In [36]:
```
import nltk
import tagutils; reload(tagutils)
from tagutils import *
from IPython.core.display import HTML
```

Tagged Corpora

In [37]:
```
from nltk.corpus import brown
```

Remember that we were working with manually tagged corpora. The tags are awfully hard to visualize by themselves, so I'll use a simple color coding scheme to display them a little more clearly.

In [38]:
```
stags([("World","NN")])
```

Out[38]:
```
"<font color='blue' size=+1>World</font>"
```

In [39]:
```
HTML(stags(brown.tagged_sents()[100]))
```

Out[39]:
Daniel personally led the fight for the measure , which he had watered down considerably since its rejection by two previous Legislatures , in a public hearing before the House Committee on Revenue and Taxation .

Some Test Sentences

Let's create some untagged sample sentences

In [40]:
```
text = """
John loves Jane.

Peter and John both like Mozart.

Peter speaks like a politician.

Banana flies like fruit.

Some flies like banana.

Think different.

Think differently.

John likes ice cream and Peter likes to sing.

The man harvested a ripe coconut with a bush knife.

On the basis of the physical interpretation of distance which has been
indicated, we are also in a position to establish the distance between
two points on a rigid body by means of measurements.

We will not speak of all Queequeg's peculiarities here; how he eschewed
coffee and hot rolls, and applied his undivided attention to beefsteaks, done rare.

Voters in Catalonia delivered victory to separatist parties
in a regional election on Sunday, raising the likelihood that Spain's
most powerful economic region will hold an independence referendum that
Madrid has vowed to block.
"""
```

In [41]:
```
misc_sents = [nltk.word_tokenize(s) for s in nltk.sent_tokenize(text)]
from nltk.corpus import gutenberg
alice_sents = gutenberg.sents("carroll-alice.txt")[130:140]
sents = misc_sents+alice_sents
sents[3]
```

Out[41]:
```
['Banana', 'flies', 'like', 'fruit', '.']
```

The Default Tagger

In [42]:
```
dtagged = [nltk.pos_tag(s) for s in sents]
```

In [43]:
```
HTML("<p>\n".join(["[%d] "%i + stags(s) for i,s in enumerate(dtagged)]))
```

Out[43]:

[0] John loves Jane .

[1] Peter and John both like Mozart .

[2] Peter speaks like a politician .

[3] Banana flies like fruit .

[4] Some flies like banana .

[5] Think different .

[6] Think differently .

[7] John likes ice cream and Peter likes to sing .

[8] The man harvested a ripe coconut with a bush knife .

[9] On the basis of the physical interpretation of distance which has been indicated , we are also in a position to establish the distance between two points on a rigid body by means of measurements .

[10] We will not speak of all Queequeg 's peculiarities here ; how he eschewed coffee and hot rolls , and applied his undivided attention to beefsteaks , done rare .

[11] Voters in Catalonia delivered victory to separatist parties in a regional election on Sunday , raising the likelihood that Spain 's most powerful economic region will hold an independence referendum that Madrid has vowed to block .

[12] Alice took up the fan and gloves , and , as the hall was very hot , she kept fanning herself all the time she went on talking : ' Dear , dear !

[13] How queer everything is to - day !

[14] And yesterday things went on just as usual .

[15] I wonder if I ' ve been changed in the night ?

[16] Let me think : was I the same when I got up this morning ?

[17] I almost think I can remember feeling a little different .

[18] But if I ' m not the same , the next question is , Who in the world am I ?

[19] Ah , THAT ' S the great puzzle !

[20] ' And she began thinking over all the children she knew that were of the same

The TnT Tagger in Python

In [44]:
```
import nltk.tag
import os
import cPickle
```

Ordinarily, taggers are *trained* on tagged data. Here is an example of how to do this.

The TnT tagger is a tagger based on n-grams. We will see later how that works.

Here you see how the tagger is trained using tagged sentences and the `train` method. The trained model is written to disk with `cPickle.dump`.

If a dumped, trained model already exists, the time consuming training step is skipped and the saved model is loaded instead.

In [45]:
```
tntfile = "brown.tnt.model"
if not os.path.exists(tntfile):
    print "# training and writing",tntfile
    tnt = nltk.tag.TnT()
    tnt.train(brown.tagged_sents())
    with open(tntfile,"wb") as stream:
        cPickle.dump(tnt,stream)
else:
    print "# loading",tntfile
    with open(tntfile,"rb") as stream:
        tnt = cPickle.load(stream)
```

```
# loading brown.tnt.model
```

In [46]:
```
tnt.tag(sents[2])
```

Out[46]:
```
[('Peter', 'NP'),
 ('speaks', 'VBZ'),
 ('like', 'CS'),
 ('a', 'AT'),
 ('politician', 'NN'),
 ('.', '.')]
```

In [47]:
```
HTML(stags(tnt.tag(sents[2])))
```

Out[47]: Peter speaks like a politician .

In [48]:
```
alltagged = [tnt.tag(s) for s in sents]
```

In [49]:
```
HTML("<p>\n".join(["[%d] "%i + stags(s) for i,s in enumerate(alltagged)]))
```

Out[49]:

[0] John loves Jane .

[1] Peter and John both like Mozart .

[2] Peter speaks like a politician .

[3] *[Banana]* flies like fruit .

[4] Some flies like banana .

[5] Think different .

[6] Think differently .

[7] John likes ice cream and Peter likes to sing .

[8] The man harvested a ripe coconut with a bush knife .

[9] On the basis of the physical interpretation of distance which has been indicated , we are also in a position to establish the distance between two points on a rigid body by means of measurements .

[10] We will not speak of all *[Queequeg] ['s]* peculiarities here ; how he eschewed coffee and hot rolls , and applied his undivided attention to *[beefsteaks]* , done rare .

[11] Voters in *[Catalonia]* delivered victory to *[separatist]* parties in a regional election on Sunday , raising the likelihood that Spain *['s]* most powerful economic region will hold an independence referendum that Madrid has vowed to block .

[12] Alice took up the fan and gloves , and , as the hall was very hot , she kept fanning herself all the time she went on talking : ' Dear , dear !

[13] How queer everything is to - day !

[14] And yesterday things went on just as usual .

[15] I wonder if I ' *[ve]* been changed in the night ?

[16] Let me think : was I the same when I got up this morning ?

[17] I almost think I can remember feeling a little different .

[18] But if I ' *[m]* not the same , the next question is , Who in the world am I ?

[19] Ah , *[THAT]* ' S the great puzzle !

[20] ' And she began thinking over all the children she knew that were of the same age as

In [50]:
```
alltagged[13]
```

Out[50]:
```
[('How', 'WRB'),
 ('queer', 'JJ'),
 ('everything', 'PN'),
 ('is', 'BEZ'),
 ('to', 'TO'),
 ('-', 'IN'),
 ('day', 'NN'),
 ('!', '.')]
```

More Taggers

NLTK provides a common, simple interface to a number of different taggers.

Brill Tagger

A *Brill Tagger* takes:

- an existing tagger
- a set of tagged sentences

It uses the existing tagger and attempts to learn transformational rules that improve the performance of the existing tagger.

HunposTagger

The *HunposTagger* is an HMM-based tagger, a reimplementation of the TnT tagger. It is written in OCAML; NLTK contains a Python interface.

Mallet Tagger

Mallet is a machine learning toolkit for statistical natural language processing written in Java. The Mallet tagger is based on CRF's. We may be able to cover CRF's later in the class.