LING 408/508: Computational Techniques for Linguists

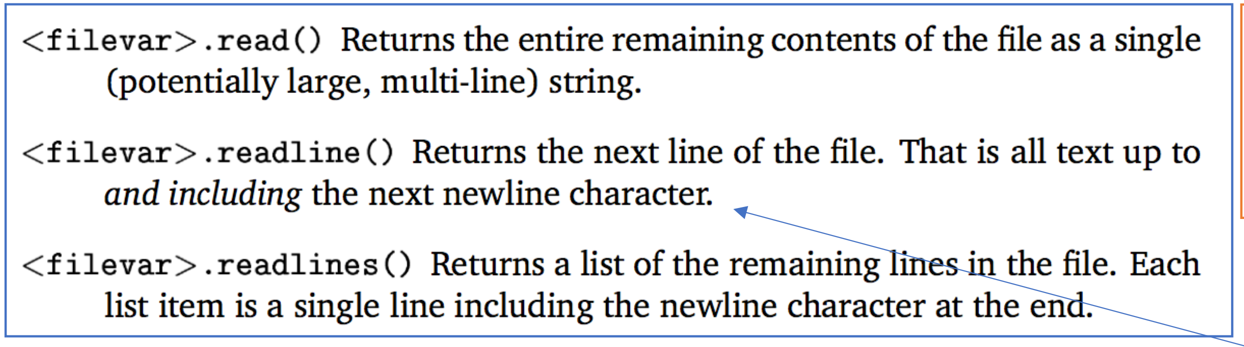
Lecture 28

Today's Topics

- Did people get nltk and nltk_data installed?
  - instructions are on last time's slides
- Reminder: term project!
  - be working on it, or
  - propose something quickly if you haven't thought of it yet!

Recap: Python and Files

- Like all other programming languages, uses a file handle, called **file variable**: open()
- `infile = open("file.txt","r")`   `outfile = open("results.txt,"w")`



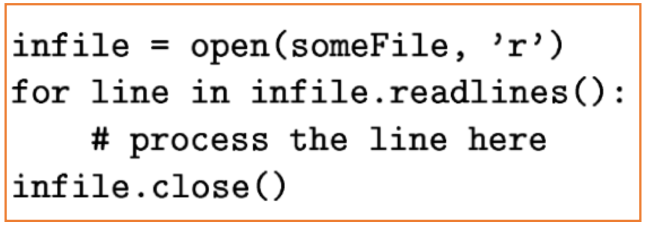
`<filevar>.read()` Returns the entire remaining contents of the file as a single (potentially large, multi-line) string.

`<filevar>.readline()` Returns the next line of the file. That is all text up to *and including* the next newline character.

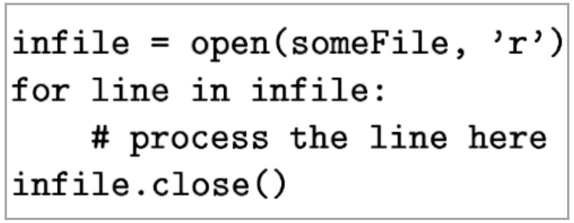
`<filevar>.readlines()` Returns a list of the remaining lines in the file. Each list item is a single line including the newline character at the end.

f = open(fname)
f = open(fname, encoding="utf-8")
f = open(fname, encoding="latin-1")
f = open(fname, encoding="ascii")

Removing the newline:
`.strip()`
`.rstrip()`
`.lstrip()`

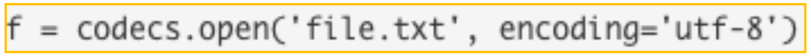


```
infile = open(someFile, 'r')
for line in infile.readlines():
    # process the line here
infile.close()
```



```
infile = open(someFile, 'r')
for line in infile:
    # process the line here
infile.close()
```

Python 2.7



```
f = codecs.open('file.txt', encoding='utf-8')
```

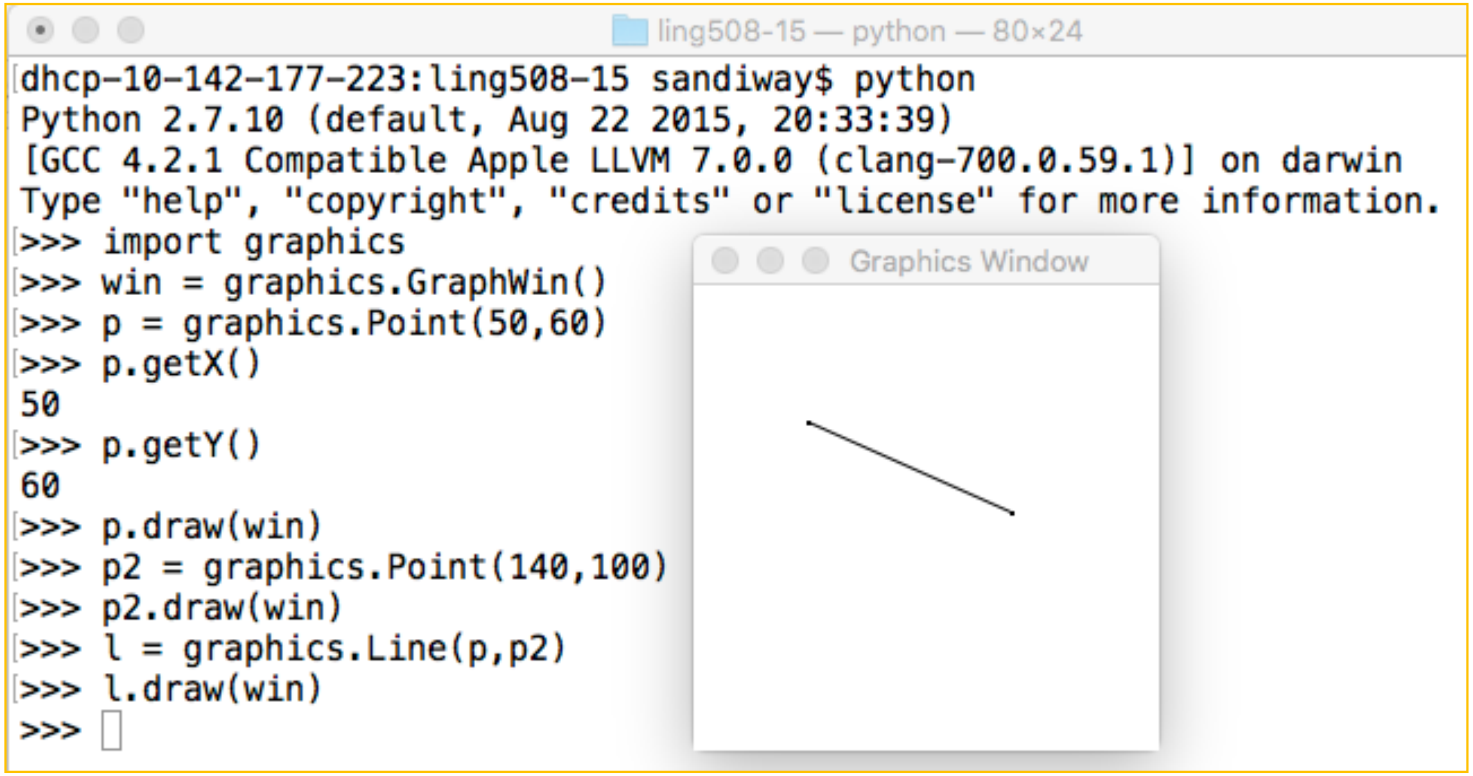
Graphics

- Graphics is not usually part of a programming language

  However, html5 has canvas and svg built-in
- Traditionally:
  - Graphics ➔ Libraries and Toolkits
- Example:
  - Tk (designed by John Ousterhout as an extension to Tcl)
  - Tcl (Tool Command Language, also designed by John Ousterhout)
- Tk was originally built on top of the (portable) X11 client-server window system for Unix but has also been ported to native, e.g. OSX, platform graphics
  - NOT FOR the client-server model, but for graphics from the Python interpreter

graphics.py

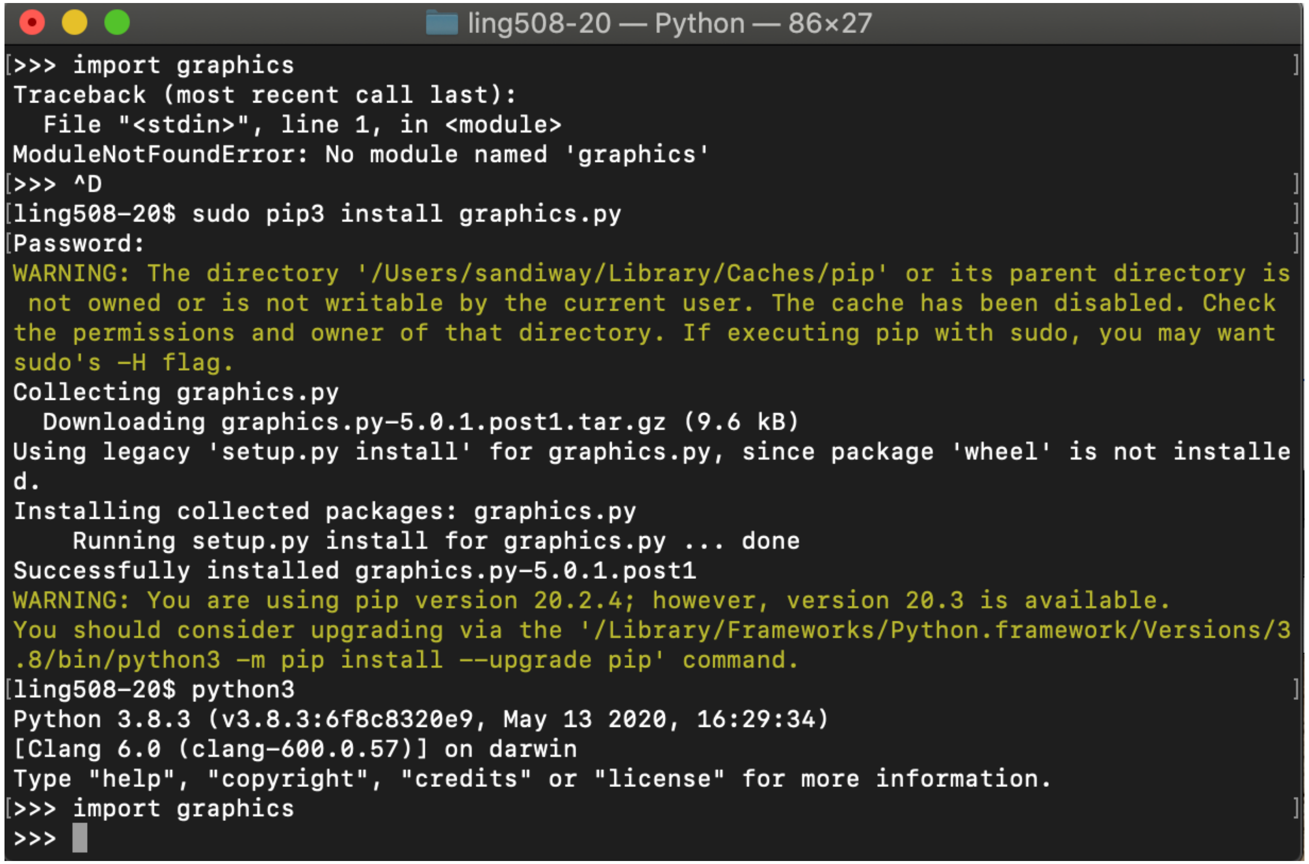
- **Example**: Chapter 5 from Zelle's pythonbook: Objects and Graphics



```
dhcp-10-142-177-223:ling508-15 sandiway$ python
Python 2.7.10 (default, Aug 22 2015, 20:33:39)
[GCC 4.2.1 Compatible Apple LLVM 7.0.0 (clang-700.0.59.1)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import graphics
>>> win = graphics.GraphWin()
>>> p = graphics.Point(50,60)
>>> p.getX()
50
>>> p.getY()
60
>>> p.draw(win)
>>> p2 = graphics.Point(140,100)
>>> p2.draw(win)
>>> l = graphics.Line(p,p2)
>>> l.draw(win)
>>> 
```

`graphics.py` installation

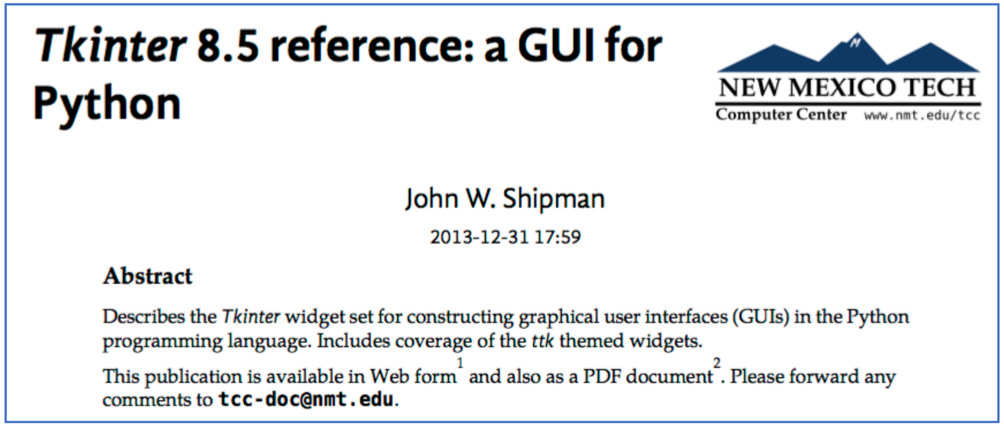
- Install:



```
[>>> import graphics
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'graphics'
[>>> ^D
[ling508-20$ sudo pip3 install graphics.py
[Password:
WARNING: The directory '/Users/sandiway/Library/Caches/pip' or its parent directory is
 not owned or is not writable by the current user. The cache has been disabled. Check
the permissions and owner of that directory. If executing pip with sudo, you may want
sudo's -H flag.
Collecting graphics.py
  Downloading graphics.py-5.0.1.post1.tar.gz (9.6 kB)
Using legacy 'setup.py install' for graphics.py, since package 'wheel' is not installe
d.
Installing collected packages: graphics.py
    Running setup.py install for graphics.py ... done
Successfully installed graphics.py-5.0.1.post1
WARNING: You are using pip version 20.2.4; however, version 20.3 is available.
You should consider upgrading via the '/Library/Frameworks/Python.framework/Versions/3
.8/bin/python3 -m pip install --upgrade pip' command.
[ling508-20$ python3
Python 3.8.3 (v3.8.3:6f8c8320e9, May 13 2020, 16:29:34)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> import graphics
>>> 
```

Graphics

- Zelle uses Tk, graphics.py is the *glue* between Tk and Python:
  - **`import graphics.py`**
- Tkinter:
  - Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. It is a thin object-oriented layer on top of Tcl/Tk.
  - **`http://infohost.nmt.edu/tcc/help/pubs/tkinter/tkinter.pdf`**



Tkinter 8.5 reference: a GUI for Python

NEW MEXICO TECH
Computer Center www.nmt.edu/tcc

John W. Shipman

2013-12-31 17:59

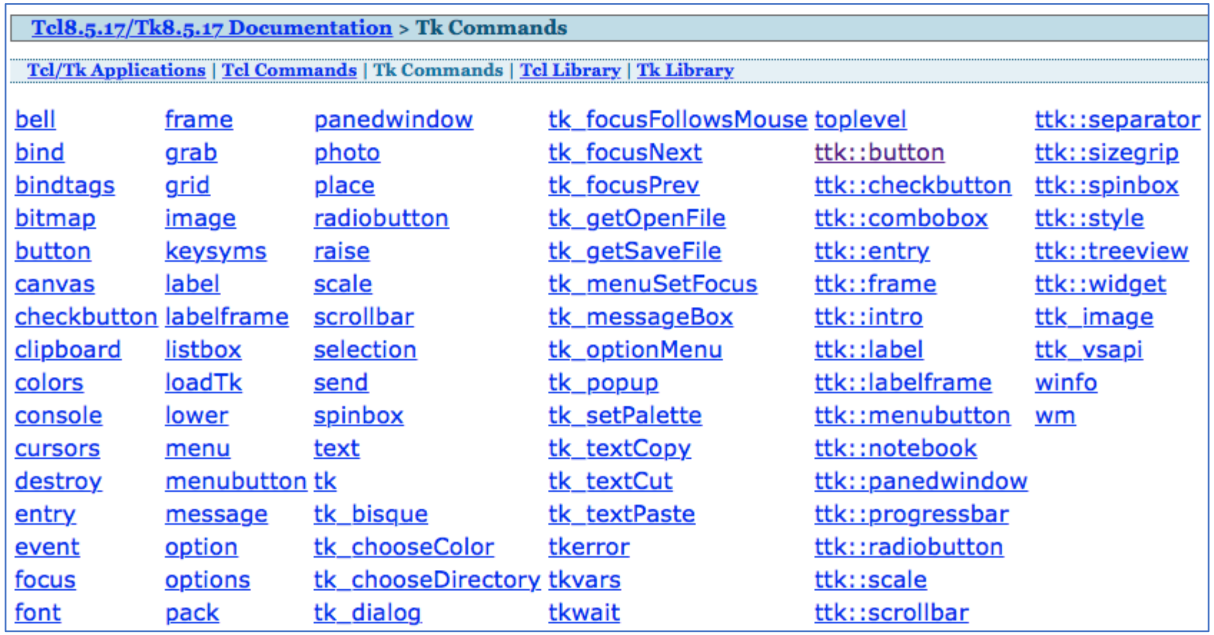
Abstract

Describes the *Tkinter* widget set for constructing graphical user interfaces (GUIs) in the Python programming language. Includes coverage of the *ttk* themed widgets.

This publication is available in Web form[1] and also as a PDF document[2]. Please forward any comments to **`tcc-doc@nmt.edu`**.

Tk

- Examples:
  - **https://www.tcl.tk/man/tcl8.5/TkCmd/contents.htm**



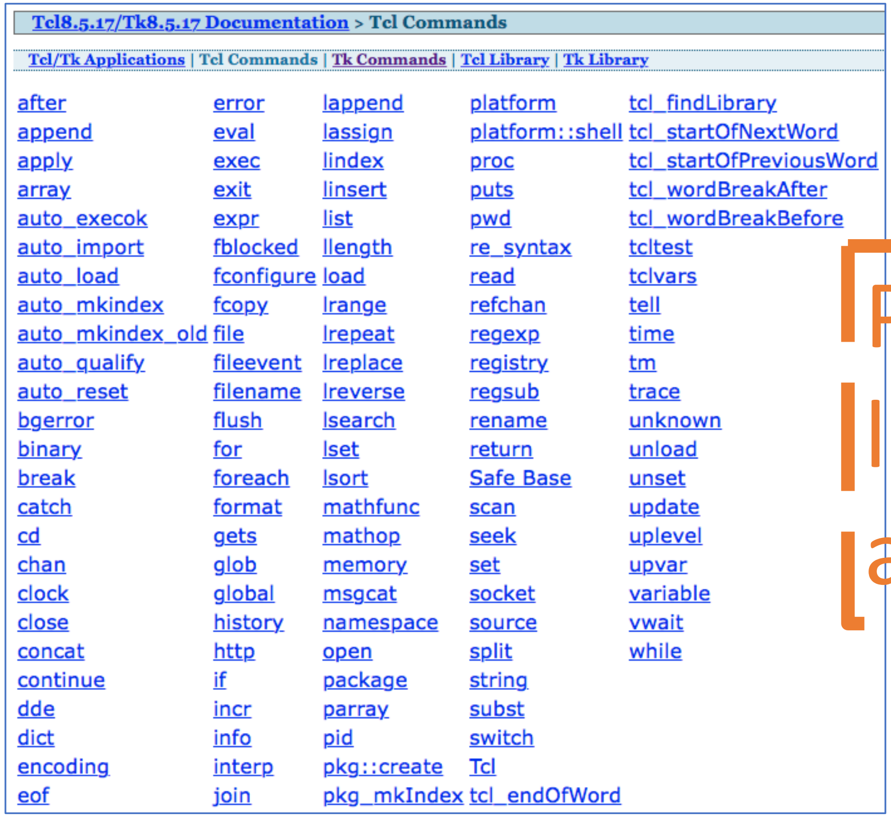
Tcl8.5.17/Tk8.5.17 Documentation > Tk Commands

Tcl/Tk Applications | Tcl Commands | Tk Commands | Tcl Library | Tk Library

bell
bind
bindtags
bitmap
button
canvas
checkbutton
clipboard
colors
console
cursors
destroy
entry
event
focus
font
frame
grab
grid
image
keysyms
label
labelframe
listbox
loadTk
lower
menu
menubutton
message
option
options
pack
panedwindow
photo
place
radiobutton
raise
scale
scrollbar
selection
send
spinbox
text
tk
tk_bisque
tk_chooseColor
tk_chooseDirectory
tk_dialog
tk_focusFollowsMouse
tk_focusNext
tk_focusPrev
tk_getOpenFile
tk_getSaveFile
tk_menuSetFocus
tk_messageBox
tk_optionMenu
tk_popup
tk_setPalette
tk_textCopy
tk_textCut
tk_textPaste
tkerror
tkvars
tkwait
toplevel
ttk::button
ttk::checkbutton
ttk::combobox
ttk::entry
ttk::frame
ttk::intro
ttk::label
ttk::labelframe
ttk::menubutton
ttk::notebook
ttk::panedwindow
ttk::progressbar
ttk::radiobutton
ttk::scale
ttk::scrollbar
ttk::separator
ttk::sizegrip
ttk::spinbox
ttk::style
ttk::treeview
ttk::widget
ttk_image
ttk_vsapi
winfo
wm

Tcl

- A simple interpreted programming language:



Tcl8.5.17/Tk8.5.17 Documentation > Tcl Commands

Tcl/Tk Applications | Tcl Commands | Tk Commands | Tcl Library | Tk Library

after
append
apply
array
auto_execok
auto_import
auto_load
auto_mkindex
auto_mkindex_old
auto_qualify
auto_reset
bgerror
binary
break
catch
cd
chan
clock
close
concat
continue
dde
dict
encoding
eof

error
eval
exec
exit
expr
fblocked
fconfigure
fcopy
file
fileevent
filename
flush
for
foreach
format
gets
glob
global
history
http
if
incr
info
interp
join

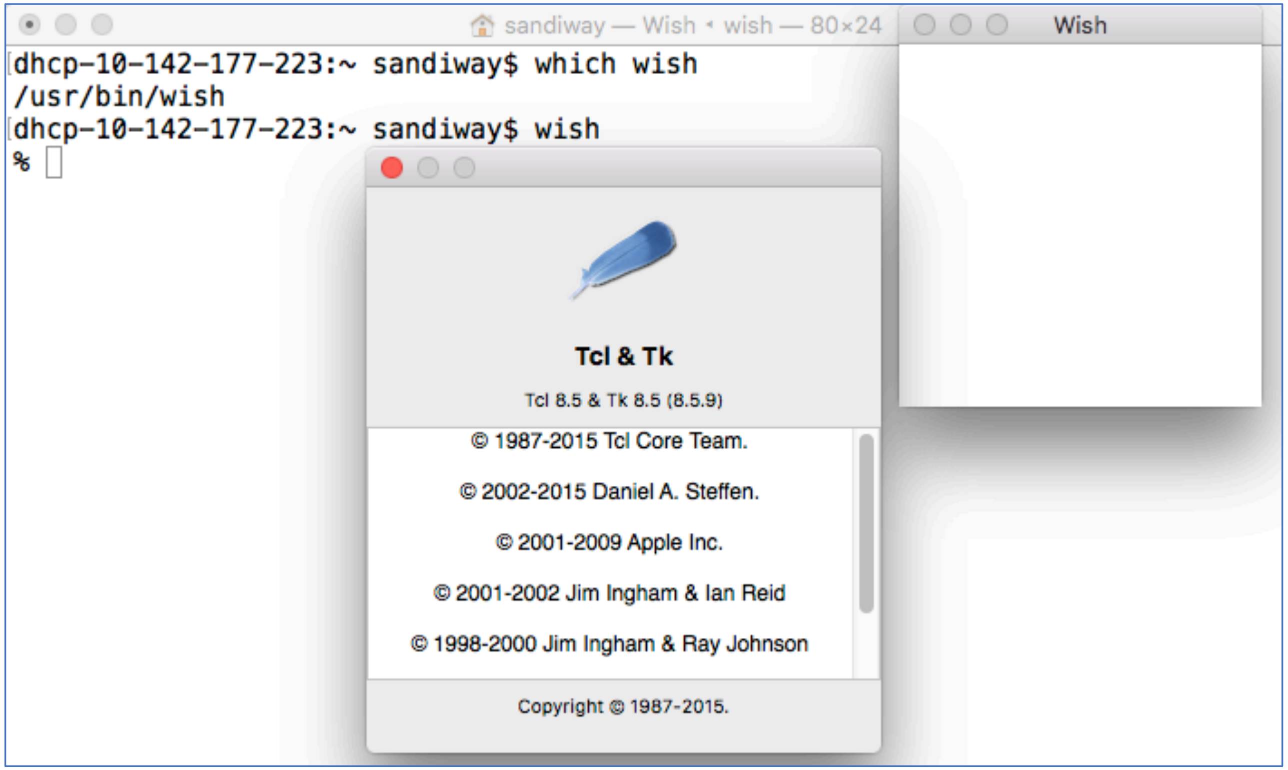
lappend
lassign
lindex
linsert
list
llength
load
lrange
lrepeat
lreplace
lreverse
lsearch
lset
lsort
mathfunc
mathop
memory
msgcat
namespace
open
package
parray
pid
pkg::create
pkg_mkIndex

platform
platform::shell
proc
puts
pwd
re_syntax
read
refchan
regexp
registry
regsub
rename
return
Safe Base
scan
seek
set
socket
source
split
string
subst
switch
Tcl
tcl_endOfWord

tcl_findLibrary
tcl_startOfNextWord
tcl_startOfPreviousWord
tcl_wordBreakAfter
tcl_wordBreakBefore
tcltest
tclvars
tell
time
tm
trace
unknown
unload
unset
update
uplevel
upvar
variable
vwait
while

Programming
languages
are everywhere!

Tcl/Tk on Mac OS X



```
dhcp-10-142-177-223:~ sandiway$ which wish
/usr/bin/wish
dhcp-10-142-177-223:~ sandiway$ wish
%
```

Tcl & Tk

Tcl 8.5 & Tk 8.5 (8.5.9)

© 1987-2015 Tcl Core Team.

© 2002-2015 Daniel A. Steffen.

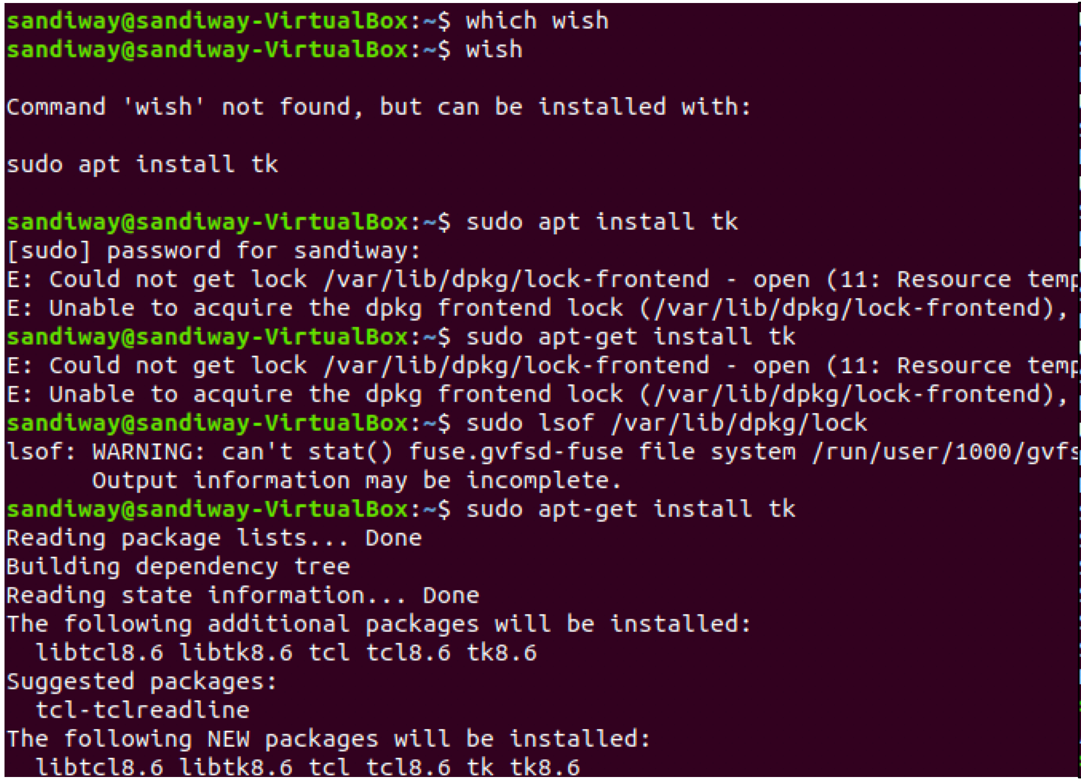
© 2001-2009 Apple Inc.

© 2001-2002 Jim Ingham & Ian Reid

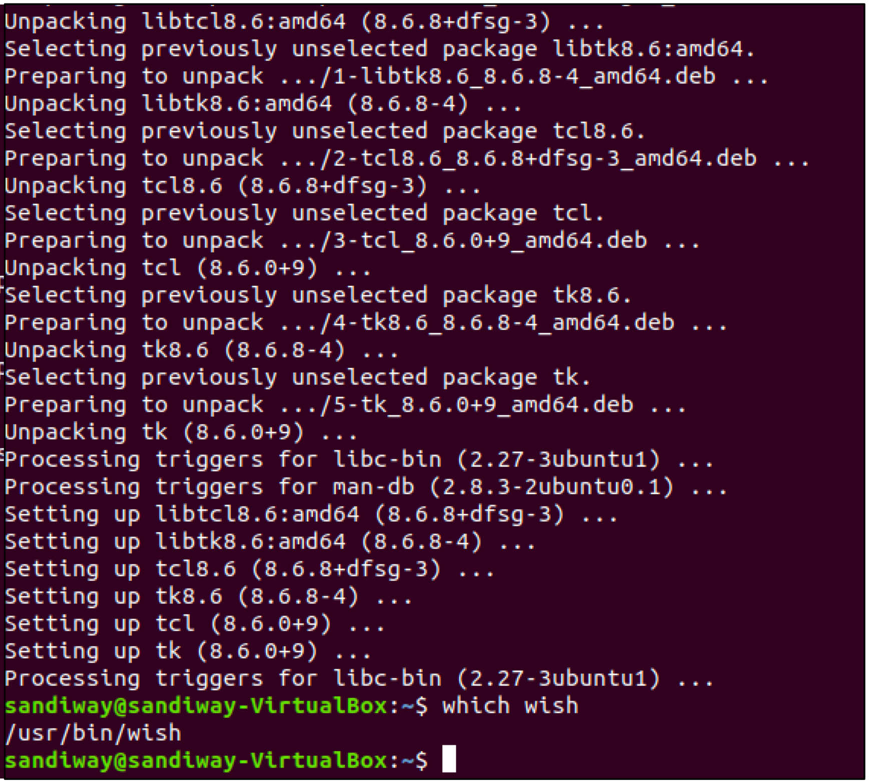
© 1998-2000 Jim Ingham & Ray Johnson

Copyright © 1987-2015.

Tcl/Tk on Ubuntu

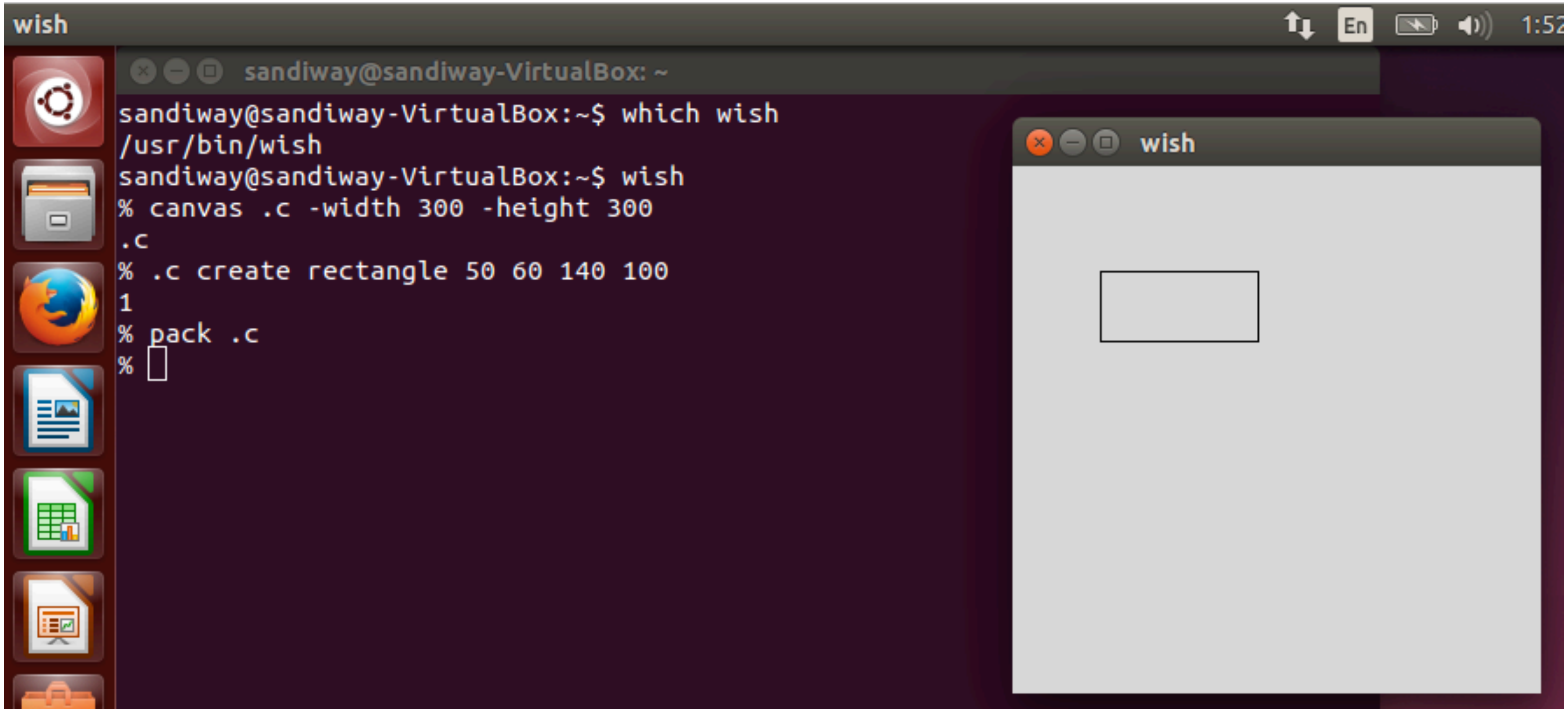


```
sandiway@sandiway-VirtualBox:~$ which wish
sandiway@sandiway-VirtualBox:~$ wish

Command 'wish' not found, but can be installed with:

sudo apt install tk

sandiway@sandiway-VirtualBox:~$ sudo apt install tk
[sudo] password for sandiway:
E: Could not get lock /var/lib/dpkg/lock-frontend - open (11: Resource temp
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend),
sandiway@sandiway-VirtualBox:~$ sudo apt-get install tk
E: Could not get lock /var/lib/dpkg/lock-frontend - open (11: Resource temp
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend),
sandiway@sandiway-VirtualBox:~$ sudo lsof /var/lib/dpkg/lock
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1000/gvfs
      Output information may be incomplete.
sandiway@sandiway-VirtualBox:~$ sudo apt-get install tk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libtcl8.6 libtk8.6 tcl tcl8.6 tk8.6
Suggested packages:
  tcl-tclreadline
The following NEW packages will be installed:
  libtcl8.6 libtk8.6 tcl tcl8.6 tk tk8.6
```



```
Unpacking libtcl8.6:amd64 (8.6.8+dfsg-3) ...
Selecting previously unselected package libtk8.6:amd64.
Preparing to unpack .../1-libtk8.6_8.6.8-4_amd64.deb ...
Unpacking libtk8.6:amd64 (8.6.8-4) ...
Selecting previously unselected package tcl8.6.
Preparing to unpack .../2-tcl8.6_8.6.8+dfsg-3_amd64.deb ...
Unpacking tcl8.6 (8.6.8+dfsg-3) ...
Selecting previously unselected package tcl.
Preparing to unpack .../3-tcl_8.6.0+9_amd64.deb ...
Unpacking tcl (8.6.0+9) ...
Selecting previously unselected package tk8.6.
Preparing to unpack .../4-tk8.6_8.6.8-4_amd64.deb ...
Unpacking tk8.6 (8.6.8-4) ...
Selecting previously unselected package tk.
Preparing to unpack .../5-tk_8.6.0+9_amd64.deb ...
Unpacking tk (8.6.0+9) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up libtcl8.6:amd64 (8.6.8+dfsg-3) ...
Setting up libtk8.6:amd64 (8.6.8-4) ...
Setting up tcl8.6 (8.6.8+dfsg-3) ...
Setting up tk8.6 (8.6.8-4) ...
Setting up tcl (8.6.0+9) ...
Setting up tk (8.6.0+9) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
sandiway@sandiway-VirtualBox:~$ which wish
/usr/bin/wish
sandiway@sandiway-VirtualBox:~$
```

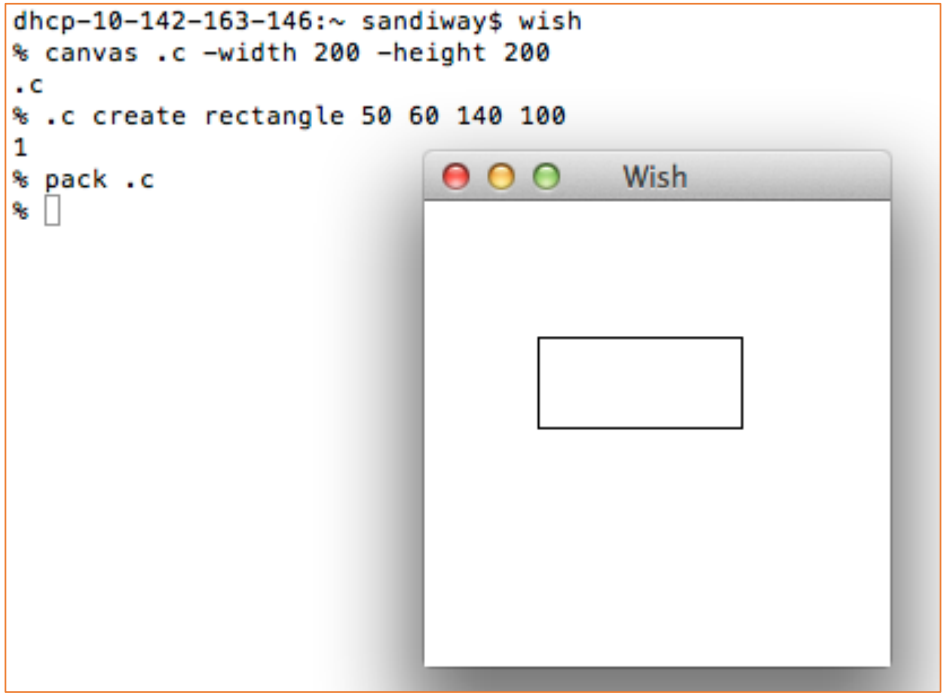
Tcl/Tk on Ubuntu



```
sandiway@sandiway-VirtualBox:~$ which wish
/usr/bin/wish
sandiway@sandiway-VirtualBox:~$ wish
% canvas .c -width 300 -height 300
.c
% .c create rectangle 50 60 140 100
1
% pack .c
% 
```

Tcl/Tk Examples

- Command line:
  - wish   interpreter for tcl/tk



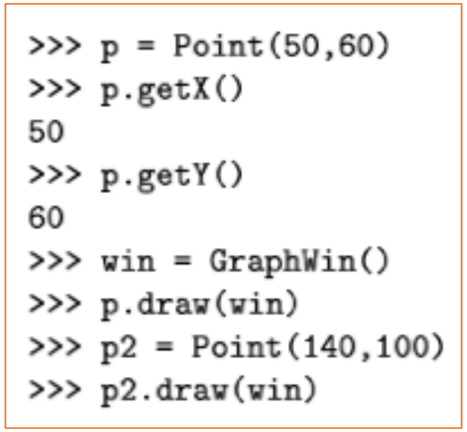
```
dhcp-10-142-163-146:~ sandiway$ wish
% canvas .c -width 200 -height 200
.c
% .c create rectangle 50 60 140 100
1
% pack .c
% 
```

Concepts:

canvas	place to draw coordinate system (0,0) topleft
pack	where to put canvas

graphics.py

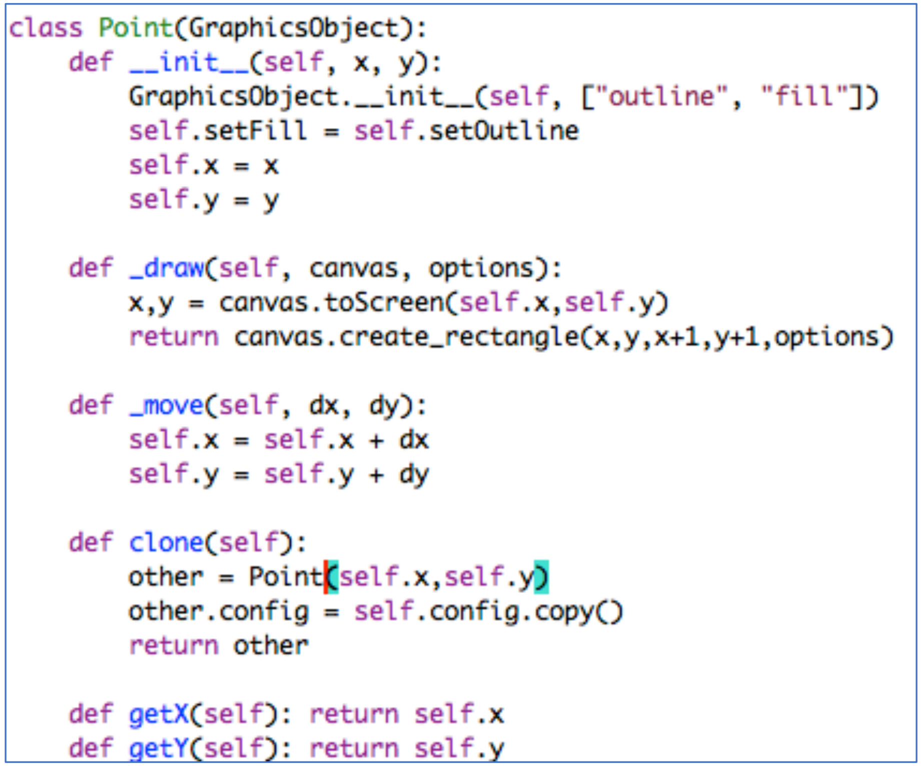
- Point:



```
>>> p = Point(50,60)
>>> p.getX()
50
>>> p.getY()
60
>>> win = GraphWin()
>>> p.draw(win)
>>> p2 = Point(140,100)
>>> p2.draw(win)
```

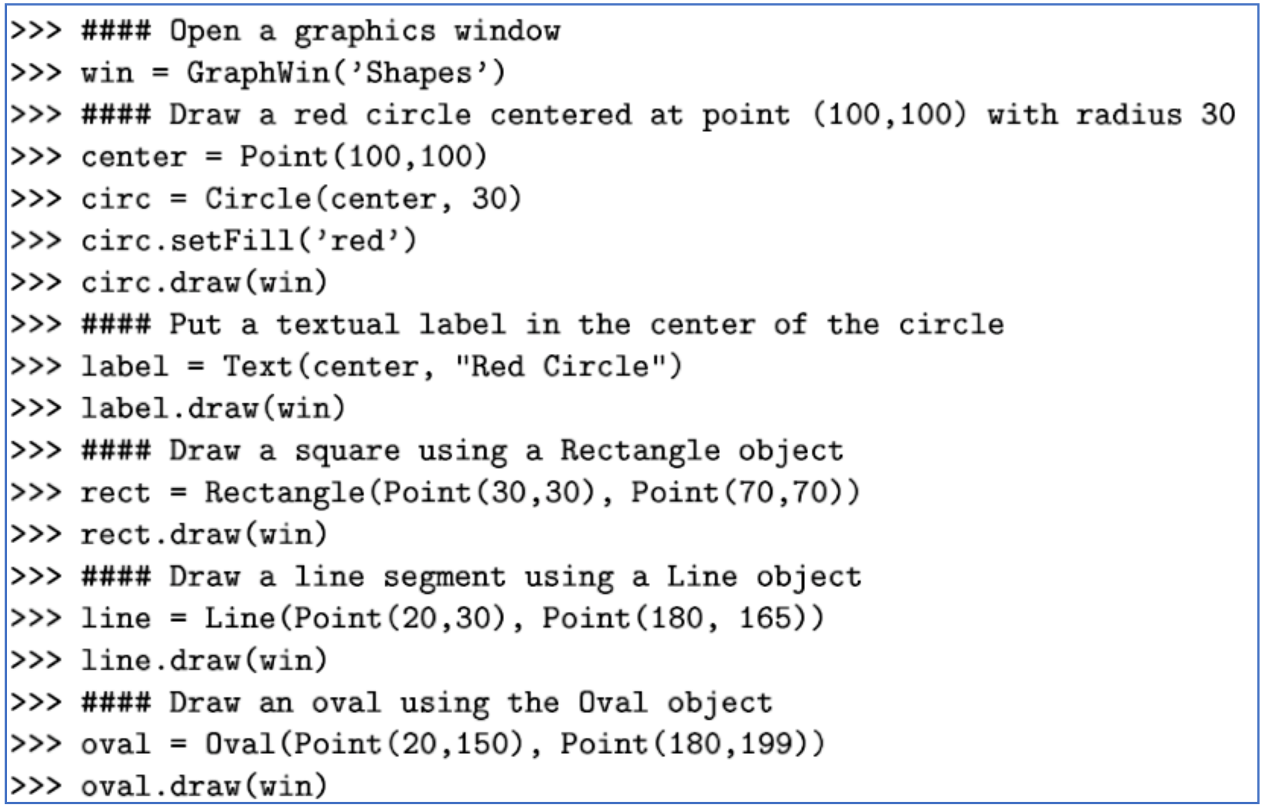
```
sudo apt-get install python3-tk
```

Implemented as a rectangle:



```
class Point(GraphicsObject):
    def __init__(self, x, y):
        GraphicsObject.__init__(self, ["outline", "fill"])
        self.setFill = self.setOutline
        self.x = x
        self.y = y

    def _draw(self, canvas, options):
        x,y = canvas.toScreen(self.x,self.y)
        return canvas.create_rectangle(x,y,x+1,y+1,options)

    def _move(self, dx, dy):
        self.x = self.x + dx
        self.y = self.y + dy

    def clone(self):
        other = Point(self.x,self.y)
        other.config = self.config.copy()
        return other

    def getX(self): return self.x
    def getY(self): return self.y
```

graphics.py

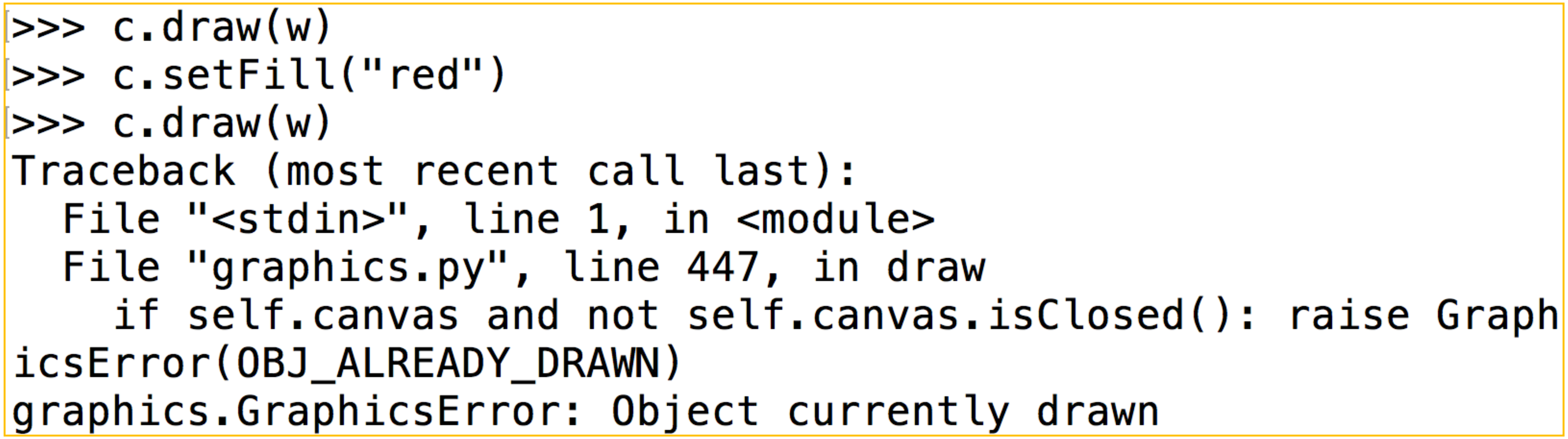


```
>>> #### Open a graphics window
>>> win = GraphWin(’Shapes’)
>>> #### Draw a red circle centered at point (100,100) with radius 30
>>> center = Point(100,100)
>>> circ = Circle(center, 30)
>>> circ.setFill(’red’)
>>> circ.draw(win)
>>> #### Put a textual label in the center of the circle
>>> label = Text(center, "Red Circle")
>>> label.draw(win)
>>> #### Draw a square using a Rectangle object
>>> rect = Rectangle(Point(30,30), Point(70,70))
>>> rect.draw(win)
>>> #### Draw a line segment using a Line object
>>> line = Line(Point(20,30), Point(180, 165))
>>> line.draw(win)
>>> #### Draw an oval using the Oval object
>>> oval = Oval(Point(20,150), Point(180,199))
>>> oval.draw(win)
```

page 129

graphics.py

- Call method **draw()** once only

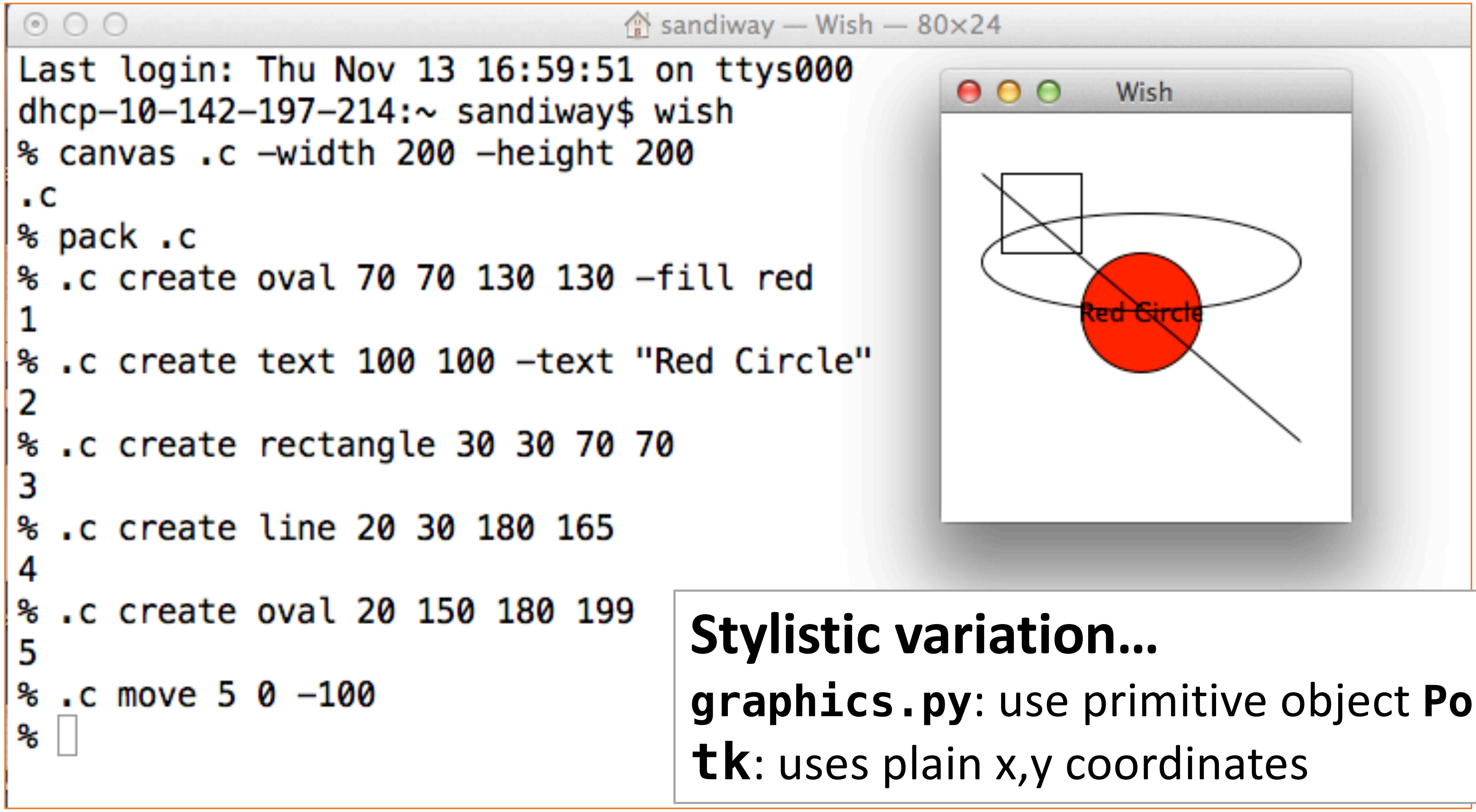


```
>>> c.draw(w)
>>> c.setFill("red")
>>> c.draw(w)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "graphics.py", line 447, in draw
    if self.canvas and not self.canvas.isClosed(): raise Graph
icsError(OBJ_ALREADY_DRAWN)
graphics.GraphicsError: Object currently drawn
```

graphics.py

- Object inheritance (see graphics.py code):
- _Bbox: move
    - Rectangle: .draw()
    - Oval: .draw()
        - Circle: .getRadius()
    - Line: .draw(), .setArrow()

Directly in **`tcl/tk`**



```
Last login: Thu Nov 13 16:59:51 on ttys000
dhcp-10-142-197-214:~ sandiway$ wish
% canvas .c -width 200 -height 200
.c
% pack .c
% .c create oval 70 70 130 130 -fill red
1
% .c create text 100 100 -text "Red Circle"
2
% .c create rectangle 30 30 70 70
3
% .c create line 20 30 180 165
4
% .c create oval 20 150 180 199
5
% .c move 5 0 -100
% 
```

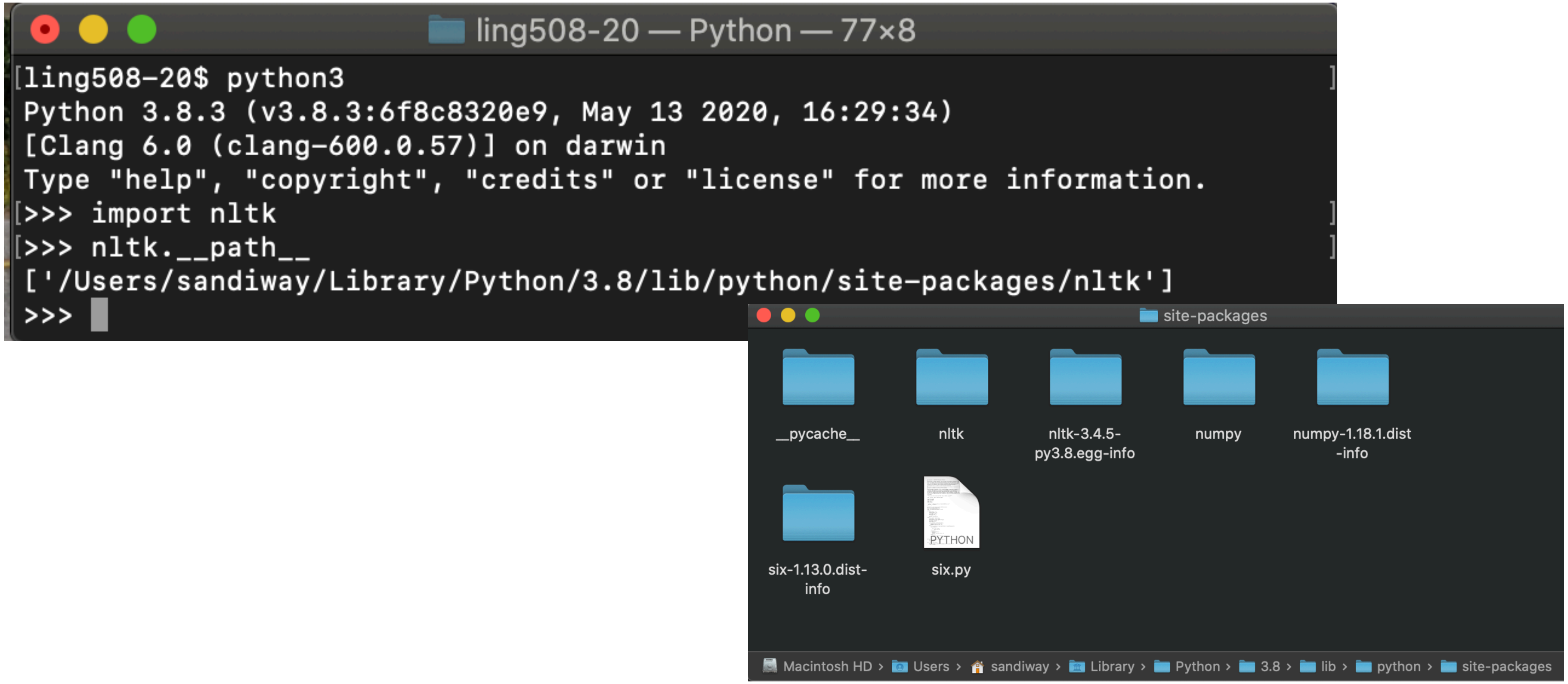
Stylistic variation…

`graphics.py`: use primitive object **`Point`**

`tk`: uses plain x,y coordinates

nltk

- Where is it installed on my computer?

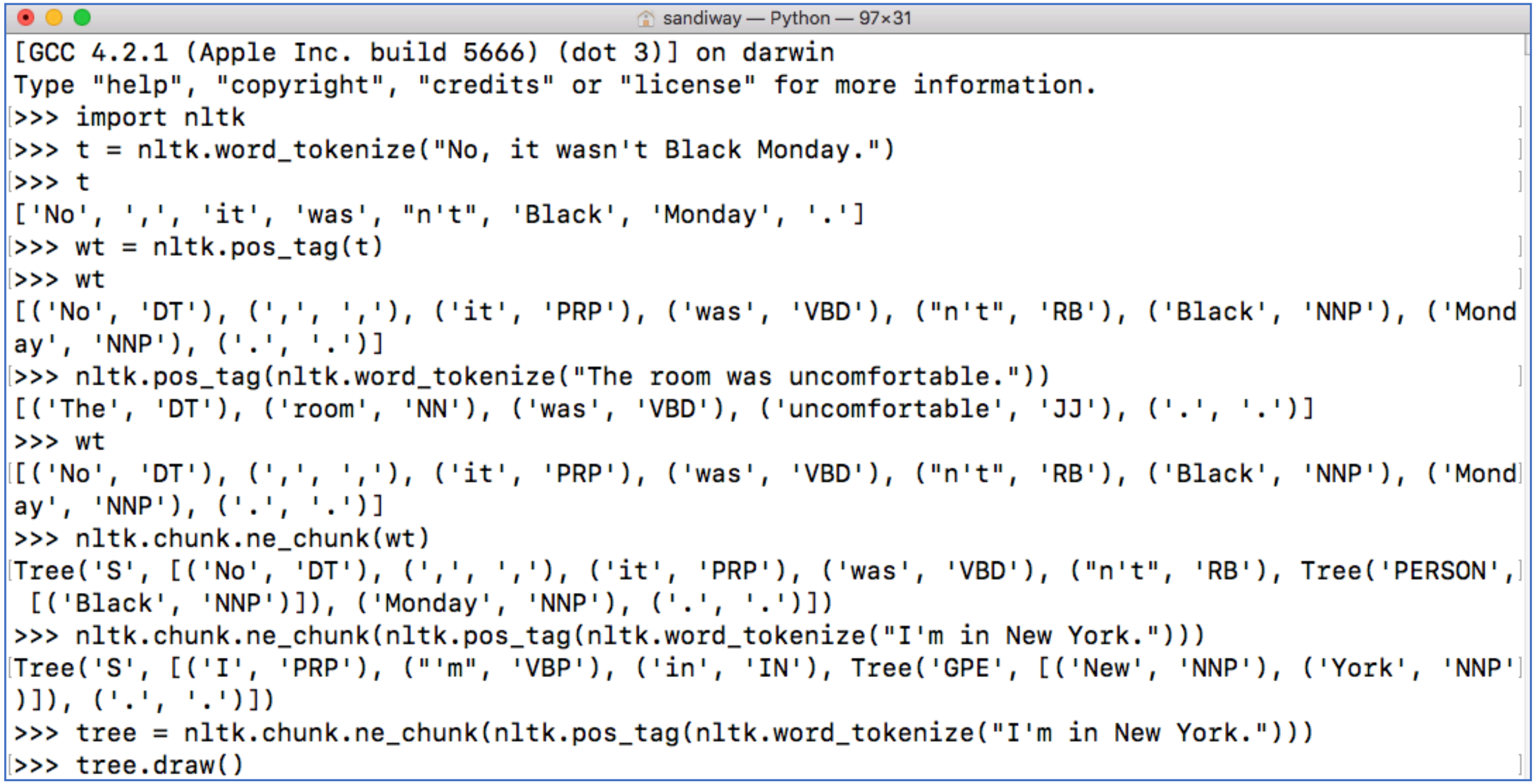


ling508-20 — Python — 77×8

```
ling508-20$ python3
Python 3.8.3 (v3.8.3:6f8c8320e9, May 13 2020, 16:29:34)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> nltk.__path__
['/Users/sandiway/Library/Python/3.8/lib/python/site-packages/nltk']
>>> 
```

site-packages
__pycache__
nltk
nltk-3.4.5-py3.8.egg-info
numpy
numpy-1.18.1.dist-info
six-1.13.0.dist-info
PYTHON
six.py
Macintosh HD › Users › sandiway › Library › Python › 3.8 › lib › python › site-packages

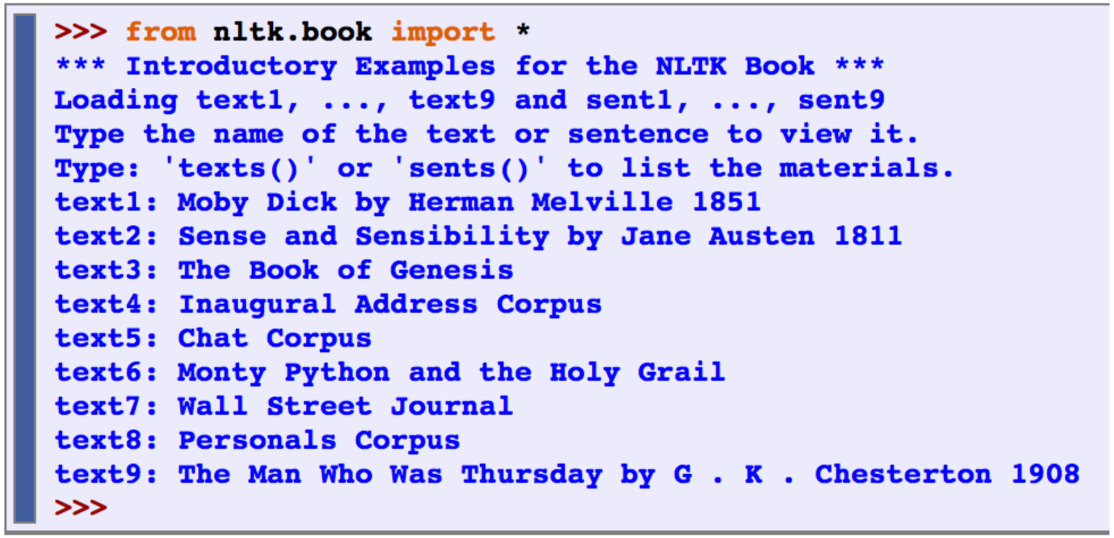
Example:



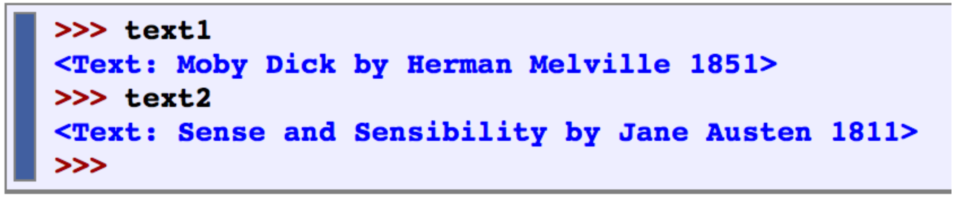
```
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> t = nltk.word_tokenize("No, it wasn't Black Monday.")
>>> t
['No', ',', 'it', 'was', "n't", 'Black', 'Monday', '.']
>>> wt = nltk.pos_tag(t)
>>> wt
[('No', 'DT'), (',', ','), ('it', 'PRP'), ('was', 'VBD'), ("n't", 'RB'), ('Black', 'NNP'), ('Mond
ay', 'NNP'), ('.', '.')]
>>> nltk.pos_tag(nltk.word_tokenize("The room was uncomfortable."))
[('The', 'DT'), ('room', 'NN'), ('was', 'VBD'), ('uncomfortable', 'JJ'), ('.', '.')]
>>> wt
[('No', 'DT'), (',', ','), ('it', 'PRP'), ('was', 'VBD'), ("n't", 'RB'), ('Black', 'NNP'), ('Mond
ay', 'NNP'), ('.', '.')]
>>> nltk.chunk.ne_chunk(wt)
Tree('S', [('No', 'DT'), (',', ','), ('it', 'PRP'), ('was', 'VBD'), ("n't", 'RB'), Tree('PERSON',
 [('Black', 'NNP')]), ('Monday', 'NNP'), ('.', '.')])
>>> nltk.chunk.ne_chunk(nltk.pos_tag(nltk.word_tokenize("I'm in New York.")))
Tree('S', [('I', 'PRP'), ("'m", 'VBP'), ('in', 'IN'), Tree('GPE', [('New', 'NNP'), ('York', 'NNP'
)]), ('.', '.')])
>>> tree = nltk.chunk.ne_chunk(nltk.pos_tag(nltk.word_tokenize("I'm in New York.")))
>>> tree.draw()
```

nltk book: Language Processing and Python

1 Computing with Language: Texts and Words: http://www.nltk.org/book/ch01.html

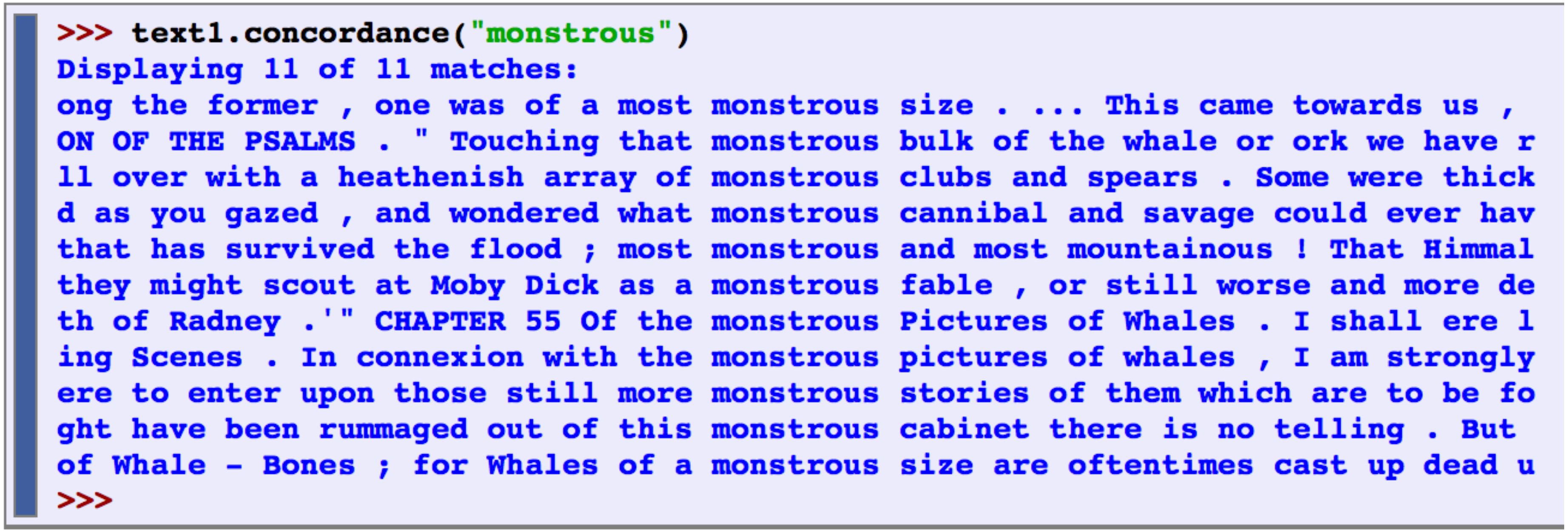


```
>>> from nltk.book import *
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
>>>
```



```
>>> text1
<Text: Moby Dick by Herman Melville 1851>
>>> text2
<Text: Sense and Sensibility by Jane Austen 1811>
>>>
```

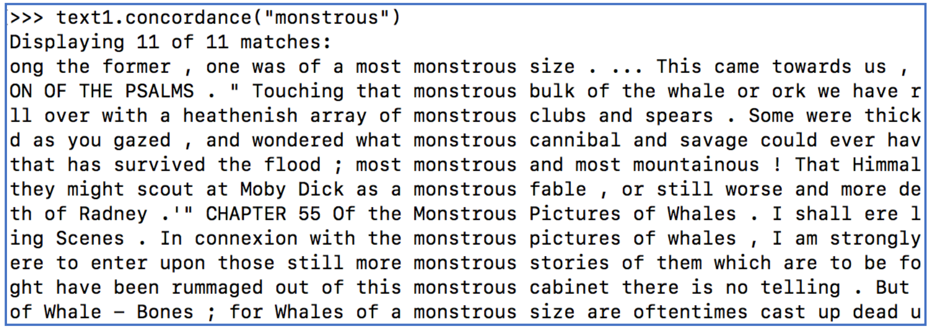
nltk book: Language Processing and Python



```
>>> text1.concordance("monstrous")
Displaying 11 of 11 matches:
ong the former , one was of a most monstrous size . ... This came towards us ,
ON OF THE PSALMS . " Touching that monstrous bulk of the whale or ork we have r
ll over with a heathenish array of monstrous clubs and spears . Some were thick
d as you gazed , and wondered what monstrous cannibal and savage could ever hav
that has survived the flood ; most monstrous and most mountainous ! That Himmal
they might scout at Moby Dick as a monstrous fable , or still worse and more de
th of Radney .'" CHAPTER 55 Of the monstrous Pictures of Whales . I shall ere l
ing Scenes . In connexion with the monstrous pictures of whales , I am strongly
ere to enter upon those still more monstrous stories of them which are to be fo
ght have been rummaged out of this monstrous cabinet there is no telling . But
of Whale - Bones ; for Whales of a monstrous size are oftentimes cast up dead u
>>>
```

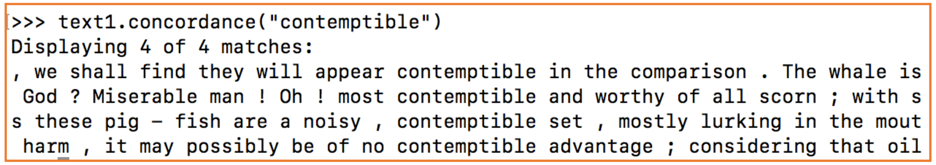
nltk book: Language Processing and Python

monstrous

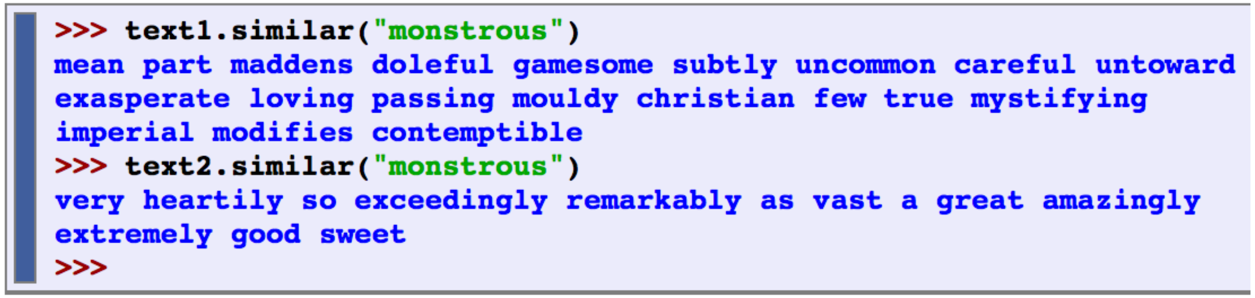


```
>>> text1.concordance("monstrous")
Displaying 11 of 11 matches:
ong the former , one was of a most monstrous size . ... This came towards us ,
ON OF THE PSALMS . " Touching that monstrous bulk of the whale or ork we have r
ll over with a heathenish array of monstrous clubs and spears . Some were thick
d as you gazed , and wondered what monstrous cannibal and savage could ever hav
that has survived the flood ; most monstrous and most mountainous ! That Himmal
they might scout at Moby Dick as a monstrous fable , or still worse and more de
th of Radney .'" CHAPTER 55 Of the Monstrous Pictures of Whales . I shall ere l
ing Scenes . In connexion with the monstrous pictures of whales , I am strongly
ere to enter upon those still more monstrous stories of them which are to be fo
ght have been rummaged out of this monstrous cabinet there is no telling . But
of Whale - Bones ; for Whales of a monstrous size are oftentimes cast up dead u
```

contemptible



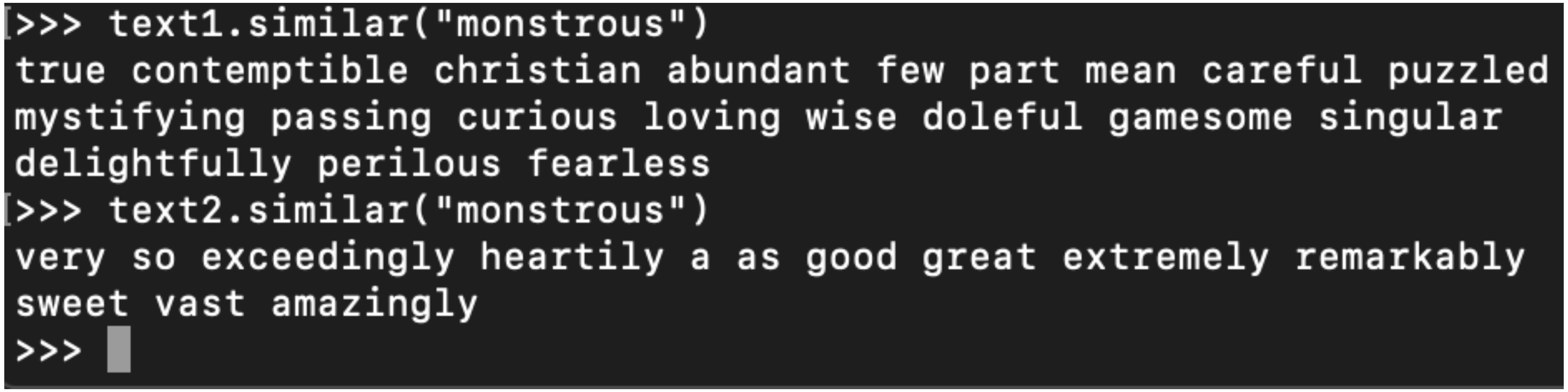
```
>>> text1.concordance("contemptible")
Displaying 4 of 4 matches:
, we shall find they will appear contemptible in the comparison . The whale is
 God ? Miserable man ! Oh ! most contemptible and worthy of all scorn ; with s
s these pig - fish are a noisy , contemptible set , mostly lurking in the mout
 harm , it may possibly be of no contemptible advantage ; considering that oil
```



```
>>> text1.similar("monstrous")
mean part maddens doleful gamesome subtly uncommon careful untoward
exasperate loving passing mouldy christian few true mystifying
imperial modifies contemptible
>>> text2.similar("monstrous")
very heartily so exceedingly remarkably as vast a great amazingly
extremely good sweet
>>>
```

nltk book: Language Processing and Python

- On my mac
  - your order may be different depending on dict implementation:

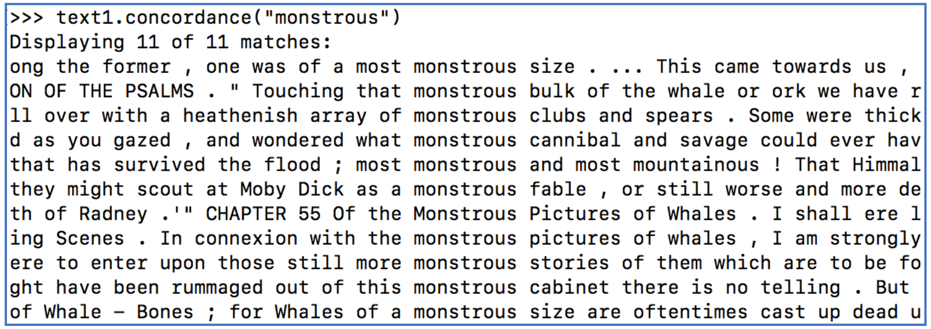


```
>>> text1.similar("monstrous")
true contemptible christian abundant few part mean careful puzzled
mystifying passing curious loving wise doleful gamesome singular
delightfully perilous fearless
>>> text2.similar("monstrous")
very so exceedingly heartily a as good great extremely remarkably
sweet vast amazingly
>>>
```

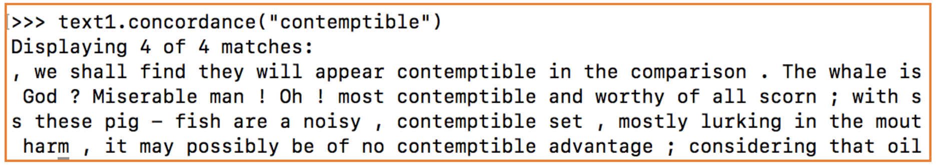
nltk book: Language Processing and Python

monstrous

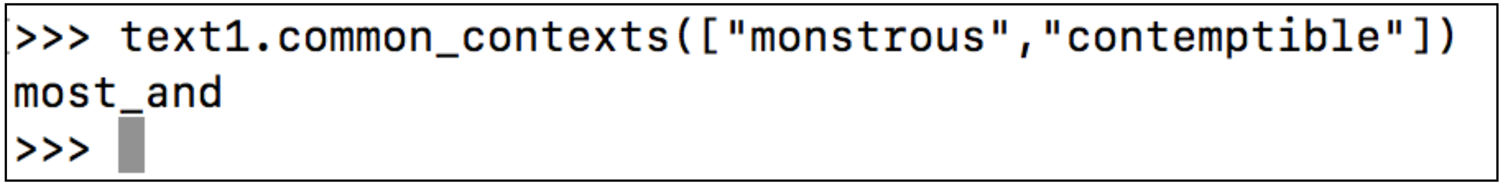
contemptible



```
>>> text1.concordance("monstrous")
Displaying 11 of 11 matches:
ong the former , one was of a most monstrous size . ... This came towards us ,
ON OF THE PSALMS . " Touching that monstrous bulk of the whale or ork we have r
ll over with a heathenish array of monstrous clubs and spears . Some were thick
d as you gazed , and wondered what monstrous cannibal and savage could ever hav
that has survived the flood ; most monstrous and most mountainous ! That Himmal
they might scout at Moby Dick as a monstrous fable , or still worse and more de
th of Radney .'" CHAPTER 55 Of the Monstrous Pictures of Whales . I shall ere l
ing Scenes . In connexion with the monstrous pictures of whales , I am strongly
ere to enter upon those still more monstrous stories of them which are to be fo
ght have been rummaged out of this monstrous cabinet there is no telling . But
of Whale - Bones ; for Whales of a monstrous size are oftentimes cast up dead u
```

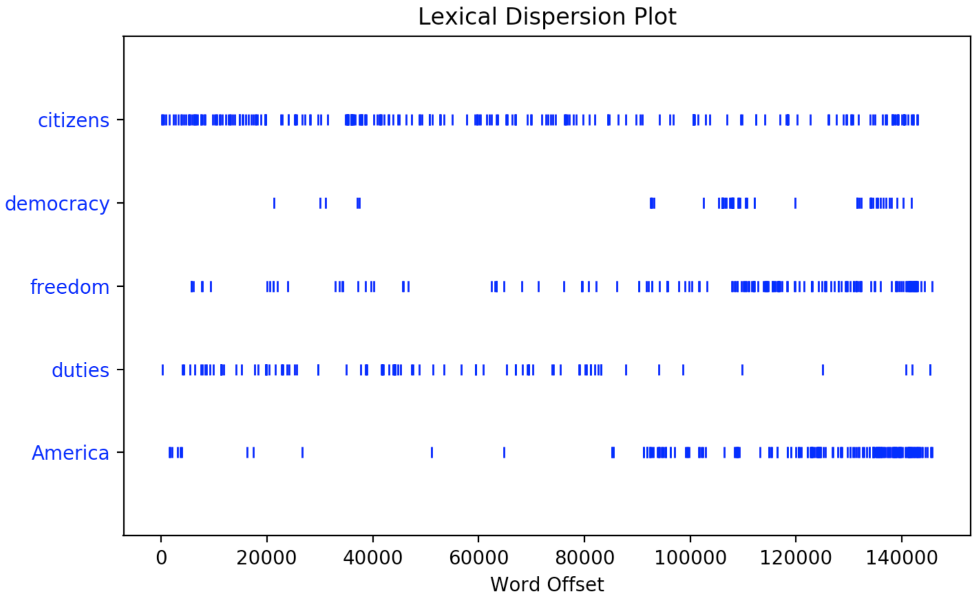


```
>>> text1.concordance("contemptible")
Displaying 4 of 4 matches:
, we shall find they will appear contemptible in the comparison . The whale is
 God ? Miserable man ! Oh ! most contemptible and worthy of all scorn ; with s
s these pig - fish are a noisy , contemptible set , mostly lurking in the mout
 harm , it may possibly be of no contemptible advantage ; considering that oil
```



```
>>> text1.common_contexts(["monstrous","contemptible"])
most_and
>>>
```

nltk book: Language Processing and Python



Inaugural
Presidential
Addresses

```
text4.dispersion_plot(["citizens", "democracy", "freedom", "duties", "America"])
```

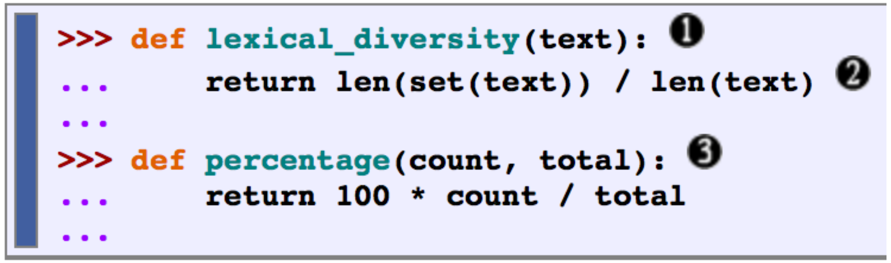
nltk book: Language Processing and Python

- 1.4 Counting Vocabulary
  - `<text>` placeholder for some text object
  - `<word>` placeholder for a word

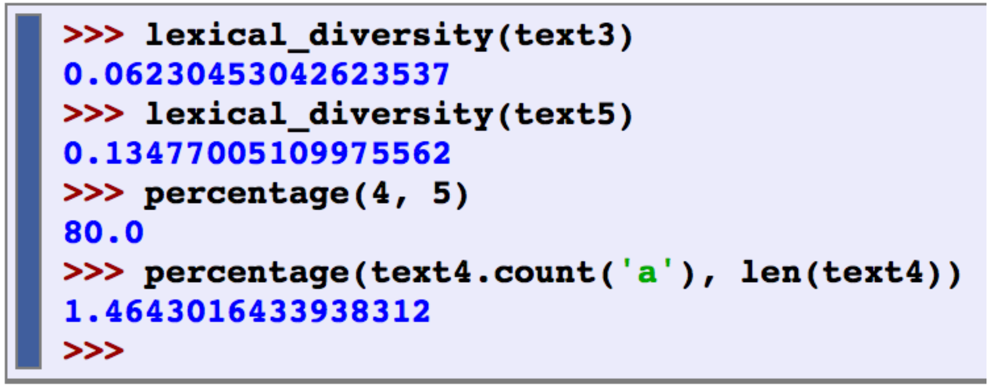
1.	`len(<text>)`	word count
2.	`set(<text>)`	no duplicate words
3.	`len(set(<text>))`	no. of different words
4.	`len(set(<text>)) / len(<text>)`	lexical diversity
5.	`<text>.count(<word>)`	# of times <word> occurs in <text>
6.	`100 * <text>.count(<word>) / len(<text>)`	% of <text> taken up by <word>

nltk book: Language Processing and Python

- Defining functions



```
>>> def lexical_diversity(text): ❶
...     return len(set(text)) / len(text) ❷
...
>>> def percentage(count, total): ❸
...     return 100 * count / total
...
```



```
>>> lexical_diversity(text3)
0.06230453042623537
>>> lexical_diversity(text5)
0.13477005109975562
>>> percentage(4, 5)
80.0
>>> percentage(text4.count('a'), len(text4))
1.4643016433938312
>>>
```