

Lecture 21

*408/508 Computational
Techniques for Linguists*

Next Programming Language

So far, in this course:

1. Binary: encoding of numbers and characters (Unicode)
2. Bash: command line and shell scripting: `#!/bin/bash`
3. HTML, CSS, Javascript + DOM (*how to access HTML elements*)
4. Apache2 Webserver (serve HTML pages, run a program: GET, POST)

For the rest of the semester, we'll use:

- Python (**homework:** *install it if necessary*)

Today's Topics

- Why Python
- Homework 9: install Python 3 if not already there
- **Heads up:** Homework 10: install nltk (natural language toolkit)
- Python numbers
- Examples of what you can do with Python + nltk

Why Python?

NLTK 3.0 documentation

[NEXT](#) | [MODULES](#) | [INDEX](#)

Natural Language Toolkit

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to [over 50 corpora and lexical resources](#) such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active [discussion forum](#).

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

[Natural Language Processing with Python](#) provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The book is being updated for Python 3 and NLTK 3. (The original Python 2 version is still available at http://nltk.org/book_1ed.)

[Some simple things you can do with NLTK](#)

- NLTK is written in Python

- macOS: may need to install Developer Tools first...

WSL on Windows: *already installed*

```
sandiway@DESKTOP-VEPP64: ~$ lsb_release -d
Description:      Ubuntu 20.04.6 LTS
sandiway@DESKTOP-VEPP64: ~$ which python
/usr/bin/python
sandiway@DESKTOP-VEPP64: ~$ which python3
/usr/bin/python3
sandiway@DESKTOP-VEPP64: ~$ python3
Python 3.8.10 (default, May 26 2023, 14:05:08)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exit
Use exit() or Ctrl-D (i.e. EOF) to exit
>>> exit()
sandiway@DESKTOP-VEPP64: ~$ python
Python 3.8.10 (default, May 26 2023, 14:05:08)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
sandiway@DESKTOP-VEPP64: ~$ which pip
/home/sandiway/.local/bin/pip
sandiway@DESKTOP-VEPP64: ~$ which pip3
/home/sandiway/.local/bin/pip3
sandiway@DESKTOP-VEPP64: ~$
```

Python3 on Windows 10/11

The screenshot displays the Microsoft Store interface for the Python 3.11 application. The page features a search bar at the top, a navigation sidebar on the left, and a main content area. The main content area includes the Python 3.11 logo, the developer name 'Python Software Foundation', a 'Get' button, a 4.2 star average rating from 449 users, and an ESRB 'E' rating for 'EVERYONE'. Below this, there are sections for 'Screenshots' and 'Description'. The 'Screenshots' section shows two windows: a Command Prompt window running Python 3.11 and an IDLE Shell window. The Command Prompt window shows the following output:

```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

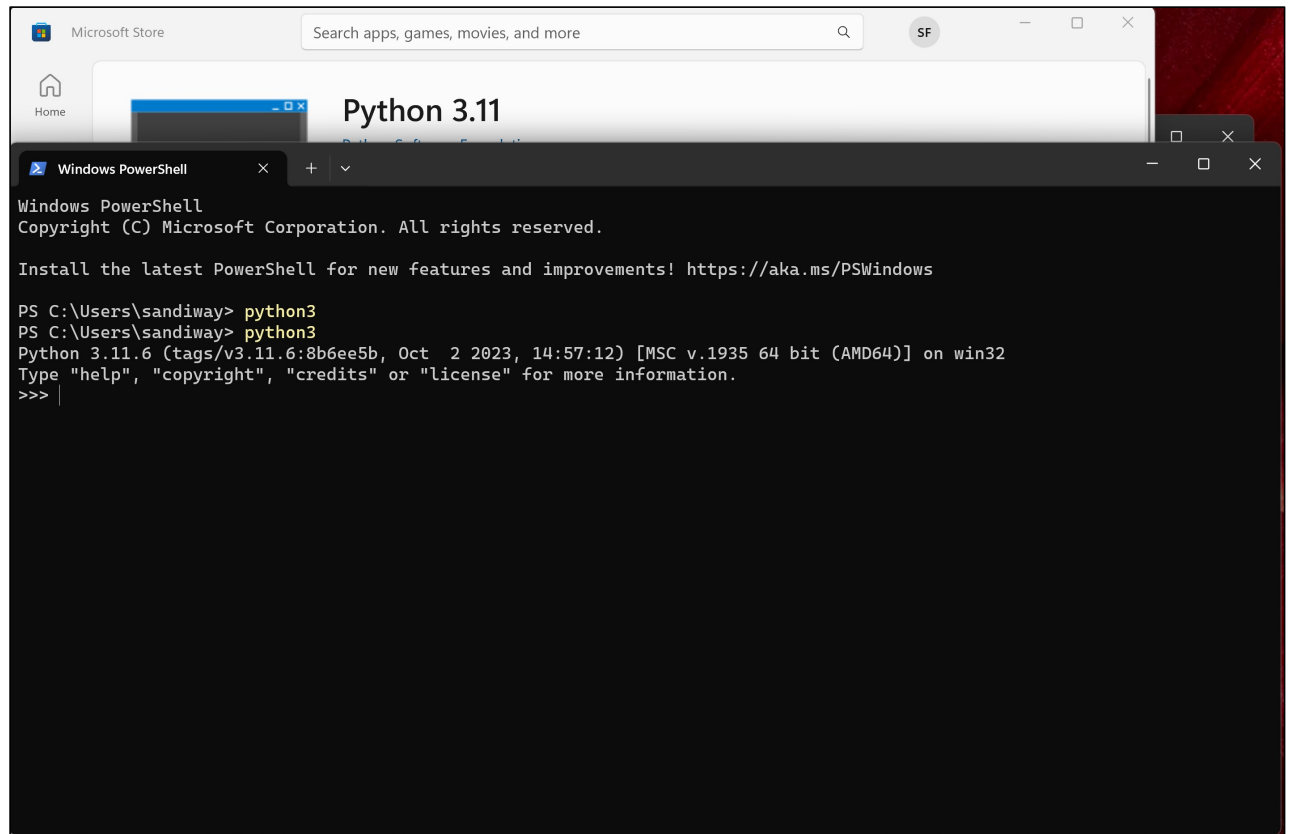
C:\Users\Pythonista>python3.11
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import sys
>>> print(f"Welcome, to Python {sys.version_info.major}.{sys.version_info.minor}")
Welcome, to Python 3.11
>>> |
```

The IDLE Shell window shows the following output:

```
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12)
Type "help", "copyright", "credits" or "license()" for mor
>>>
```

The 'Description' section is currently empty. The Windows taskbar is visible at the bottom of the screen.

Python3 on Windows 10/11



The screenshot shows a Windows Store window for Python 3.11 and a Windows PowerShell terminal window. The PowerShell window displays the output of running the 'python3' command, showing the Python version and build information.

```
Microsoft Store
Search apps, games, movies, and more
Python 3.11

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\sandiway> python3
PS C:\Users\sandiway> python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

Install Python 3

- python.org: install python3 on Windows or macOS (*if not installed*)

The screenshot shows the Python.org website with the navigation menu open to 'Downloads'. The 'macOS' option is selected, leading to the 'Download for macOS' page. The 'Python 3.12.0' button is highlighted with a red box. A text box on the right says 'do not install Python 2.7'. The page also includes a search bar, a 'Donate' button, and a terminal window showing Python code execution.

```
# Python 3: Si
>>> 1 / 2
0.5
>>> 2 ** 3
8
>>> 17 / 3 #
5.666666666666666
>>> 17 // 3 #
5
```

python™

Donate Search GO

About Downloads Documentation Community Success Stories News Events

All releases
Source code
Windows
macOS
Other Platforms
License
Alternative Implementations

Download for macOS

Python 3.12.0

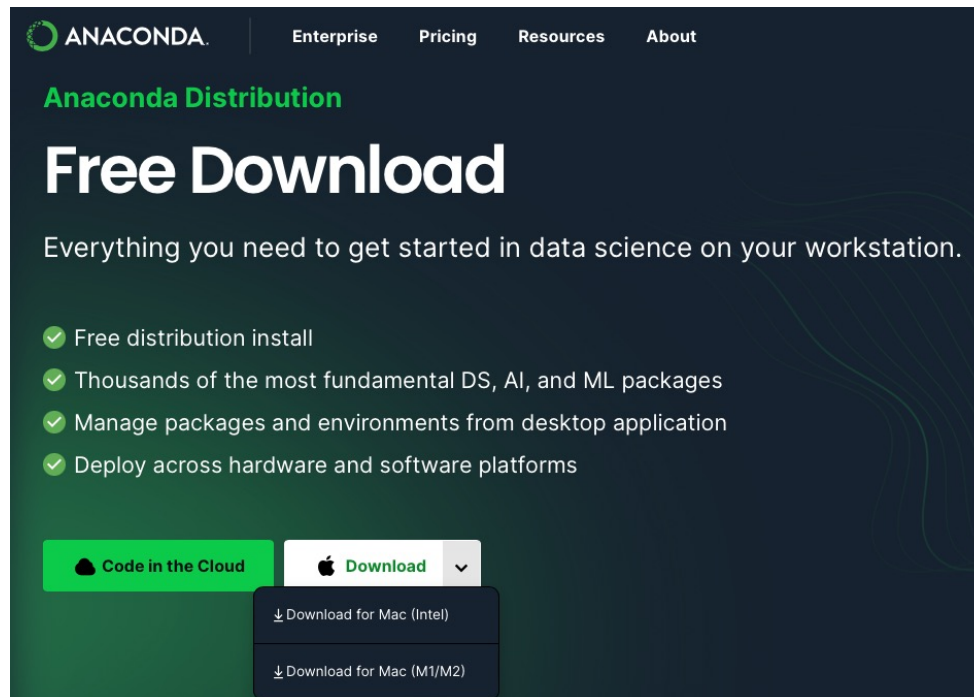
Not the OS you are looking for? Python can be used on many operating systems and environments. [View the full list of downloads.](#)

rouping. [More](#)

do not install Python 2.7

Install Python 3

- <https://www.anaconda.com/download>
- <https://docs.anaconda.com/free/anaconda/install/windows/>



The screenshot shows the Anaconda website's 'Free Download' page. The header includes the Anaconda logo and navigation links for Enterprise, Pricing, Resources, and About. The main heading is 'Free Download' with the subtext 'Anaconda Distribution'. Below this, a tagline reads 'Everything you need to get started in data science on your workstation.' A list of four benefits is provided, each with a green checkmark: 'Free distribution install', 'Thousands of the most fundamental DS, AI, and ML packages', 'Manage packages and environments from desktop application', and 'Deploy across hardware and software platforms'. At the bottom, there are two buttons: 'Code in the Cloud' and 'Download'. The 'Download' button is open, showing two options: 'Download for Mac (Intel)' and 'Download for Mac (M1/M2)'.

ANACONDA. Enterprise Pricing Resources About

Anaconda Distribution

Free Download

Everything you need to get started in data science on your workstation.

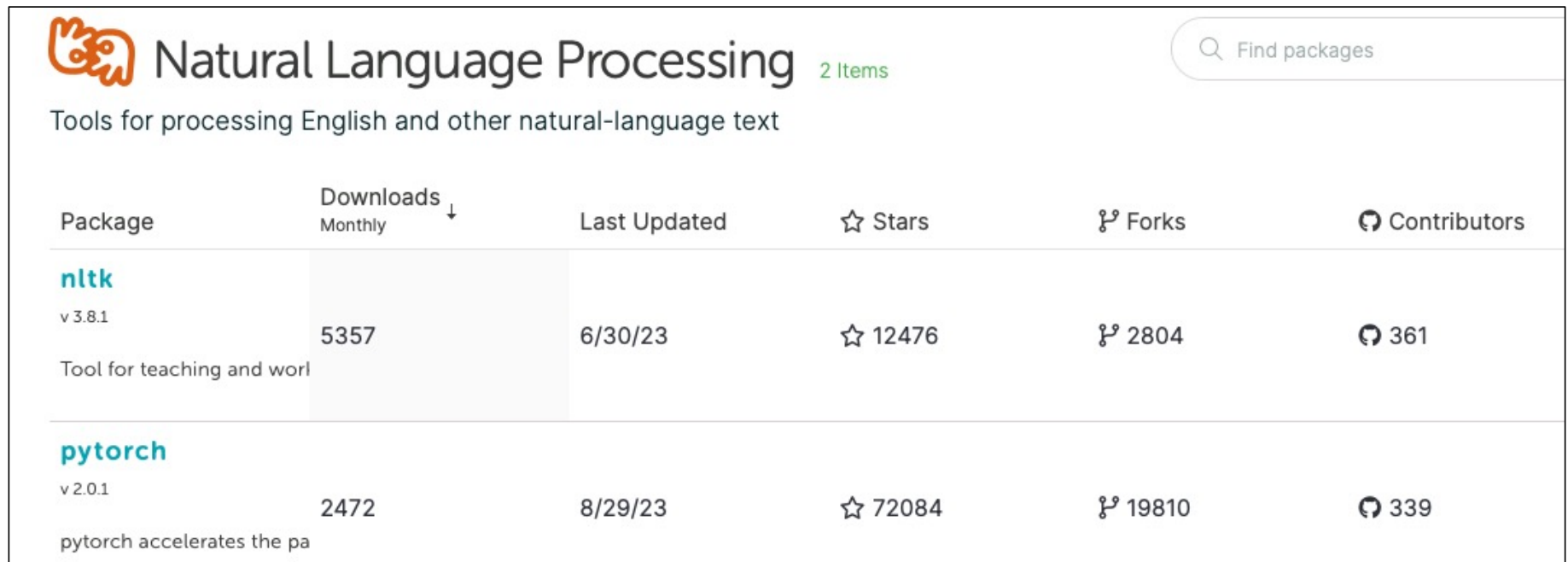
- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms


Code in the Cloud Download

- ↓ Download for Mac (Intel)
- ↓ Download for Mac (M1/M2)

Install Python 3

- <https://anaconda.cloud/package-categories/natural-language-processing>



 **Natural Language Processing** 2 Items

Tools for processing English and other natural-language text

Package	Downloads Monthly ↓	Last Updated	☆ Stars	🔗 Forks	👤 Contributors
nltk v 3.8.1 Tool for teaching and worl	5357	6/30/23	☆ 12476	🔗 2804	👤 361
pytorch v 2.0.1 pytorch accelerates the pa	2472	8/29/23	☆ 72084	🔗 19810	👤 339

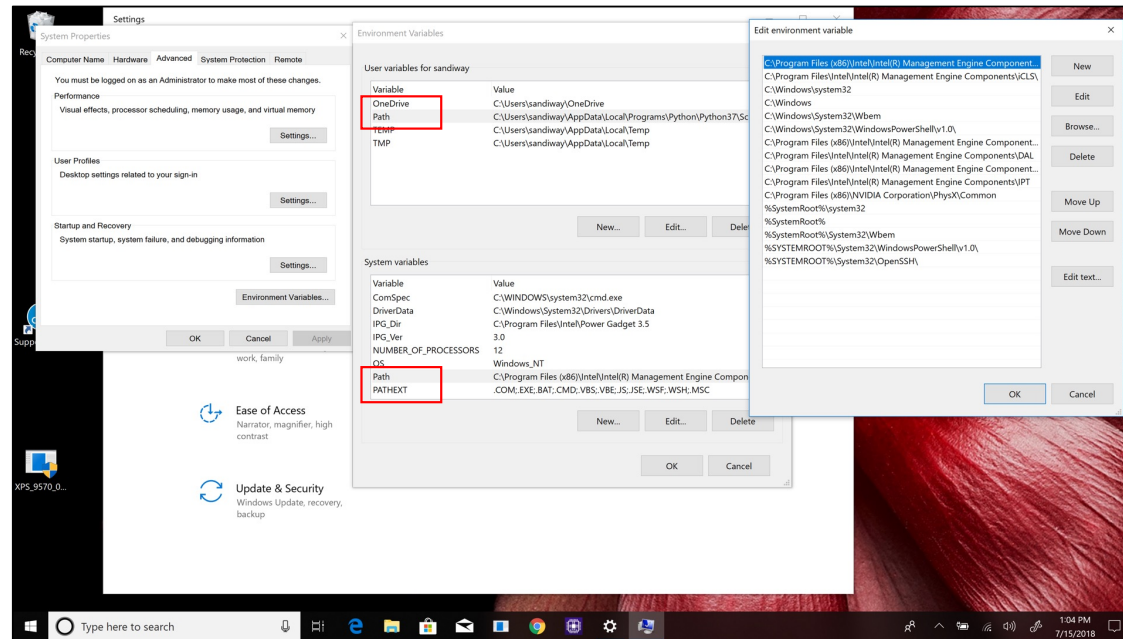
Python 2.7: Obsoleted after macOS Catalina

- macOS Catalina (version 10.15) - 2019

```
[~$ /usr/bin/python  
WARNING: Python 2.7 is not recommended.  
This version is included in macOS for compatibility with legacy software.  
Future versions of macOS will not include Python 2.7.  
Instead, it is recommended that you transition to using 'python3' from within Terminal.  
  
Python 2.7.16 (default, Nov 9 2019, 05:55:08)  
[GCC 4.2.1 Compatible Apple LLVM 11.0.0 (clang-1100.0.32.4) (-macos10.15-objc-s  
on darwin  
Type "help", "copyright", "credits" or "license" for more information.  
[>>> ^D
```

Windows 10: Environment Variables

- if you need to manually add the directory for the Python executable to your PATH





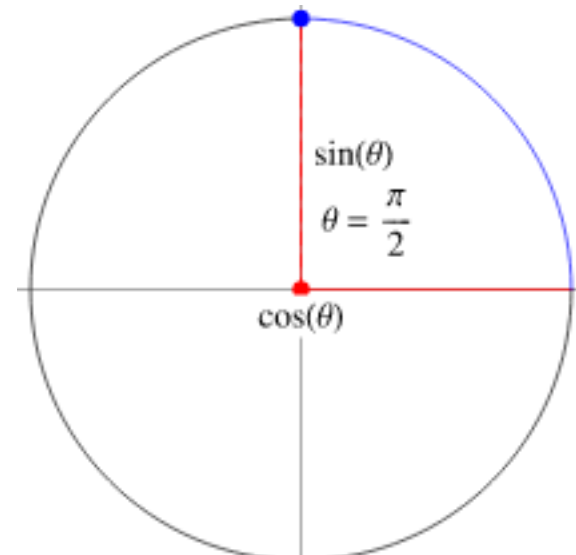
Introduction

- Start with using the Python interpreter to do simple things, e.g. as a calculator.
- Later, we'll install the Natural Language Toolkit (nltk) and this will allow to do some interesting operations on language data.
- A brief look ahead to some examples...

Python: Numbers

- At the interpreter:

```
$ python3
Python 3.9.12 (main, Jun 1 2022, 06:34:44)
[Clang 12.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more
information.
>>> 4+5
9
>>> math.pi
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'math' is not defined
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(math.pi/2)
1.0
>>>
```



Python: Numbers

type: built-in function

```
>>> type(2*3-1)
<class 'int'>
>>> type(math.pi)
<class 'float'>
>>> import sys
>>> sys.maxsize
9223372036854775807
>>> type(sys.maxsize)
<class 'int'>
>>> type(sys.maxsize+1)
<class 'int'>
>>> sys.int_info
sys.int_info(bits_per_digit=30, sizeof_digit=4)
```

$\mathbf{\leftarrow}$ `math.log(sys.maxsize)/math.log(2)`
`63.0 (bits)`

arithmetic operators:

operator	operation
+	addition
-	subtraction
*	multiplication
/	division
**	exponentiation
%	remainder
abs()	absolute value

Table 3.1: Python built-in numeric operations.

Python integers

- Recall bc in the Bash shell?
- Python 3: int can also go to any size (*limited by available memory*):

```
>>> import sys
>>> sys.int_info
sys.int_info(bits_per_digit=30, sizeof_digit=4)
>>> sys.maxsize
9223372036854775807
>>> 2**63 - 1
9223372036854775807
>>> █
```

```
>>> 2**1000
10715086071862673209484250490600018105614048117055336074437503883703510511249361224931983788156958581275946729
17553146825187145285692314043598457757469857480393456777482423098542107460506237114187795418215304647498358194
1267398767559165543946077062914571196477686542167660429831652624386837205668069376
```


Python: Numbers

```
import math  
math.pi
```

```
[>>> from math import pi, sin  
[>>> sin(pi/2)  
1.0
```

Python	Mathematics	English
<code>pi</code>	π	An approximation of pi.
<code>e</code>	e	An approximation of e .
<code>sin(x)</code>	$\sin x$	The sine of x .
<code>cos(x)</code>	$\cos x$	The cosine of x .
<code>tan(x)</code>	$\tan x$	The tangent of x .
<code>asin(x)</code>	$\arcsin x$	The inverse of sine x .
<code>acos(x)</code>	$\arccos x$	The inverse of cosine x .
<code>atan(x)</code>	$\arctan x$	The inverse of tangent x .
<code>log(x)</code>	$\ln x$	The natural (base e) logarithm of x
<code>log10(x)</code>	$\log_{10} x$	The common (base 10) logarithm of x .
<code>exp(x)</code>	e^x	The exponential of x .
<code>ceil(x)</code>	$\lceil x \rceil$	The smallest whole number $\geq x$
<code>floor(x)</code>	$\lfloor x \rfloor$	The largest whole number $\leq x$

Table 3.2: Some math library functions.

Python: complex numbers

- Example:

sqrt is the square root function, e.g. $\text{sqrt}(4)=2$, $\text{sqrt}(9)=3$ etc.

```
>>> math.sqrt(-1)
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
ValueError: math domain error
```

- Complex number library:

- <https://docs.python.org/3/library/cmath.html>

- i is j in Python

```
>>> import cmath
```

```
>>> cmath.sqrt(-1)
```

```
1j
```

```
>>> i = cmath.sqrt(-1)
```

```
>>> i*i
```

```
(-1+0j)
```

Python: complex numbers

$$e^{i\pi} + 1 = 0$$

- Euler's Identity:
- https://en.wikipedia.org/wiki/Euler%27s_identity

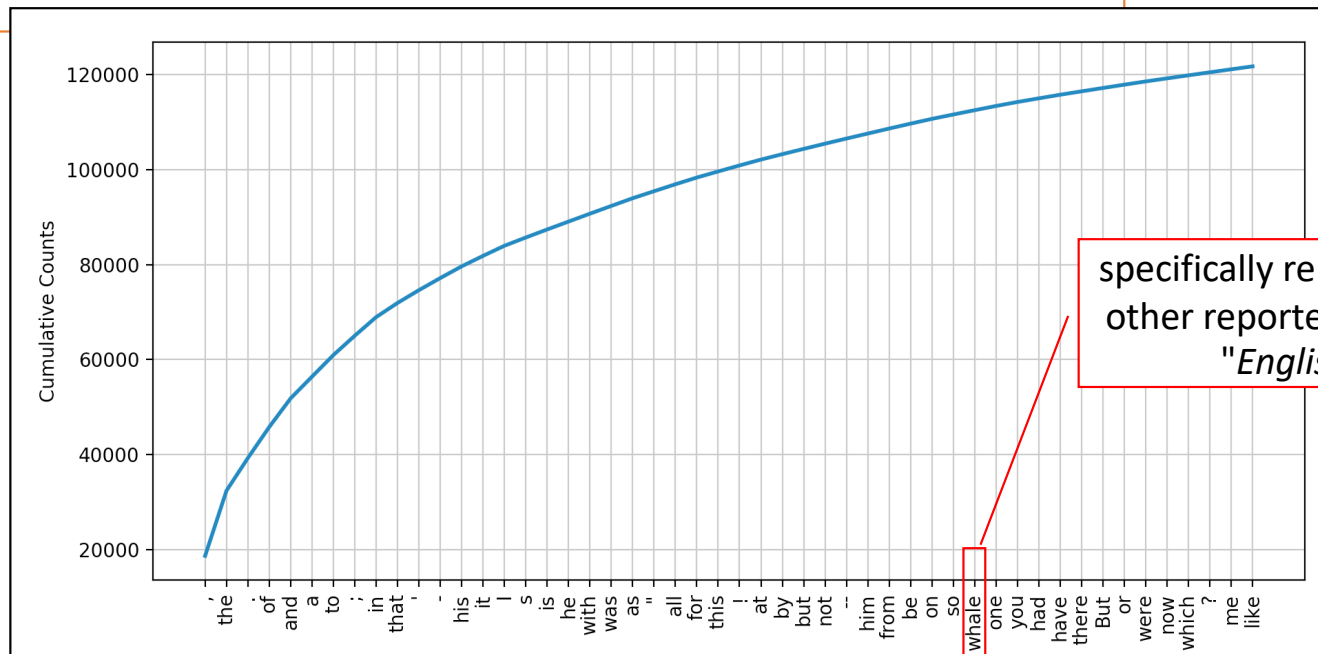
```
~$ python3
Python 3.8.3 (v3.8.3:6f8c8320e9, May 13 2020, 16:29:34)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> import cmath
[>>> i = cmath.sqrt(-1)
[>>> from math import exp, pi
[>>> exp(i*pi) + 1
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can't convert complex to float
[>>> from cmath import exp
[>>> exp(i*pi) + 1
1.2246467991473532e-16j
>>>
```

Why Python?

- `nltk`: natural language toolkit (nltk.org)
- install this (*now or homework for next time*)

nltk: Distribution of words in *Moby Dick*

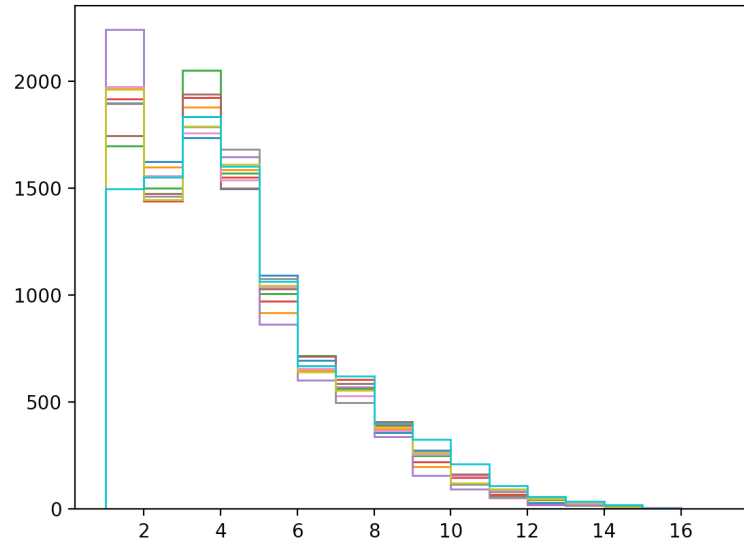
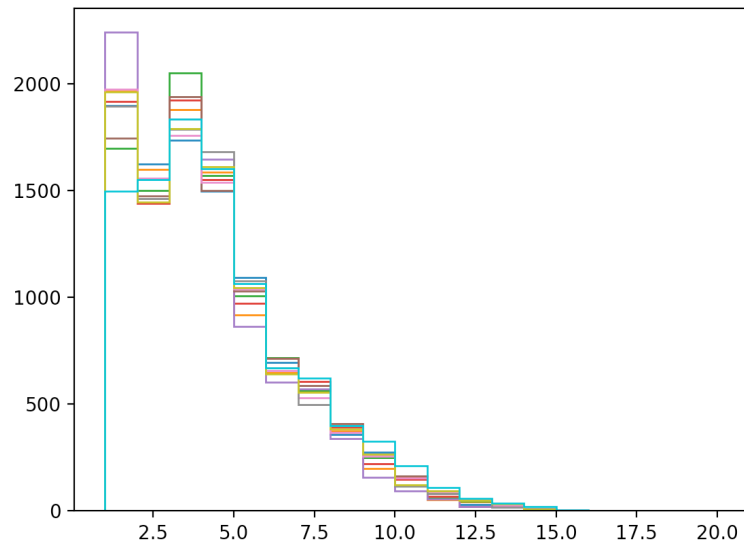
```
>>> print(fdlist1)
<FreqDist with 19317 samples and 260819 outcomes>
>>> fdlist1.most_common(20)
[(',', 18713), ('the', 13721), ('.', 6862), ('of', 6536), ('and', 6024), ('a', 4569), ('to', 4542), (';', 4072), ('in', 3916), ('that', 2982), ('"', 2684), ('-', 2552), ('his', 2459), ('it', 2209), ('I', 2124), ('s', 1739), ('is', 1695), ('he', 1661), ('with', 1659), ('was', 1632)]
>>> fdlist1.plot(50, cumulative=True)
>>>
```



nltk: Stylometry: word length distribution

```
len1s = [len1[i*10000:i*10000+10000] for i in range(10)]  
for l in len1s:  
    plt.hist(l, bins=np.arange(min(l),max(l)+1), histtype='step')  
plt.show()
```

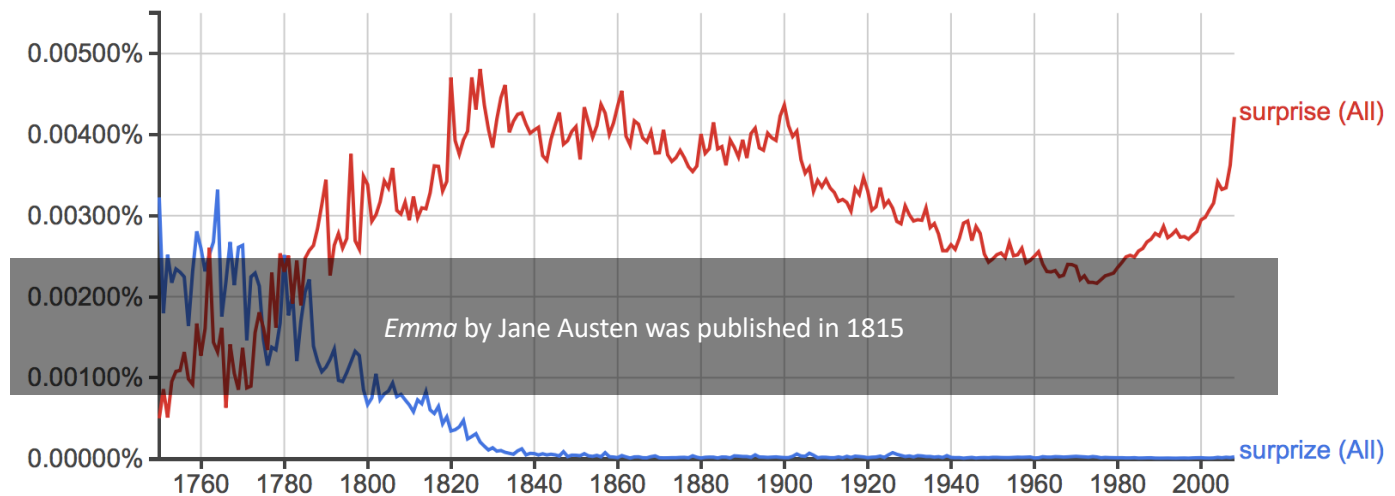
Forensic linguistics



Google:
relative
frequency of
two spellings

Google Books Ngram Viewer

Graph these comma-separated phrases: case-insensitive
between and from the corpus with smoothing of . [Search lots of books](#)



nltk: Concordance

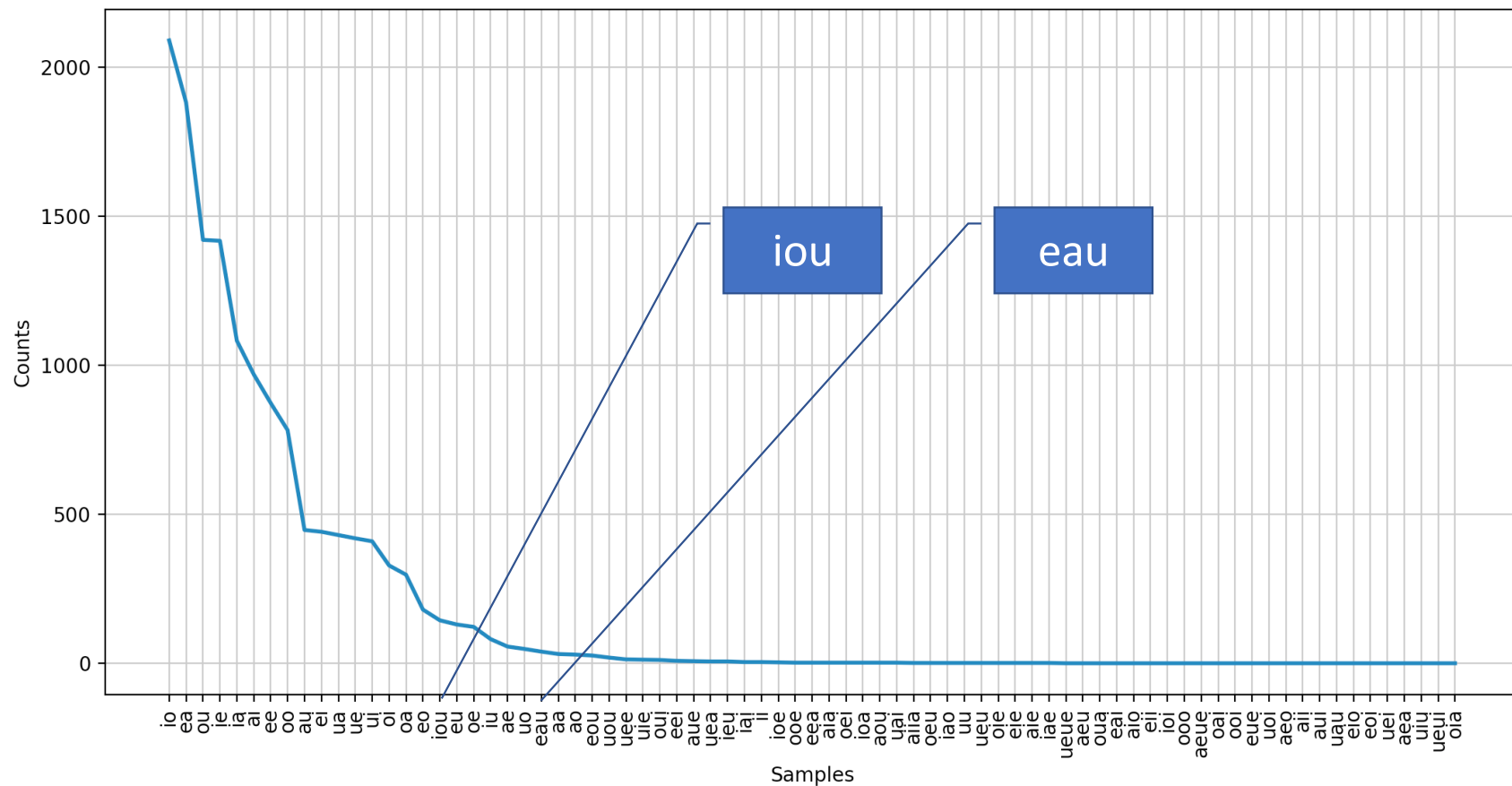
```
>>> import nltk
>>> emma =
nltk.Text(nltk.corpus.gutenberg
rg.words('austen-emma.txt'))
>>>
emma.concordance("surprize")
Displaying 25 of 37 matches:
```

```
er father , was sometimes taken by surprize at his being still able to pity `
hem do the other any good ." " You surprize me ! Emma must do Harriet good : a
Knightley actually looked red with surprize and displeasure , as he stood up ,
r . Elton , and found to his great surprize , that Mr . Elton was actually on
d aid ." Emma saw Mrs . Weston ' s surprize , and felt that it must be great ,
father was quite taken up with the surprize of so sudden a journey , and his f
y , in all the favouring warmth of surprize and conjecture . She was , moreove
he appeared , to have her share of surprize , introduction , and pleasure . Th
ir plans ; and it was an agreeable surprize to her , therefore , to perceive t
talking aunt had taken me quite by surprize , it must have been the death of m
f all the dialogue which ensued of surprize , and inquiry , and congratulation
the present . They might chuse to surprize her ." Mrs . Cole had many to agre
the mode of it , the mystery , the surprize , is more like a young woman ' s s
to her song took her agreeably by surprize -- a second , slightly but correct
" " Oh ! no -- there is nothing to surprize one at all .-- A pretty fortune ;
t to be considered . Emma ' s only surprize was that Jane Fairfax should accep
of your admiration may take you by surprize some day or other ." Mr . Knightle
ation for her will ever take me by surprize .-- I never had a thought of her i
expected by the best judges , for surprize -- but there was great joy . Mr .
sound of at first , without great surprize . " So unreasonably early !" she w
d Frank Churchill , with a look of surprize and displeasure .-- " That is easy
; and Emma could imagine with what surprize and mortification she must be retu
tled that Jane should go . Quite a surprize to me ! I had not the least idea !
. It is impossible to express our surprize . He came to speak to his father o
g engaged !" Emma even jumped with surprize ;-- and , horror - struck , exclai
```

```
>>> >>> emma.concordance("surprise")
Displaying 1 of 1 matches:
```


```
that Emma could not but feel some surprise , and a little displeasure , on he
```


nltk: Counting frequency of occurrences of sequences of vowels in English



Virginia Woolf

- Literary Style: *Stream of consciousness*
 - we look at using nltk to explore this



stream of con·scious·ness

noun **PSYCHOLOGY**

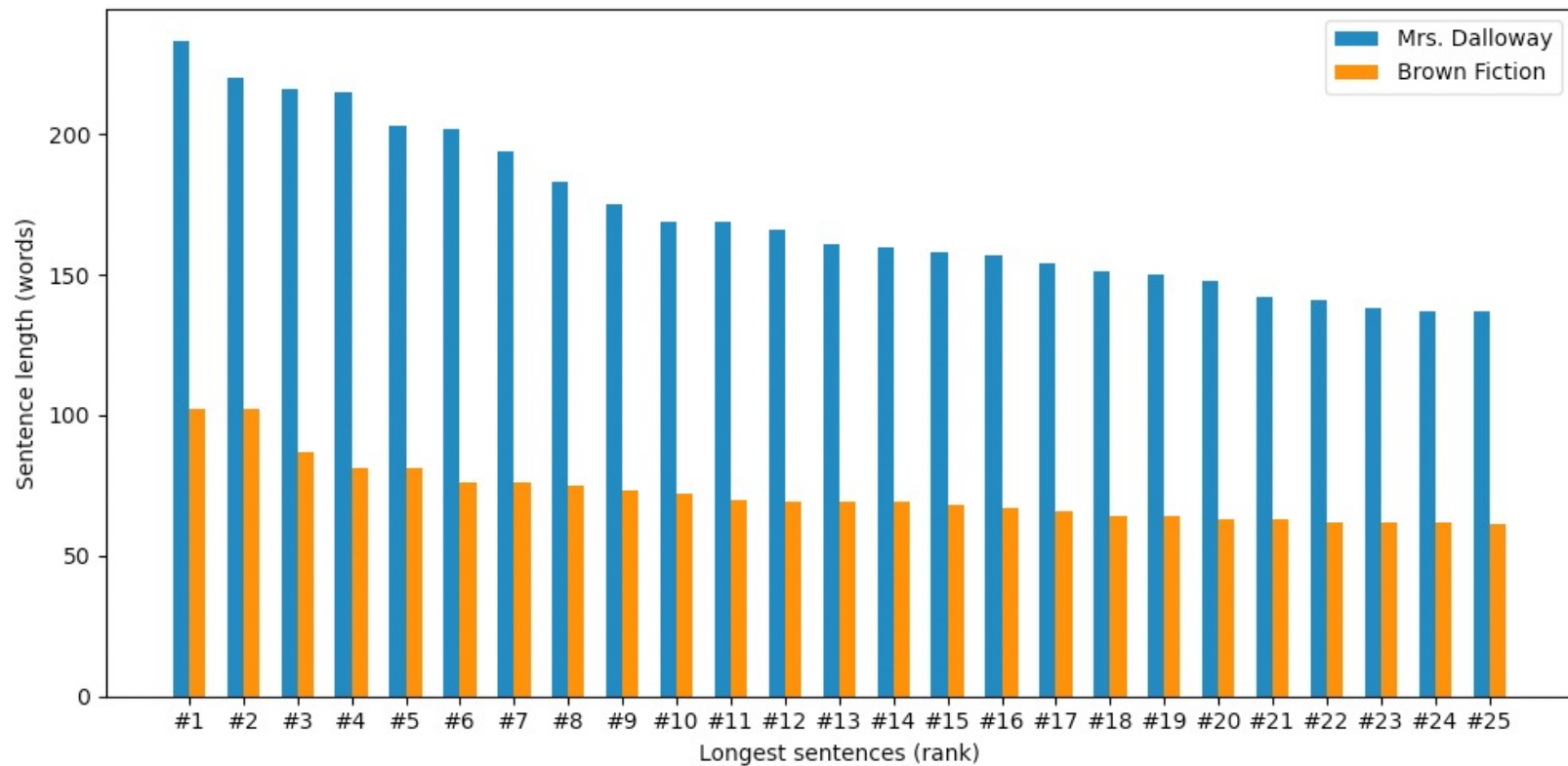
a person's thoughts and conscious reactions to events, perceived as a continuous flow. The term was introduced by William James in his *Principles of Psychology* (1890).

- a literary style in which a character's thoughts, feelings, and reactions are depicted in a continuous flow uninterrupted by objective description or conventional dialogue. James Joyce, Virginia Woolf, and Marcel Proust are among its notable early exponents.

"a stream-of-consciousness monologue"

'How fresh, how calm, stiller than this of course, the air was in the early morning; like the flap of a wave; the kiss of a wave; chill and sharp and yet (for a girl of eighteen as she then was) solemn, feeling as she did, standing there at the open window, that something awful was about to happen; looking at the flowers, at the trees with the smoke winding off them and the rooks rising, falling; standing and looking until Peter Walsh said, "Musing among the vegetables?"'

Mrs. Dalloway vs. Brown Corpus Fiction Top25



Famous Buffalo sentence



- considered grammatical, and
- makes sense
- but hard to parse even for native speakers

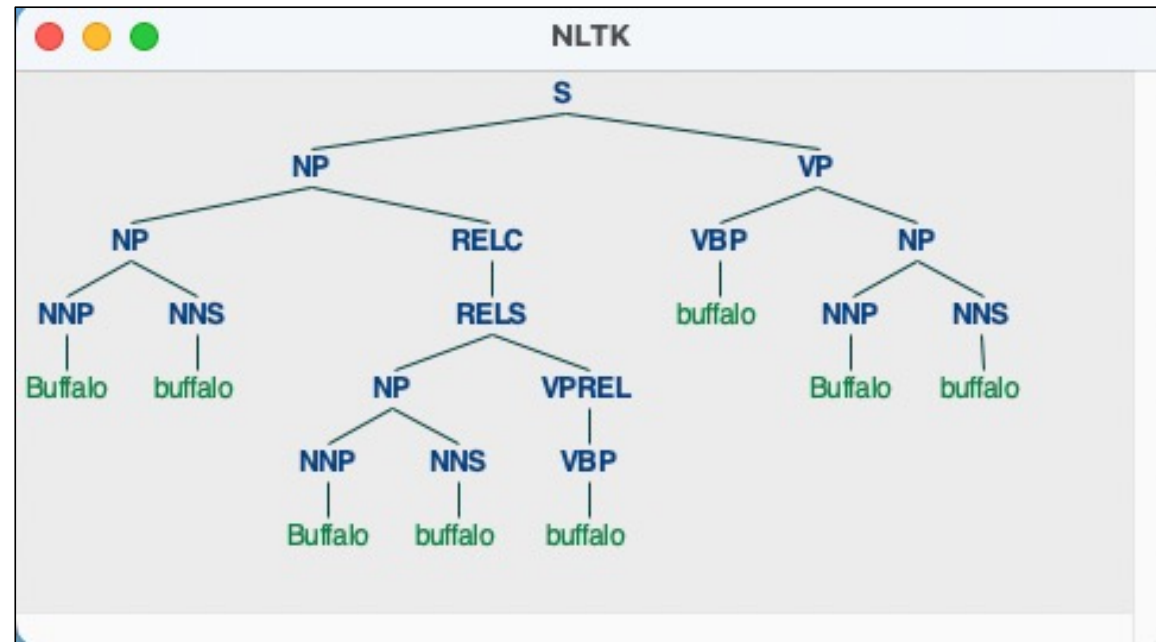
- 8 consecutive occurrences of the word *buffalo*

picture borrowed from Analytical Grammar/Grammar Planet on Facebook, who borrowed it from somewhere else...

Context Free Grammar Parsing

buffalo.txt

```
1 S -> NP VP |
2 NP -> NNP NNS |
3 VP -> VBP NP |
4 VP -> VBD NP |
5 NNP -> 'Buffalo' |
6 NNS -> 'buffalo' |
7 VBP -> 'buffalo' |
8 VBD -> 'buffaloed' |
9 NP -> NP RELC |
10 RELC -> REL RELS |
11 RELC -> RELS |
12 REL -> 'that' |
13 RELS -> NP VPREL |
14 VPREL -> VBP |
15 VPREL -> VBD |
```

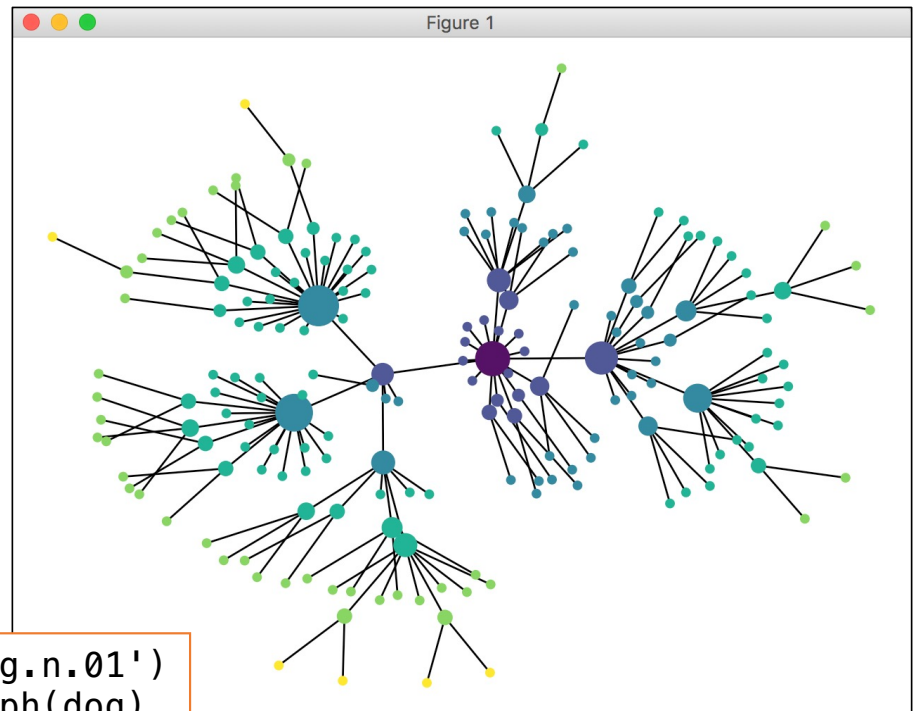


nltk: WordNet relations: types of dogs

graph.py on course website

```
1 import networkx as nx
2 from networkx.drawing.nx_agraph import graphviz_layout
3 import matplotlib.pyplot as plt
4 from nltk.corpus import wordnet as wn
5
6 def traverse(graph, start, node):
7     graph.depth[node.name] = node.shortest_path_distance(start)
8     for child in node.hyponyms():
9         graph.add_edge(node.name, child.name)
10        traverse(graph, start, child)
11
12 def hyponym_graph(start):
13     G = nx.Graph()
14     G.depth = {}
15     traverse(G, start, start)
16     return G
17
18 def graph_draw(graph):
19     nx.draw(graph, pos=graphviz_layout(graph), node_size = [16 * graph.degree
20     (n) for n in graph], node_color = [graph.depth[n] for n in graph], with_label...
21     s = False)
22     plt.show()
```

```
dog = wn.synset('dog.n.01')
graph = hyponym_graph(dog)
graph_draw(graph)
```



nltk: WordNet relations: parts of a car

```
from nltk.corpus import wordnet as wn
c = wn.synset('car.n.01')
g = graph(c, 'part_meronyms')
graph_draw(g)
```

