TagCloud2: Build a tag cloud of a 2012 presidential debate. (Python3) (http://www.cse.msu.edu/~cse231/PracticeOfComputingUsingPython/06_Dictionaries/TagCloud2/)

In [1]:

```
# Functions adapted from ProgrammingHistorian (updated to Python3)
# http://niche.uwo.ca/programming-historian/index.php/Tag_clouds

# Take one long string of words and put them in an HTML box.
# If desired, width, background color & border can be changed in the functio
n
# This function stuffs the "body" string into the the HTML formatting string
.
def make_HTML_box(body):
    '''Required -- body (string), a string of words
       Return -- a string that specifies an HTML box containing the body
    '''
    box_str = """<div style=\"
    width: 640px;
    background-color: rgb(250,250,250);
    border: 1px grey solid;
    text-align: center\" >{:s}</div>
    """
    return box_str.format(body)

def make_HTML_word(word,cnt,high,low):
    ''' make a word with a font size to be placed in the box. Font size is s
caled
    between high and low (to be user set). high and low represent the high
    and low counts in the document. cnt is the count of the word
    Required -- word (string) to be formatted
             -- cnt (int) count of occurances of word
             -- high (int) highest word count in the document
             -- low (int) lowest word count in the document
    Return -- a string formatted for HTML that is scaled with respect to cnt
'''
    ratio = (cnt-low)/float(high-low)
    font_size = high*ratio + (1-ratio)*low
    font_size = int(font_size)
    word_str = '<span style=\"font-size:{:s}px;\">{:s}</span>'
    return word_str.format(str(font_size), word)

def print_HTML_file(body,title):
    ''' create a standard html page (file) with titles, header etc.
    and add the body (an html box) to that page. File created is title+'.htm
l'
    Required -- body (string), a string that specifies an HTML box
    Return -- nothing'''
    fd = open(title+'.html','w')
    the_str="""
    <html> <head>
    <title>"""+title+"""</title>
    </head>

    <body>
    <h1>"""+title+'</h1>'+'\n'+body+'\n'+"""<hr>
    </body> </html>
    """
    fd.write(the_str)
    fd.close()
```

In [2]:

```
import re
import nltk
from IPython.core.display import display, HTML

#from htmlFunctions import *

# Load files
debates_f = ["debate.txt","debateTWO.txt"]
stop_words_f = "stopWords.txt"

debates = []
stop_words = []

for file in debates_f:
    with open(file) as f:
        data = f.read().split("\n")
        debates.append(data)
with open(stop_words_f) as f:
    stop_words = f.read().split("\n")

# Check correct loading
for d in debates:
    print(len(d))
print(len(stop_words))
```

```
880
917
591
```

In [3]:

```
speaker1 = ["PRESIDENT BARACK OBAMA", "PRESIDENT OBAMA"]
speaker2 = ["MITT ROMNEY", "MR. ROMNEY"]
#script1 = []
#script2 = []

change = False
speaker = 0
scripts = [[],[]]

for d in debates:
    for l in d:
        if l.startswith(speaker1[0] + ":") or l.startswith(speaker1[1]):
            speaker = 0
            change = True
        elif l.startswith(speaker2[0] + ":") or l.startswith(speaker2[1]):
            speaker = 1
            change = True
        else:
            change = False

        if change:
            splitpoint = l.find(":")
            word_str = l[splitpoint + 1:].lower()
        else:
            word_str = l.lower()

        word_tokens = re.findall(r"\w\w\w+", word_str) # at least 3 characte
rs long
        filtered = [w for w in word_tokens if not w in stop_words]
        scripts[speaker] = scripts[speaker] + filtered

print(len(scripts[0]),len(scripts[1]))
```

```
6034 6262
```

In [4]:

```
# Get frequencies

# Calculate frequency distribution
fdist1 = nltk.FreqDist(scripts[0])
fdist2 = nltk.FreqDist(scripts[1])

word_freq = [[],[]]
freq = []
# Output top 50 words

for word, frequency in fdist1.most_common(20):
    word_freq[0].append((word, frequency))
    freq.append(frequency)
    print(u'{}: {}'.format(word, frequency))
print("###")
for word, frequency in fdist2.most_common(20):
    word_freq[1].append((word, frequency))
    freq.append(frequency)
    print(u'{}: {}'.format(word, frequency))
```

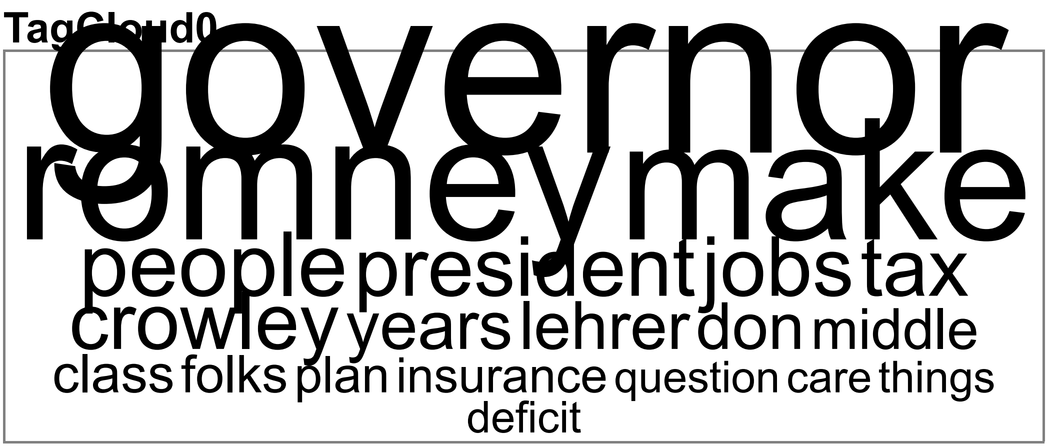
```
governor: 150
romney: 110
make: 104
people: 55
president: 51
jobs: 51
tax: 50
crowley: 49
years: 42
lehrer: 41
don: 40
middle: 35
class: 33
folks: 32
plan: 31
insurance: 30
question: 27
care: 27
things: 27
deficit: 27
###
people: 155
president: 138
don: 62
years: 59
jobs: 59
make: 59
lehrer: 59
america: 56
tax: 56
percent: 51
plan: 50
government: 50
number: 48
cut: 47
work: 39
taxes: 38
economy: 38
question: 37
put: 37
country: 35
```

In [7]:

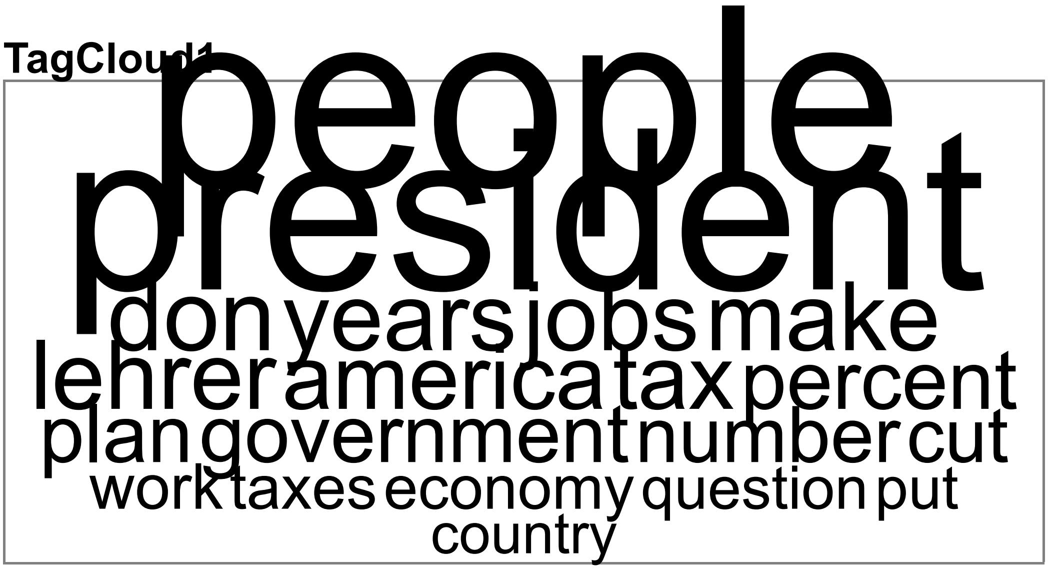
```
high_count=max(freq)
low_count=min(freq)
print(high_count, low_count)

for i, pairs in enumerate(word_freq):
    body=''
    for word,cnt in pairs:
        body = body + " " + make_HTML_word(word,cnt,high_count,low_count)
    box = make_HTML_box(body)  # creates HTML in a box
    print_HTML_file(box,'TagCloud' + str(i))  # writes HTML to file name 'te
stFile.html'
    #display(HTML(box)) # Display HTML
    display(HTML(filename = 'TagCloud' + str(i) + ".html")) # Display HTML
```

```
155 27
```

TagCloud0



TagCloud1



In [ ]: