# Web Scraping Tutorial Notes

1. Install scrapy
    a. Review http://scrapy.org/ for prerequisites and instructions
    b. Install scrapy with: `conda install scrapy`
    c. Update the cryptography package with: `conda update cryptography`
    d. Install service_identity package with: `pip install service_identity`
    e. Verify scrapy in command prompt with: `scrapy version`
2. In a command prompt or terminal window, change directory to the root and create a new project with the command: `scrapy startproject craigslistRent`
3. Go to craigslist page and identify the elements of data you want from it
    a. Right click the page and choose "view source" to see where these data elements are in the HTML
    b. Review http://www.w3schools.com/xpath/default.asp to figure out how to access your desired data elements in the HTML using XPath
4. Edit *items.py* to reflect the desired data elements
5. Create a new spider python file to scrape the rental listings page
    a. Give your spider a name, like *indexSpider*
    b. Add the URLs to crawl
    c. Add the data cleaning functions
    d. Add the parse function that selects data elements with XPath
6. Test with: `scrapy crawl indexSpider`
7. Run the scraper: `scrapy crawl indexSpider -o temp-rentals.csv -t csv`
    a. `indexSpider` is the name you provided for your spider
    b. `-o temp-rentals.csv` tells it to output to a file called *temp-rentals.csv*
    c. `-t csv` tells it to format the data output to this file as comma-separated values
    d. Remember that each time it runs, it appends, not overwrites the output file
8. The previous step should have created a file called *temp-rentals.csv* with your data in it. Next we need to create a spider that follows each link to the listing's page to acquire lat-long data
9. Create a new spider python file to get lat-long from the individual listings' pages
    a. Give your spider a name, like *latlongSpider*
    b. Add the URLs to crawl dynamically by reading them from *rents.csv*
    c. Add the parse function that selects data elements with XPath and cleans up the data
10. Run the scraper: `scrapy crawl latlongSpider -o temp-latlong.csv -t csv`
11. The previous step should have created a file called *temp-latlong.csv* with your lat-long data in it. Finally, use pandas and the *MergeData.py* script to merge the rental listings from *temp-rentals.csv* with the lat-long data in *temp-latlong.csv*. Your final combined output is in *craigslist-timestamp.csv*.