* This content analysis project will focus on performing data analysis on unstructured material.

Your program is to visit the U.S. Patents website (see at the bottom of this assignment), automatically scrape the individual hyperlinks for each application, and then extract out just the abstract from each patent application.  Then, each abstract is outputted to a single text file, and then you come up with a process to create a keyword concept list for the abstracts, the results of which both get printed to the screen and outputted to a single output file - remove duplicate keyword concepts from the list.  See the Bag of Words description on Blackboard if you need information regarding Bag of Words processing.

You are to use the following libraries in whatever way you see fit:  RE, requests, beautifulsoup and nltk.

Be sure to format all command line output appropriately.  Label all output appropriately.

~~Grading for the project is as follows.  You CANNOT skip partial-credit activities.  You work at your own pace, and you can submit project work to this online hyperlink for grading as many times as you like up to, and including, the due date/time — only the last submission is graded (old submissions are zero’ed out, which has no impact on your final grade).  Based on the passing of each partial-credit activity, you earn the following grade — this is your rubric:~~

* + ~~No activity passed: 0%~~
  + ~~Activity 1 passed: 65%~~
  + ~~Activity 1 & 2 passed: 75%~~
  + ~~Activity 1, 2 & 3 passed: 85%~~
  + ~~Activity 1, 2 & 3 passed: 90%:  Be sure to document your code and data well.  You will also be graded on how well you can describe how it works.  The comments and documentation required here exceed the typical documentation that you would provide in your code to allow another programmer to modify it.  You need to show that you understand how it is working and why you make the choices that you did.  Always be sure, at the top of any source code that you submit, you include four pieces of information: authors' names, creation date, last modification date, and brief description of your source code.  Your code should also be MODULARIZED -- define your own functions/classes to organize your code and avoid "spaghetti code."~~
  + ~~Activity 1, 2 & 3 passed: 100%:  You should include a document that describes your understanding of how well the solution works and in what situations you can apply this to other problems.  Filename:~~**~~P2Documentation\_LASTNAME\_AndrewID.docx~~**

~~Each partial-credit activity builds upon each other, and you may have to rework/rewrite source code to complete the more difficult partial-credit activity.~~  ~~This should NOT be a GUI program — it it text-based.~~  
  
~~Submit your source code (.py files(s) ) for~~**~~each~~**~~fully completed partial activities through this project hyperlink, and your MS Word document for a 100%, via~~**~~ONE~~**~~final submission before the due date/time.  One final submission with all files attached — do~~**~~NOT~~**~~submit each activity separately or your work will not be graded.~~

**Final Project Due date/time:  Friday, April 28th @ 11:59pm ET~~.  No late work will be accepted!~~**

**~~YOU MUST USE PYTHON VERSION 3 OR LATER IN THIS CLASS - OTHERWISE YOU EARN A 0% ON THE ASSIGNMENT.~~**

**\*\*\*\*\*\*\*\*\*\***

**URL:**

The URL will search the US Patent Application database.  These are patent applications that have been submitted, not the patents that have been awarded yet.  It will provide a web page that lists the patent applications as hyperlinks to the applications.

http://appft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fnetahtml%2FPTO%2Fsearch-adv.html&r=0&p=1&f=S&l=50&Query=aanm%2F%22***carnegie+mellon***%22+AND+PD%2F***4***%2F***1***%2F***2016***-%3E***6***%2F***30***%2F***2016***&d=PG01

~~The URL has specific search criteria embedded into it, that have been bolded and underlined for emphasis:~~

~~carnegie+mellon is “carnegie mellon” meaning it searches for that string in the organization (AANM) field.~~

~~4 1 2016 6 30 2016 is “4/1/2016 to 6/30/2016” meaning to search for patent applications that were submitted between 4/1/2016 and 6/30/2016, which is the second quarter of the year.~~

~~For this assignment, use Carnegie Mellon -- but for future reference, you can substitute any company name for carnegie+mellon.  If the company name is more than one word, you need to use the plus sign for the space.~~

~~\*\*\*\*\*\*\*\*\*\*~~

Check out these URLs:

* + http://docs.python-requests.org/
  + https://www.crummy.com/software/BeautifulSoup/bs4/doc/
  + http://www.nltk.org/book/ch03.html
  + http://flockhart.virtualave.net/RBIF0100/regexp.html
* https://blackboard.andrew.cmu.edu/images/ci/sets/set01/document_on.gif

**Summary of Bag of Words**

Summary of “Bag of Words” text processing:

--What’s my end goal?  The end goal of text analysis, for our purposes, is to analyze a domain-specific piece of unstructured text (it can be one article, or even a collection of articles on a specific topic) to find a list of the most important concepts in the unstructured text., along with each concept’s frequency of appearance in the unstructured text  We determine if a concept is important to the subject based on frequency – how often does that concept appear in the unstructured text?  This can be a very powerful tool that you are building, which can help you analyze a topic!

--What’s my approach?  You want to take the unstructured text block, scrub the data to make the text standardized, extract the concepts, and count each concept’s frequency to determine concept importance to the topic at hand.

--What is the process?

\*To scrub the data, you perform:

   \*\*Joining compound concepts == compound concepts are phrases comprised of multiple words.  When the individual words are put together, the individual words’ meaning changes.  “First aid”, or “civil war”, are two examples of compound concepts.  Join the words that comprise the compound concepts together with an underscore; eliminate the whitespace.  For example, replace “first aid” with “first\_aid” to preserve the components that make up the compound concept.

   \*\*Perform normalization == standardize your text.  Replace similar concepts with a single synonym – for instance, replace “maroon”, “dark red”, and “deep red” with “maroon”.  This will give you a more accurate frequency count.  Spell out acronyms – for instance, replace “USA” and “America” with “United\_States\_of\_America”.  Deal with contractions.

   \*\*Perform stemming == get to the root of your words.  Make nouns singular and replace verbs with their present tense forms.  This also improves your frequency count.

   \*\*Noise / stop words removal == replace any non-meaningful words with whitespace once you preserve your compound concepts with underscore insertion.  This can include pronouns, prepositions, verbs of being…

\*Once you scrub/standardize your text, pull out the concepts and count each concept’s frequency.  Frequency is a proxy for importance.  Sort your concept list by frequency so the most important concepts float to the top of the list.

--What does this give me?  By creating this programming tool, you can then analyze any domain-specific topic text to determine the core ideas in the text.  Note that your tool is not article-specific; it is domain specific.  You can analyze any unstructured text with your tool!  It’s very powerful!  You can also process multiple pieces of unstructured text at once, finding the common, important concepts across multiple items.

* https://blackboard.andrew.cmu.edu/images/ci/sets/set01/document_on.gif

**Project #2 Activity #1**

Perform the web scraping described in the project description:

* + scrape the master website;
  + extract the URLs using looping;
  + scrape each URL and only save the abstract on each page;

Print the abstracts to an external text file and the monitor.

You need to provide comments in your program. You should have a comment block at the start of your code that identifies the author, original creation date, last modification date, and gives a description of the program. You will need to provide comments for each procedure and variable. Comments will be a part of each activity grade.

Submit your source code (.py file(s) ) **[filename = P2A1\_LASTNAME\_AndrewID.py]** through Blackboard. All work must be submitted via Blackboard.

**Please keep in mind that you cannot skip activities. You work at your own pace.**

* https://blackboard.andrew.cmu.edu/images/ci/sets/set01/document_on.gif

**Project #2 Activity #2**

Using activiy 1, produce a list of meaningful tokens/concepts from the abstracts.  To do this, you must scrub the abstract data first:

* + Preserve compound concepts -- replace multi-word concepts in the abstracts into single word phrases;  (eg: replace "Karen Bigrigg" with "Karen\_Bigrigg");
  + Perform normalization;
  + Perform stemming;
  + Perform any other replacements you need to eliminate redundancy in your data.

After scrubbing, produce your tokens/concepts list and print them both to the screen and an external text file.

You need to provide comments in your program. You should have a comment block at the start of your code that identifies the author, original creation date, last modification date, and gives a description of the program. You will need to provide comments for each procedure and variable. Comments will be a part of each activity grade.

Submit your source code (.py file(s) ) **[filename = P2A2\_LASTNAME\_AndrewID.py]** through Blackboard. All work must be submitted via Blackboard.

**Please keep in mind that you cannot skip activities. You work at your own pace.**

* https://blackboard.andrew.cmu.edu/images/ci/sets/set01/document_on.gif

**Project #2 Activity #3**

At this point, take your prior activity data and output to the screen and to an external text file a final list of tokens/concepts (all unique -- no duplicates), along with a frequency count stating how many times each keyword appeared in the abstracts.

You need to provide comments in your program. You should have a comment block at the start of your code that identifies the author, original creation date, last modification date, and gives a description of the program. You will need to provide comments for each procedure and variable. Comments will be a part of each activity grade.

Submit your source code (.py file(s) ) **[filename = P2A3\_LASTNAME\_AndrewID.py]** through Blackboard. All work must be submitted via Blackboard.

**Please keep in mind that you cannot skip activities. You work at your own pace**