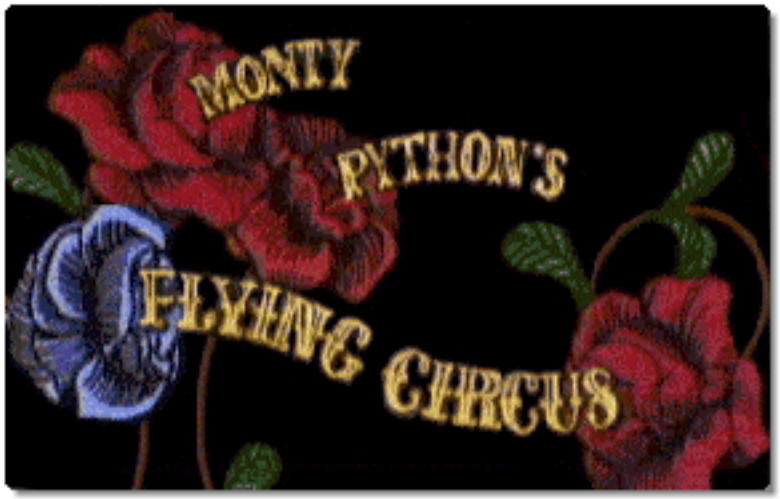
Natural Language ToolKit (NLTK) and Python

Using NLTK in NLP

- NL ToolKit provides libraries of many of the common NLP processes at various language levels
  - Leverage these libraries to process text
- Goal is to learn about and understand how NLP can be used to process text without programming all processes
  - However, some programming is required to
    - Call libraries
    - Process data
    - Customize NLP processes
  - Programming language is Python

Python and NLP

- Python is freely available for many platforms from the Python Software Foundation:
  - http://www.python.org/
  - Named for the group Monty Python



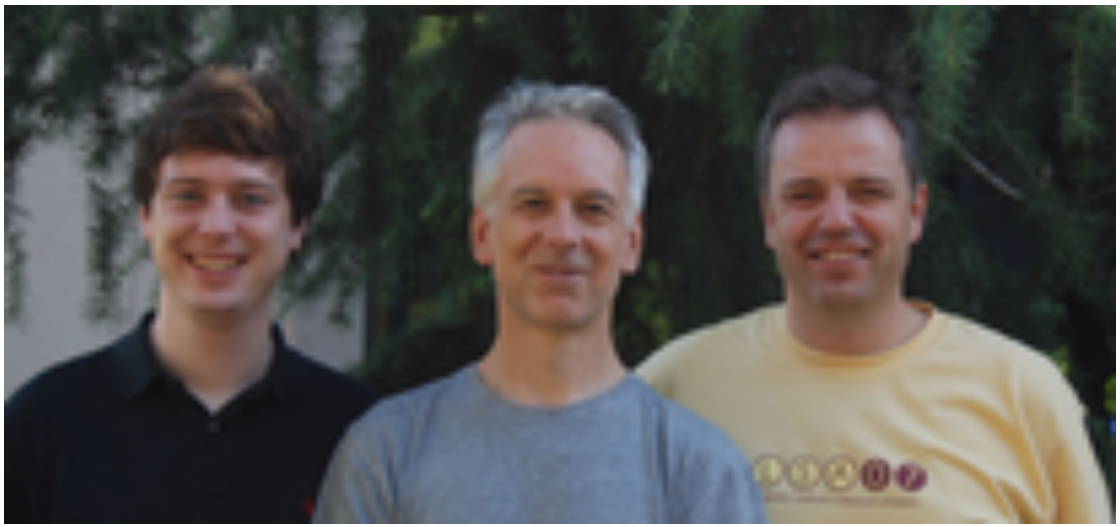
The group in 1969

Characteristics of Python

- Easy-to-learn scripting language, similar in many aspects to Perl
  - But with WYSIWYG block structure
- Object-oriented, with modules, classes, exceptions, high-level dynamic data types, similar to Java
- Strongly typed, but without type declarations (dynamic typing)
- Regular Expressions and other string processing features

Natural Language Toolkit (NLTK)

- A suite of Python libraries for symbolic and statistical natural language programming
  - Developed at the University of Pennsylvania
- Developed to be a teaching tool and a platform for research NLP prototypes
  - Data types are packaged as classes
  - Goal of code is to be clear, rather than fastest performance
- Online book: http://www.nltk.org/book/
  - Authors: Edward Loper, Ewan Kline and Steven Bird



Getting Started in Python

- Python can be run as an interactive system
  - Type in expressions or small pieces of programs to try them out
- or as a command-line system.
  - Run stored python programs
- For both, it is recommended to use IDLE, the Python development environment
  - Especially good to edit Python programs in IDLE to keep track of the indentation for block structure

Introduction to NLTK

- NLTK provides:
  - Basic classes for representing data relevant to Natural Language Processing.
  - Standard interfaces for performing NLP tasks such as tokenization, tagging and parsing
  - Standard implementation of each task which can be combined to solve complex problems

NLTK Modules

- **corpora:** a package containing modules of example text
- **tokenize:** functions to separate text strings
- **probability**: for modeling frequency distributions and probabilistic systems
- **stem –** package of functions to stem words of text
- **wordnet –** interface to the WordNet lexical resource
- **chunk –** identify short non-nested phrases in text
- **etree**: for hierarchical structure over text
- **tag**: tagging each word with part-of-speech, sense, etc.
- **parse**: building trees over text
  - recursive descent, shift-reduce, probabilistic, etc.
- **cluster**: clustering algorithms
- **draw**: visualize NLP structures and processes
- **contrib:** various pieces of software from outside contributors

Tutorials for Python and NLTK

- Python

  http://docs.python.org/tut/tut.html, the classic by Guido van Rossum

- NLTK is a SourceForge project at: http://www.nltk.org

  documentation: http://www.nltk.org/documentation, including

  book: http://www.nltk.org/book

  API: http://nltk.googlecode.com/svn/trunk/doc/api/index.html