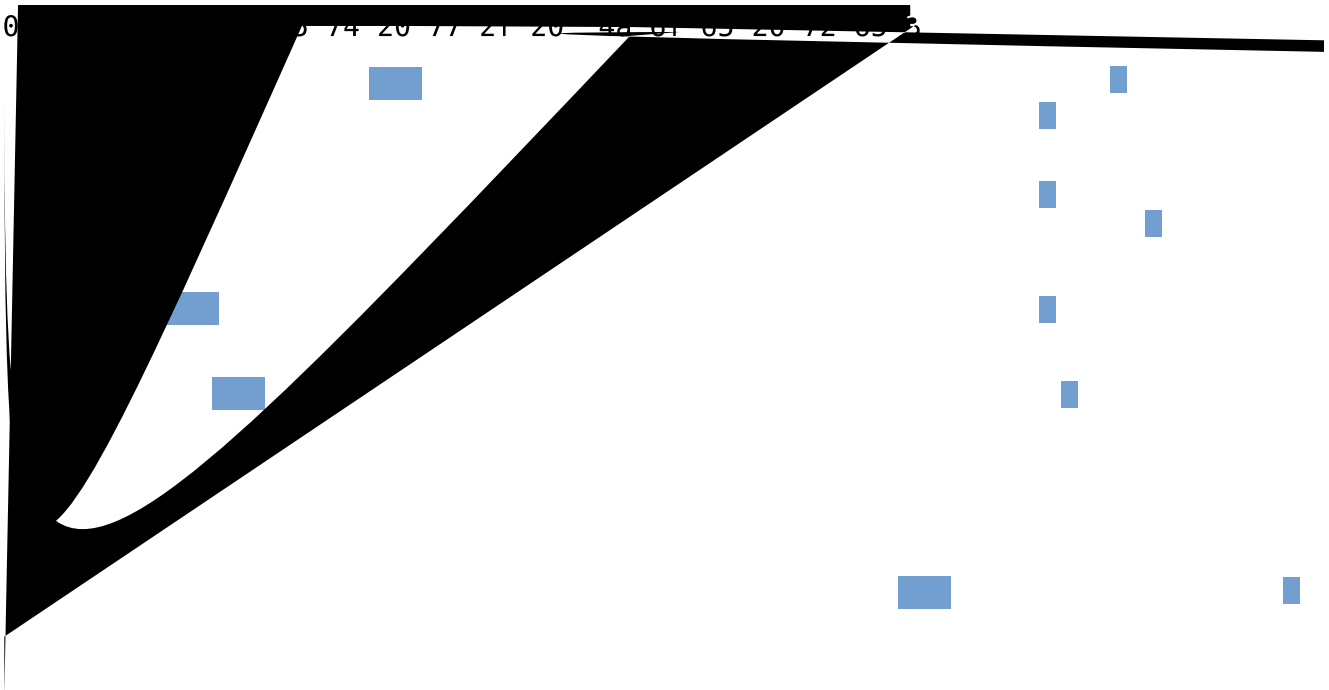
hexdump -C computers.png



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

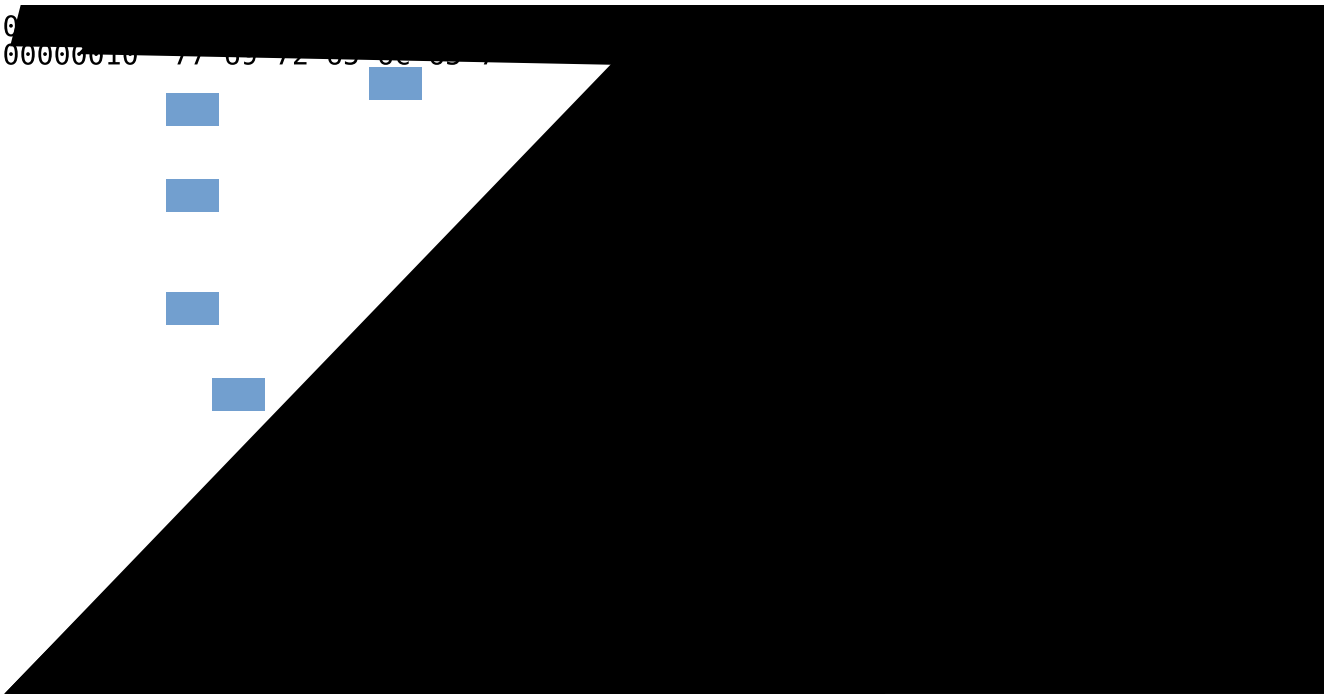
Text files are like ORFs

hexdump -C 3_4_2010.txt



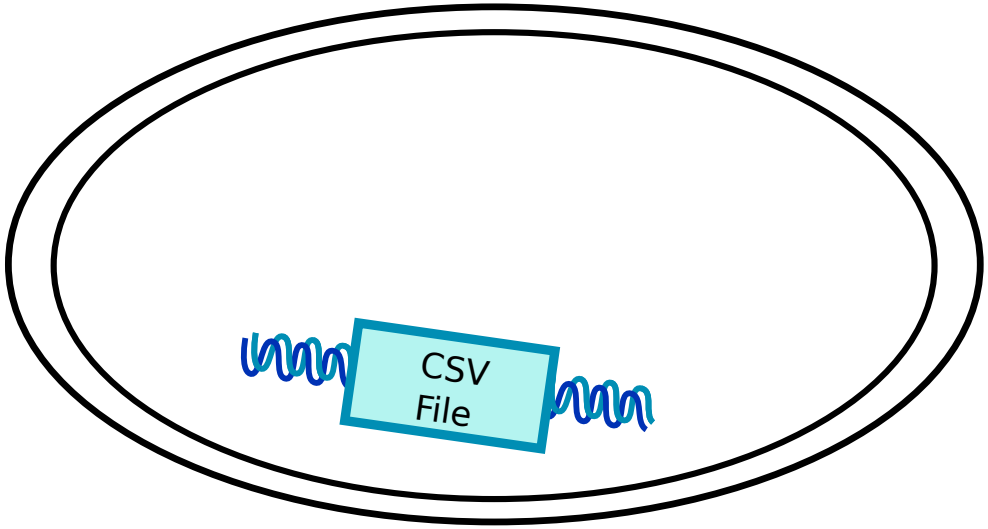
OS X *sometimes* uses CR newlines

hexdump -C macfile.txt

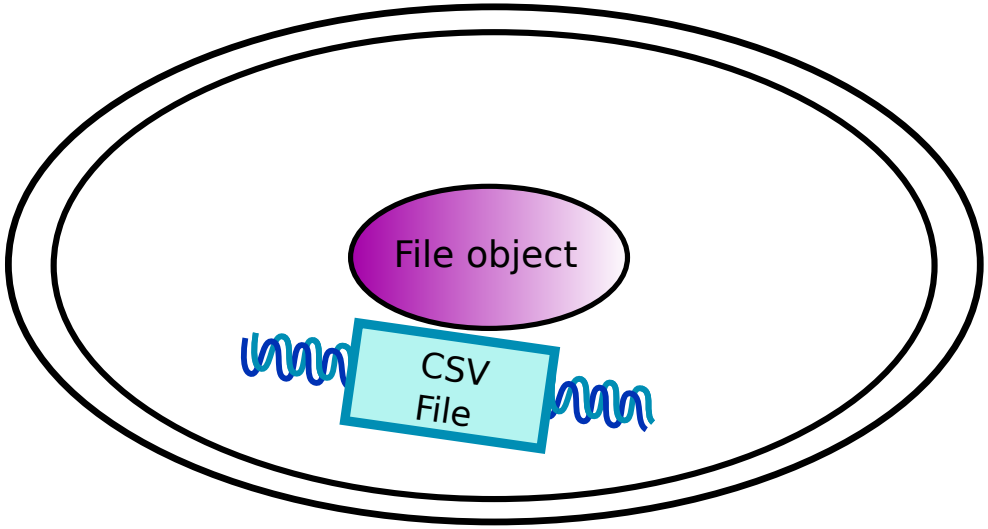


tr '\r' '\n' < macfile.txt > unixfile.txt

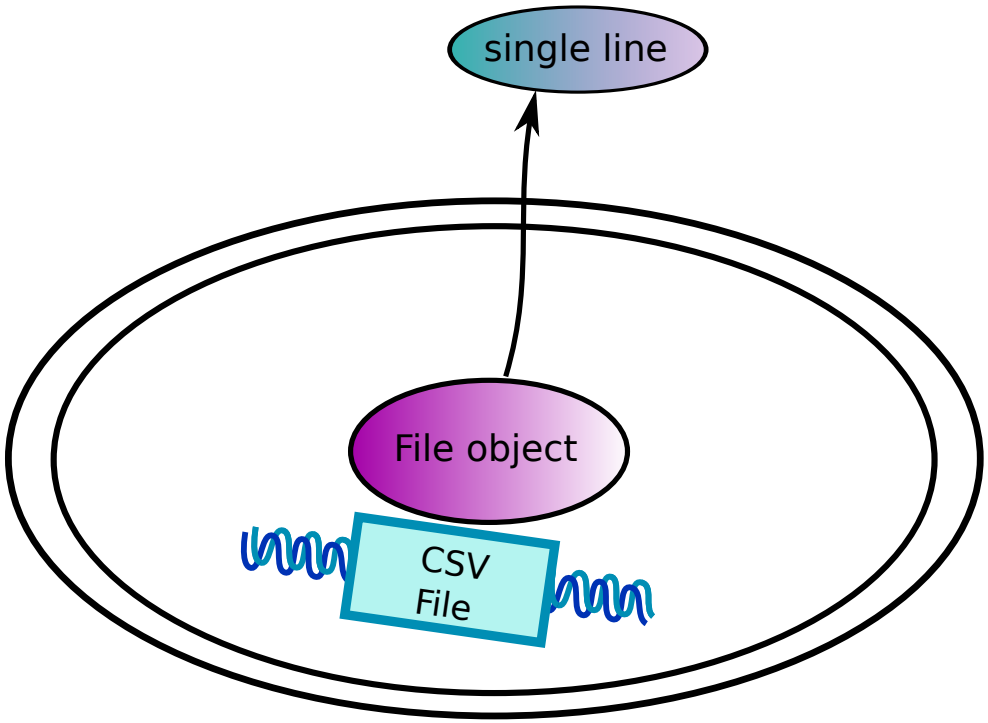
supp2data.csv



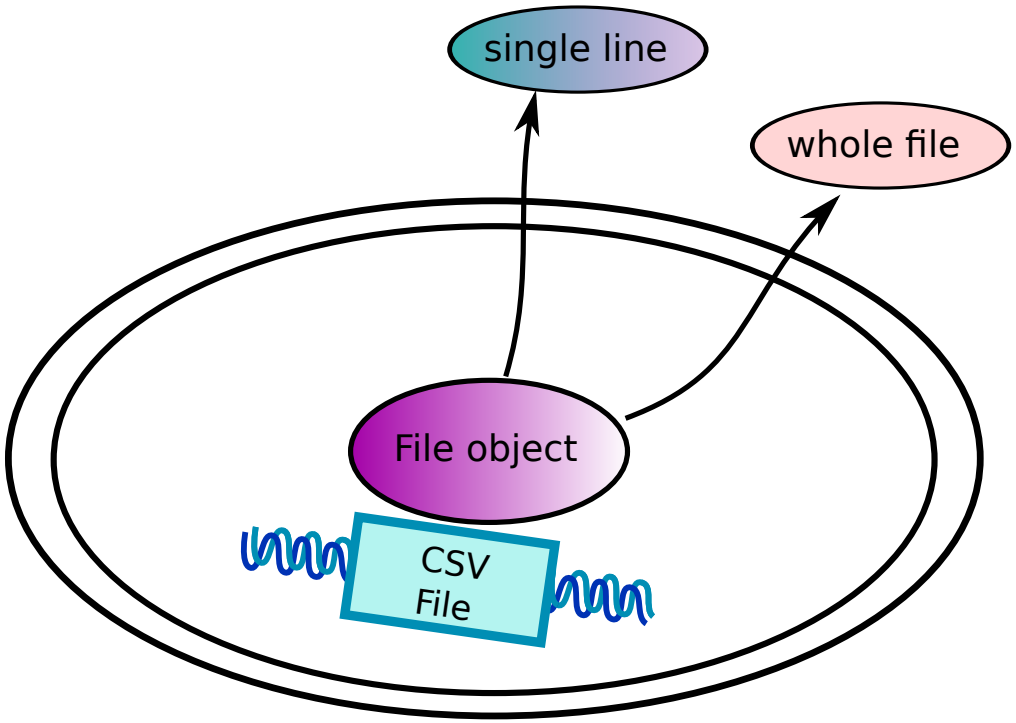
open("supp2data.csv")



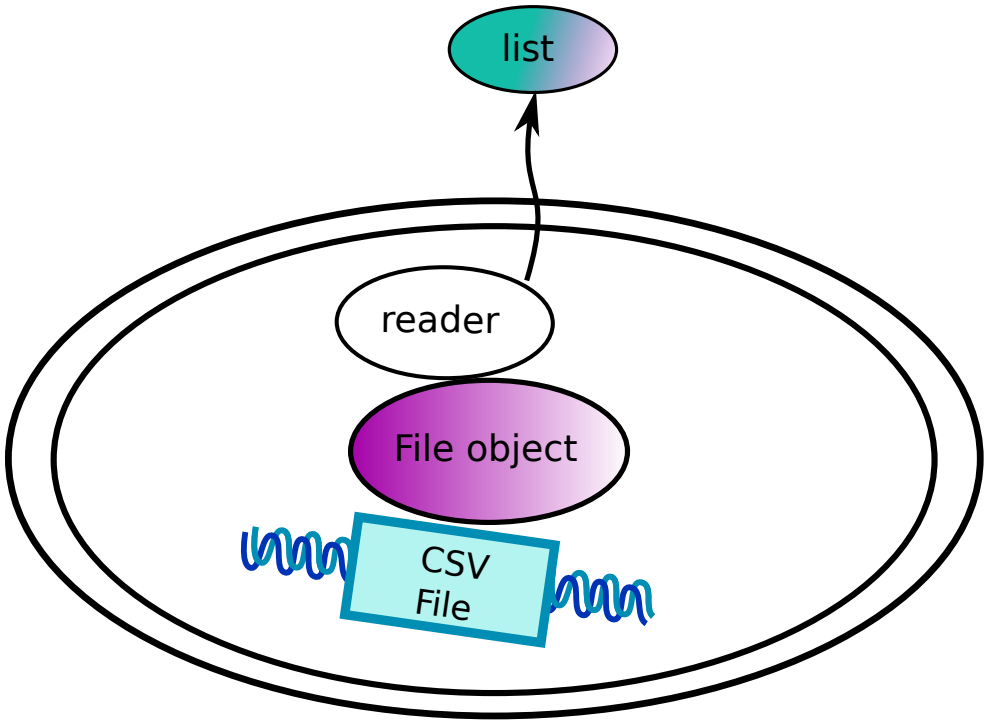
open("supp2data.csv").next()



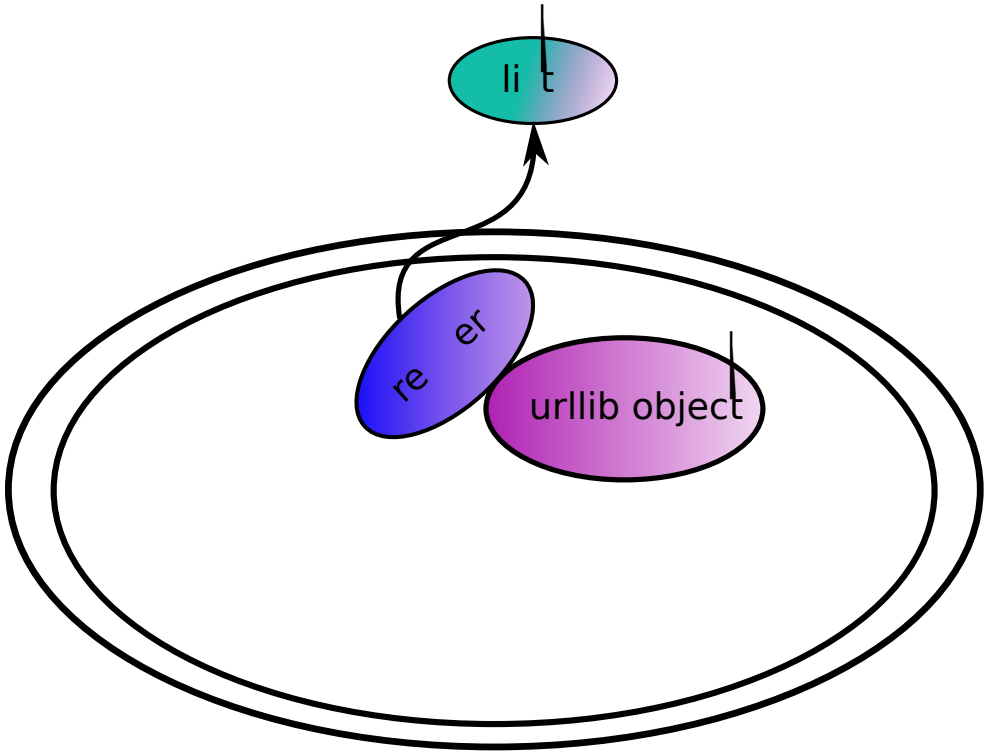
open("supp2data.csv").read()



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



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



Homework

1. Download and install Cluster3 and JavaTreeView
2. Try 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.
4. Write a function to disect supp2data.cdt into three lists of strings (gene names, gene annotations, and experimental conditions) and one matrix (list of lists) of log ratio values (as floats, using *None* or *0.* to represent missing values).
5. If you are familiar with Python classes, write a CDT class based on the parse in the previous exercise. Provide methods for looking up annotations and log ratios by gene name.

[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.