

# Introduction to the Classical Language Toolkit (CLTK)

Kyle P. Johnson, PhD

[kyle@kyle-p-johnson.com](mailto:kyle@kyle-p-johnson.com)

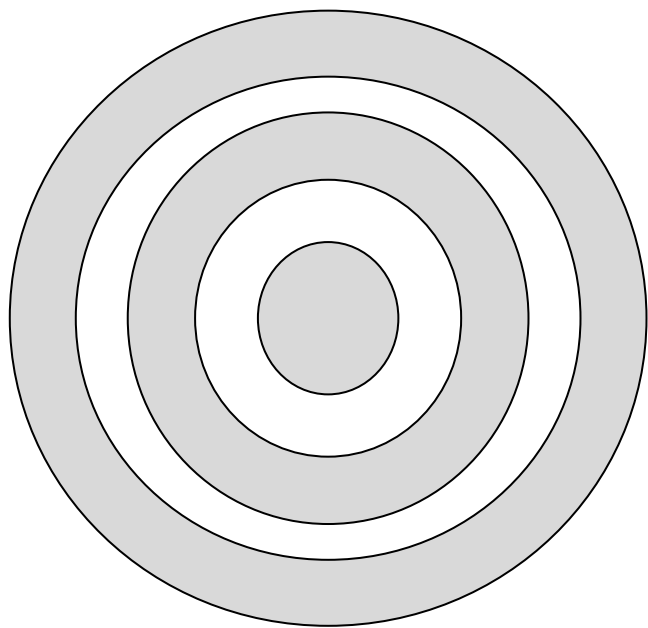
<http://cltk.org>

NYU, Classics Dept.

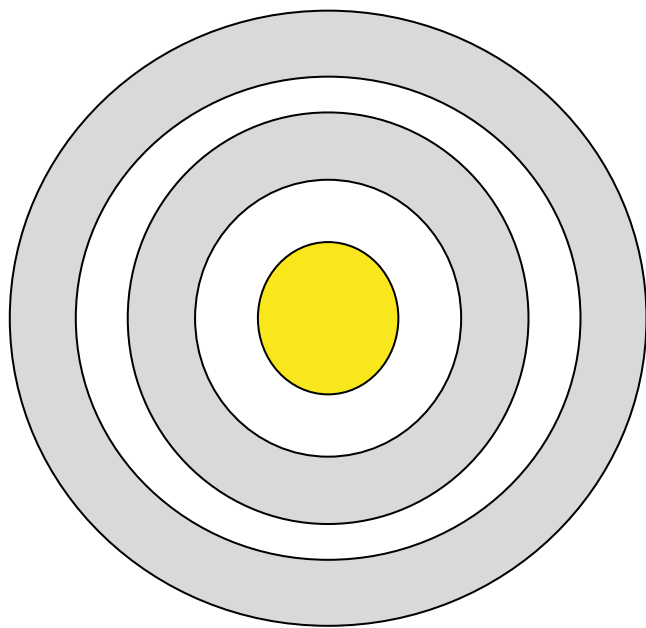
Nov 17, 2015



# The CLTK's goals ...

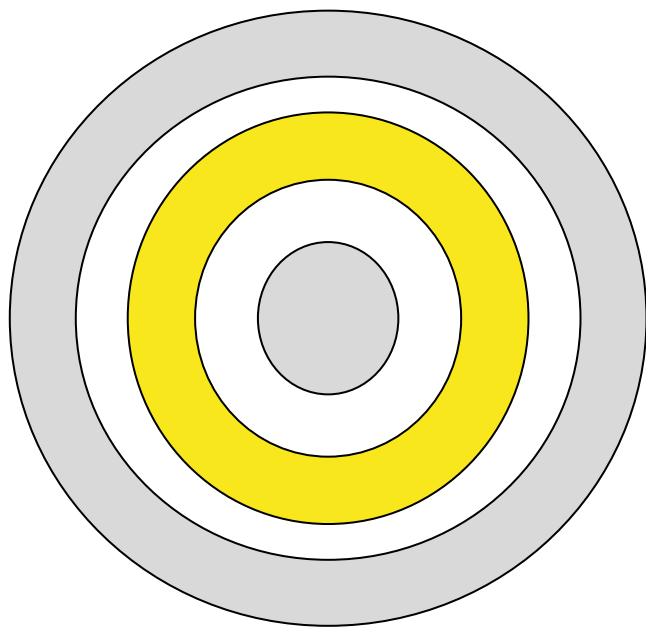


# The CLTK's goals ...



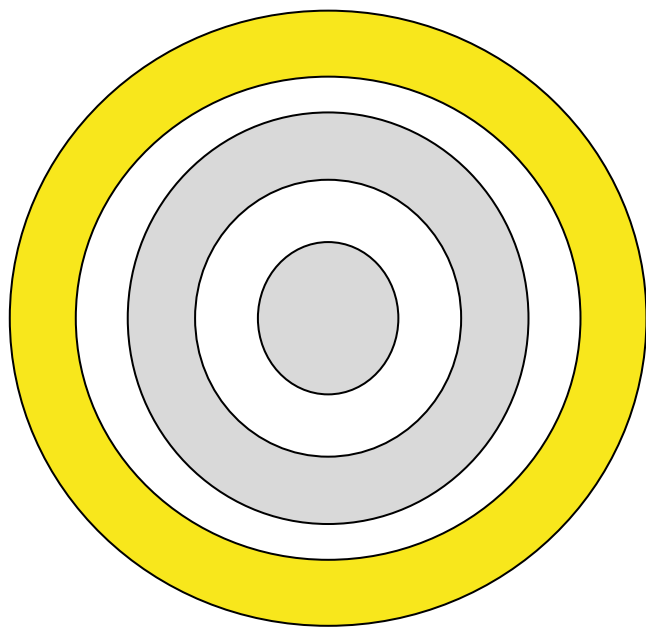
- Low: Good datasets for advanced NLP of Ancient Greek and Latin languages

# The CLTK's goals ...



- Low: Good libraries for advanced NLP of Ancient Greek and Latin languages
- Medium: Quantified Classics

# The CLTK's goals ...



- Low: Good libraries for advanced NLP of Ancient Greek and Latin languages
- Medium: Quantified Classics
- High: Framework for an integrated study of ancient literature

# Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness

# Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing

# Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing
- Package or “library”: Collection of software for a particular set of tasks



# Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing
- Package or “library”: Collection of software for a particular set of tasks
- NLTK: A prominent NLP package for the Python language

# Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing
- Package or “library”: Collection of software for a particular set of tasks
- NLTK: A prominent NLP package for the Python language
- Git: Software for distributed software development

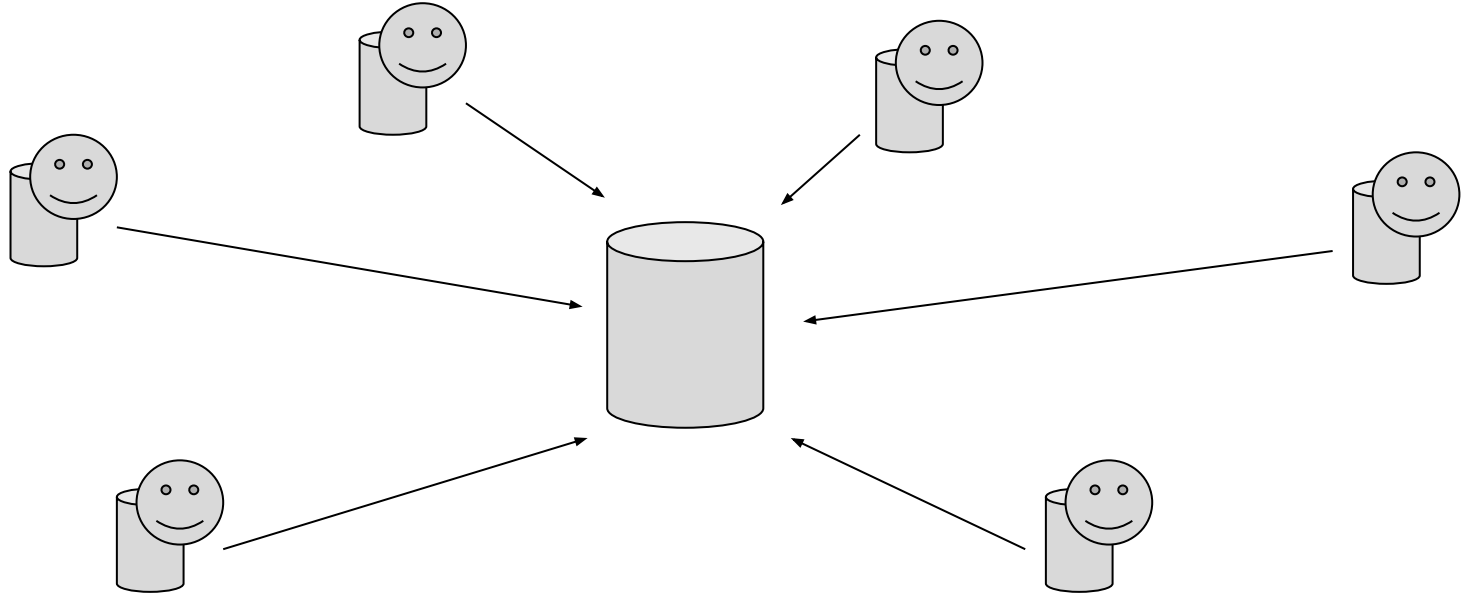
# Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing
- Package or “library”: Collection of software for a particular set of tasks
- NLTK: A prominent NLP package for the Python language
- Git: Software for distributed software development
- GitHub: A website which makes Git easy

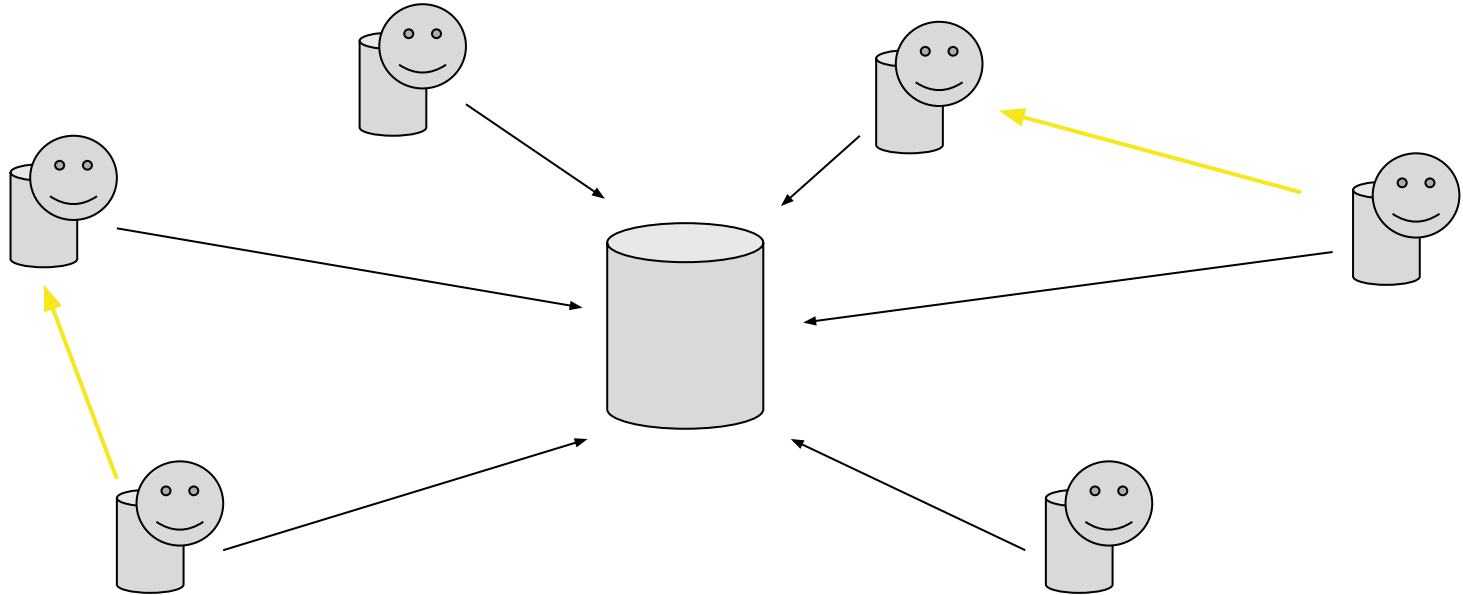
# Some basic terms

- Python: A programming language known for its easy-to-read syntax and general friendliness
- NLP: Natural language processing
- Package or “library”: Collection of software for a particular set of tasks
- NLTK: A prominent NLP package for the Python language
- Git: Software for distributed software development
- GitHub: A website which makes Git easy
- Jupyter (formerly IPython): “Scientific notebooks”, an easy way to share code

# Technical organization: Repositories



# Technical organization: Repositories



# Technical organization: Core vs. Data

- CLTK Core software
  - Led by programmers
  - Coordinates data processing
  - Downloads and installs data repositories

# Technical organization: Core vs. Data

- CLTK Core software
  - Led by programmers
  - Coordinates data processing
  - Downloads and installs data repositories
- Linguistic data repositories
  - Led by language experts
  - Plaintext corpora
  - Trained models (for machine learning)
  - Dictionaries, word lists
  - Tagged texts (for part-of-speech, dependency grammar)



# Personnel organization: Teams

- CLTK organization on GitHub (forthcoming)
  - Admins
  - Contributors
  - Unaffiliated contributors

# Personnel organization: Teams (cont'd.)

The screenshot shows the GitHub Teams interface for the 'Classical Language' organization. At the top, there's a search bar labeled 'Find a team...' and two buttons: 'My teams' and '+ New team'. Below this, there are five team cards arranged in two columns. Each card has a title, a 'SECRET' label, member and repository counts, a profile picture, and a 'Leave' button.

Team Name	Label	Members	Repositories
Owners	SECRET	1 member	27 repositories
Core Admins	SECRET	1 member	1 repository
Core Contributors	SECRET	1 member	0 repositories
Greek & Latin Admins	SECRET	1 member	12 repositories
Greek & Latin Contributors	SECRET	1 member	12 repositories

# (Really quick) quickstart

- Make virtualenv and download core

- `$ pyvenv venv`
- `$ source venv/bin/activate`
- `$ pip install cltk`

- Download and import corpora

- `$ python`
- `>>> from cltk.corpus.utils.importer import CorpusImporter`
- `>>> ci = CorpusImporter('greek')`
- `>>> ci.list_corpora`
- `['greek_software_tlg', 'greek_text_perseus', 'phi7', 'tlg', 'greek_proper_names_cltk', 'greek_models_cltk', 'greek_treebank_perseus', 'greek_lexica_perseus', 'greek_training_set_sentence_cltk', 'greek_word2vec_cltk']`
- `>>> ci.import_corpus('greek_text_perseus')`

# Setup for PHI and TLG corpora

- PHI5, PHI7, and TLG\_E
  - Not downloaded, but imported from local files
  - `>>> ci.import_corpus('tlg', '~/Documents/corpora/TLG_E/')`
  - Makes copy of corpus at `~/cltk_data/originals`
- Convert TLG from Beta Code into Unicode
  - `>>> from cltk.corpus.greek.tlgu import TLGU`
  - `>>> t = TLGU()`
  - `>>> t.convert_corpus(corpus='tlg')`
  - `>>> t.convert_corpus(corpus='phi5')`
  - Makes copy of corpus in `~/cltk_data/greek/text/tlg` or `~/cltk_data/latin/text/phi5`

# NLP for all languages

- Concordance
- Information retrieval
  - Plain and regex searching
  - Robust boolean search on the way
- n-gram: 'Ut primum nocte discussa sol'
  - bigrams: ('ut', 'primum'), ('primum', 'nocte'), ('nocte', 'discussa'), ('discussa', 'sol')
  - trigrams: ('ut', 'primum', 'nocte'), ('primum', 'nocte', 'discussa'), ('nocte', 'discussa', 'sol')
  - 5-grams: ('ut', 'primum', 'nocte', 'discussa', 'sol')
- Word frequencies
  - simple count for a word
  - complete reports for a text
- Word tokenization (via NLTK)

# NLP for Greek and Latin

- Text normalization
  - j » i, v » u (Latin)
  - Beta Code conversion (for legacy Greek texts)
  - TLG and PHI corpus specific (remove formatting)
- Sentence tokenizer
- Lemmatizer
- Stemmer (Latin)
- Word tokenizer, for enclitics (Latin)
- Stopword filtering

# NLP for Greek and Latin (cont'd.)

- Named Entity Recognition (NER)
- Part-of-speech (POS) tagger
  - From Perseus/Alphaeus treebank
  - Great work remaining to be done, convert their codes to others (Brill, PROIEL, etc)
- # TODO! Dependency grammar tagger
- Prosody scanner
- Syllabifier (Greek)
- TLG and PHI5 indices
  - File to author, genre to authors, date to authors, gender to authors, etc.
- Word2Vec

# Beyond Greek and Latin

- Chinese, Coptic, Pali, Tibetan
  - 2.5 GB (!) of Chinese Buddhist texts
  - Corpus of Coptic texts
  - Pali Tipitaka
  - Tibetan POS tagged texts and a lexicon



# Beyond Greek and Latin

- Chinese, Coptic, Pali, Tibetan
  - 2.5 GB (!) of Chinese Buddhist texts
  - Corpus of Coptic texts
  - Pali Tipitaka
  - POS tagged texts and a lexicon
- Growth areas
  - Many more ancient resources to be discovered, normalized, and incorporated into the CLTK ecosystem.
  - Great opportunities for outreach
    - To departments, disciplines, countries, and traditions
    - See [List of Classical languages](#)
  - Follow the Greek and Latin code patterns to add support any language!

# Citation

- Developed by [many talented contributors!](#)
- BibTex

```
○ @Misc{johnson2014,  
  author = {Kyle P. Johnson et al.},  
  title = {CLTK: The Classical Language Toolkit},  
  howpublished = {\url{https://github.com/kylepjohnson/cltk}},  
  note = {{DOI} 10.5281/zenodo.<current_release_id>},  
  year = {2014--2015},  
}
```

- Chicago author-date

```
○ Kyle P. Johnson et al.. (2014-2015). CLTK: The Classical Language Toolkit. DOI 10.5281/zenodo.  
  <current_release_id>
```

*Note: Current DOI release id available at: <https://github.com/kylepjohnson/cltk>*

# Resources

- [This lecture's code](#)
- Core software: <https://github.com/cltk/cltk>
  - Bug tracking: <https://github.com/cltk/cltk/issues>
    - Beginners' issues labeled **easy**
- Project repositories: <https://github.com/cltk>
- Docs: <http://docs.cltk.org>
  - Installation: <http://docs.cltk.org/en/latest/installation.html>
- Python + Command line basics
  - Intro to the command line: <http://blog.teamtreehouse.com/introduction-to-the-mac-os-x-command-line>
  - Python installation: <https://www.python.org/downloads/> (choose 3.5)
  - Good self-paced Python lessons: <http://learnpythonthehardway.org/>