In this miniproject, you will scrape New York Social Diary (http://www.newyorksocialdiary.com/ (http://www.newyorksocialdiary.com/)). This website provides photographs from the social events that the socialites might attend. Each photo has a caption that labels those who appear in the photo.

You have to find the unique list of the names of the total people who are annotated in the captions of the photos from the parties before December 1st, 2014.

To crawl the data and see the list of the party pages go to http://www.newyorksocialdiary.com/party-pictures (http://www.newyorksocialdiary.com/party-pictures).

Hints for Crawling

1. See how the url of each page changes when you go through different pages. Try to find a plan to get all the data that you need.
2. There are some photos with narrative captions. These captions are not useful for you. It's faster to download all the captions first and filter out those you don't need later.
3. To track the time, you can use python's **'datetime.strptime'** function to parse the date that is located on each party's index page.
4. Save your data so you don't need to re-scrape everytime you run your code. You can pickle it! "https://wiki.python.org/moin/UsingPickle (https://wiki.python.org/moin/UsingPickle)"

Hints for Parsing

1. As it was mentioned there are some long narrative captinos that are not useful. Set a cutoff at 250 characters for these captions.
2. It might be better to use **'re.split'** instead of **'string.split'** when you want to seperate the captions based on different punctuations.
3. There are some titles before some names that you have to find and filter. For example, "Mayor Michael Bloomberg" after his election and "Michael Bloomberg" before his election. In fact, both of these refer to the same person.
4. There are cases that couples are written as e.g. "John and Mary Smith". You have to parse this into two separate names: "John Smith" and "Mary Smith".

Submission

Submit a zip file to Canvas. The zip file should contain your python code for scraping and a number that shows the total count of the names you could fetch from the captions.

Bonus Question (10 points)

Find "who is the most popular"?

To answer this question, you have to find how many connections everyone has. You can think of this problem in terms of the graph 'https://en.wikipedia.org/wiki/Graph_%28discrete_mathematics%29 (https://en.wikipedia.org/wiki/Graph_%28discrete_mathematics%29)'. A pair of two people in a photo is considered a link. You can use python's **'networkx'** library.