

# NLTK

Natural Language ToolKit  
([www.nltk.org](http://www.nltk.org))

## What is nltk?

- A tool which allows you to do NLP such as parsing.
- We'll be using nltk to write custom CFGs for a set of given sentences and produce parse trees for those sentences to check the accuracy of the grammar.

## Download and Install (On your local machine)

- <http://www.nltk.org/download>
- windows
- Linux
- Mac

## NLTK is installed on the cs linux machines

```
rhenium{106} setenv PYTHONPATH /usr/local/lib64
rhenium{107} echo $PYTHONPATH
/usr/local/lib64
rhenium{108} python
Python 2.4.3 (#1, Sep 3 2009, 15:37:37)
[GCC 4.1.2 20080704 (Red Hat 4.1.2-46)] on linux2
Type "help", "copyright", "credits" or "license" for more
information.
>>> import nltk
>>>
```

- Slight variations depending on your shell

# Bringing up the GUI for Chart

The screenshot shows a Windows desktop with two windows. On the left is a Windows PowerShell (V2) terminal window. On the right is the Chart Parser Application GUI.

**PowerShell Terminal Output:**

```

PS C:\Documents and Settings\Kaushal> python
Python 2.5.4 (r25415088, Sep 19 2008, 09:52:17) [MSC v.1
Type "help", "copyright", "credits" or "license" for m
>>> import nltk
>>> nltk.app.chartparser()
C:\Python25\Lib\site-packages\nltk\app\__init__.py:46:
Warning: warn C:\nltk.app.wordfreq not loaded "
grammar: <
S -> NP VP
VP -> VP PP
VP -> U NP
VP -> U
NP -> Det N
NP -> NP PP
PP -> P NP
NP -> 'John'
NP -> 'I'
Det -> 'the'
Det -> 'a'
Det -> 'a'
N -> 'dog'
N -> 'cookie'
N -> 'table'
N -> 'cake'
N -> 'fork'
U -> 'ate'
U -> 'saw'
P -> 'on'
P -> 'under'
P -> 'with'
tokens = ['John', 'ate', 'the', 'cake', 'on', 'the', 't
Calling "ChartParserApp(grammar, tokens)..."
  
```

**Chart Parser Application GUI:**

The GUI window has a menu bar (File, Edit, View, Apply, Animate, Zoom, Help) and a toolbar (Edit Grammar, Edit Tree, Data). The main area contains a grid with columns labeled 'John', 'ate', 'the', 'cake', 'on', 'the', 'table' and rows labeled 0 through 7. Below the grid, there are several buttons for parsing strategies: Top Down Init Rule, Top Down Predict Rule, Bottom Up Predict Rule, Bottom Up Left-Corner Predict Rule, Fundamental Rule, Top Down Strategy, Bottom Strategy, Bottom Up Left-Corner Strategy, and Reset Parser. A status bar at the bottom indicates "Last edge generated by:" and "Step".

# Loading the grammar

The screenshot shows the Chart Parser Application GUI with the CFG Editor window open. The CFG Editor displays the following grammar rules:

```

Productions:
S => NP VP
VP => VP PP | NP IV
NP => Det N | NP PP
PP => P NP
NP => 'John' | 'I'
Det => 'the' | 'a' | 'a'
N => 'dog' | 'cookie' | 'table' | 'cake' | 'fork'
V => 'ate' | 'saw'
P => 'on' | 'under' | 'with'
  
```

The GUI window shows the same grid as in the previous screenshot, but now with the CFG Editor window open on top of it. The GUI window also shows the "Last edge generated by:" and "Step" status bar.

## Stepping through the parse

- To view the parse step by step, check the “Step” checkbox in the lower right corner.
- To view the final parse without stepping through, uncheck “Step”