

Final Project - Word Cloud

For this project, you'll create a "word cloud" from a text by writing a script. This script needs to process the text, remove punctuation, ignore case and words that do not contain all alphabets, count the frequencies, and ignore uninteresting or irrelevant words. A dictionary is the output of the `calculate_frequencies` function. The `wordcloud` module will then generate the image from your dictionary.

For the input text of your script, you will need to provide a file that contains text only. For the text itself, you can copy and paste the contents of a website you like. Or you can use a site like [Project Gutenberg](#) to find books that are available online. You could see what word clouds you can get from famous books, like a Shakespeare play or a novel by Jane Austen. Save this as a .txt file somewhere on your computer.

Now you will need to upload your input file here so that your script will be able to process it. To do the upload, you will need an uploader widget. Run the following cell to perform all the installs and imports for your word cloud script and uploader widget. It may take a minute for all of this to run and there will be a lot of output messages. But, be patient. Once you get the following final line of output, the code is done executing. Then you can continue on with the rest of the instructions for this notebook.

Enabling notebook extension fileupload/extension..
- Validating: **OK**

```
get_ipython().system('pip install wordcloud')
get_ipython().system('pip install fileupload')
get_ipython().system('pip install ipynbwidgets')
get_ipython().system('jupyter nbextension install --py user fileupload')
get_ipython().system('jupyter nbextension enable --py fileupload')

import wordcloud
import numpy as np
from matplotlib import pyplot as plt
from IPython.display import display
import fileupload
import io
import sys

Requirement already satisfied: wordcloud in /opt/conda/lib/python3.6/site-packages (1.8.2.2)
Requirement already satisfied: pillow in /opt/conda/lib/python3.6/site-packages (from wordcloud) (5.4.1)
Requirement already satisfied: numpy>=1.6.1 in /opt/conda/lib/python3.6/site-packages (from wordcloud) (3.9.3)
Requirement already satisfied: numpy>=1.6.1 in /opt/conda/lib/python3.6/site-packages (from wordcloud) (1.15.4)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib>=wordcloud) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib>wordcloud) (0.6.1)
Requirement already satisfied: pyrsnippet>=2.0.4,!=2.1.2,!=2.1.6,!=2.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib>wordcloud) (2.3.1)
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib>wordcloud) (2.6.0)
Requirement already satisfied: six in /opt/conda/lib/python3.6/site-packages (from cycler>=0.10->matplotlib>wordcloud) (1.12.0)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwisolver>=1.0.1->matplotlib>wordcloud) (40.8.0)
Requirement already satisfied: fileupload in /opt/conda/lib/python3.6/site-packages (0.1.5)
Requirement already satisfied: ipynbwidgets>=5.1 in /opt/conda/lib/python3.6/site-packages (from fileupload) (7.4.2)
Requirement already satisfied: notebook>=4.2 in /opt/conda/lib/python3.6/site-packages (from fileupload) (5.7.5)
Requirement already satisfied: traitlets>=4.2 in /opt/conda/lib/python3.6/site-packages (from fileupload) (4.3.2)
Requirement already satisfied: nbformat>=4.2.0 in /opt/conda/lib/python3.6/site-packages (from ipynbwidgets>=5.1->fileupload) (4.4.0)
Requirement already satisfied: widgetsnbextension>=3.4.0 in /opt/conda/lib/python3.6/site-packages (from ipynbwidgets>=5.1->fileupload) (3.4.2)
Requirement already satisfied: ipykernel>=4.5.1 in /opt/conda/lib/python3.6/site-packages (from ipynbwidgets>=5.1->fileupload) (4.5.1)
Requirement already satisfied: ipynbwidgets>=5.1->fileupload (7.4.2)
Requirement already satisfied: ipynbwidgets>=5.1->fileupload (7.4.0)
Requirement already satisfied: nbconvert in /opt/conda/lib/python3.6/site-packages (from notebook>=4.2->fileupload) (5.4.0)
Requirement already satisfied: prometheus-client in /opt/conda/lib/python3.6/site-packages (from notebook>=4.2->fileupload) (0.6.0)
Requirement already satisfied: terminado>=0.8.1 in /opt/conda/lib/python3.6/site-packages (from notebook>=4.2->fileupload) (0.8.1)
Requirement already satisfied: Send2Trash in /opt/conda/lib/python3.6/site-packages (from notebook>=4.2->fileupload) (1.5.0)
Requirement already satisfied: tornado>=4.1 in /opt/conda/lib/python3.6/site-packages (from notebook>=4.2->fileupload) (6.0.2)
Requirement already satisfied: jupyter-core>=4.4.0 in /opt/conda/lib/python3.6/site-packages (from notebook>=4.2->fileupload) (4.4.0)
Requirement already satisfied: ipython-genutils in /opt/conda/lib/python3.6/site-packages (from notebook>=4.2->fileupload) (0.2.0)
Requirement already satisfied: jupyter-client>=5.2.0 in /opt/conda/lib/python3.6/site-packages (from notebook>=4.2->fileupload) (5.2.0)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.6/site-packages (from notebook>=4.2->fileupload) (2.1.0)
Requirement already satisfied: pyzmq>=17 in /opt/conda/lib/python3.6/site-packages (from notebook>=4.2->fileupload) (18.0.1)
Requirement already satisfied: six in /opt/conda/lib/python3.6/site-packages (from traitlets>=4.2->fileupload) (1.12.0)
Requirement already satisfied: decorator in /opt/conda/lib/python3.6/site-packages (from traitlets>=4.2->fileupload) (4.3.2)
Requirement already satisfied: jsonschema>=2.5.0,!=2.4 in /opt/conda/lib/python3.6/site-packages (from nbformat>=4.2.0->ipynbwidgets>=5.1->fileupload) (3.0.1)
Requirement already satisfied: setuptools>=18.5 in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets>=5.1->fileupload) (40.8.0)
Requirement already satisfied: jedi>=0.10 in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets>=5.1->fileupload) (0.13.3)
Requirement already satisfied: pickleshare in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets>=5.1->fileupload) (0.7.5)
Requirement already satisfied: prompt_toolkit<2.1.0,>=2.0.0 in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets>=5.1->fileupload) (2.0.9)
Requirement already satisfied: pygments in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets>=5.1->fileupload) (2.3.1)
Requirement already satisfied: backcall in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets>=5.1->fileupload) (0.1.0)
Requirement already satisfied: pexpect in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets>=5.1->fileupload) (4.6.0)
Requirement already satisfied: mistune>=0.8.1 in /opt/conda/lib/python3.6/site-packages (from nbconvert->notebook>=4.2->fileupload) (0.8.4)
Requirement already satisfied: entrypoints>=0.2.2 in /opt/conda/lib/python3.6/site-packages (from nbconvert->notebook>=4.2->fileupload) (0.3)
Requirement already satisfied: bleach in /opt/conda/lib/python3.6/site-packages (from nbconvert->notebook>=4.2->fileupload) (3.1.0)
Requirement already satisfied: pandocfilters>=1.4.1 in /opt/conda/lib/python3.6/site-packages (from nbconvert->notebook>=4.2->fileupload) (1.4.2)
Requirement already satisfied: testpath in /opt/conda/lib/python3.6/site-packages (from nbconvert->notebook>=4.2->fileupload) (0.4.2)
Requirement already satisfied: defusedxml in /opt/conda/lib/python3.6/site-packages (from nbconvert->notebook>=4.2->fileupload) (0.5.0)
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.6/site-packages (from jupyter-client>=5.2.0->notebook>=4.2->fileupload) (2.8.0)
Requirement already satisfied: MarkupSafe>=0.23 in /opt/conda/lib/python3.6/site-packages (from jinja2>=notebook>=4.2->fileupload) (1.1.1)
Requirement already satisfied: attrs>=17.4.0 in /opt/conda/lib/python3.6/site-packages (from jsonschema>=2.5.0,!=2.4->nbformat>=4.2.0->ipynbwidgets>=5.1->fileupload) (19.1.0)
Requirement already satisfied: pyrsnippet>=0.14.0 in /opt/conda/lib/python3.6/site-packages (from jsonschema>=2.5.0,!=2.4->nbformat>=4.2.0->ipynbwidgets>=5.1->fileupload) (0.14.1)
Requirement already satisfied: parso>=0.5.0 in /opt/conda/lib/python3.6/site-packages (from jedi>=0.10->ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets>=5.1->fileupload) (0.3.4)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.6/site-packages (from prompt_toolkit<2.1.0,>=2.0.0->ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets>=5.1->fileupload) (0.1.7)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.6/site-packages (from pexpect->ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets>=5.1->fileupload) (0.6.0)
Requirement already satisfied: webencodings in /opt/conda/lib/python3.6/site-packages (from bleach->nbconvert->notebook>=4.2->fileupload) (0.5.1)
Requirement already satisfied: ipynbwidgets in /opt/conda/lib/python3.6/site-packages (7.4.2)
Requirement already satisfied: widgetsnbextension>=3.4.0 in /opt/conda/lib/python3.6/site-packages (from ipynbwidgets) (3.4.2)
Requirement already satisfied: nbformat>=4.2.0 in /opt/conda/lib/python3.6/site-packages (from ipynbwidgets) (4.4.0)
Requirement already satisfied: ipython>=4.0.0; python_version >= "3.3" in /opt/conda/lib/python3.6/site-packages (from ipynbwidgets) (4.0.0)
Requirement already satisfied: ipykernel>=4.5.1 in /opt/conda/lib/python3.6/site-packages (from ipynbwidgets) (5.1.0)
Requirement already satisfied: traitlets>=4.3.1 in /opt/conda/lib/python3.6/site-packages (from ipynbwidgets) (4.3.2)
Requirement already satisfied: notebook>=4.4.1 in /opt/conda/lib/python3.6/site-packages (from ipynbwidgets) (4.4.1)
Requirement already satisfied: ipython_genutils in /opt/conda/lib/python3.6/site-packages (from nbformat>=4.2.0->ipynbwidgets) (0.2.0)
Requirement already satisfied: jsonschema>=2.5.0,!=2.4 in /opt/conda/lib/python3.6/site-packages (from nbformat>=4.2.0->ipynbwidgets) (3.0.1)
Requirement already satisfied: jupyter_core in /opt/conda/lib/python3.6/site-packages (from nbformat>=4.2.0->ipynbwidgets) (4.4.0)
Requirement already satisfied: setuptools>=18.5 in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets) (40.8.0)
Requirement already satisfied: jedi>=0.10 in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets) (0.13.3)
Requirement already satisfied: pickleshare in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets) (0.7.5)
Requirement already satisfied: prompt_toolkit<2.1.0,>=2.0.0 in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets) (2.0.9)
Requirement already satisfied: pygments in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets) (2.3.1)
Requirement already satisfied: backcall in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets) (0.1.0)
Requirement already satisfied: pexpect in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3" -> ipynbwidgets) (4.6.0)
Requirement already satisfied: jupyter-client in /opt/conda/lib/python3.6/site-packages (from ipykernel>=4.5.1->ipynbwidgets) (5.2.0)
Requirement already satisfied: tornado>=4.2 in /opt/conda/lib/python3.6/site-packages (from ipykernel>=4.5.1->ipynbwidgets) (6.0.2)
Requirement already satisfied: six in /opt/conda/lib/python3.6/site-packages (from traitlets>=4.3.1->ipynbwidgets) (1.12.0)
Requirement already satisfied: nbconvert in /opt/conda/lib/python3.6/site-packages (from notebook>=4.4.1->widgetsnbextension>=3.4.0->ipynbwidgets) (5.4.1)
Requirement already satisfied: pyzmq>=17 in /opt/conda/lib/python3.6/site-packages (from notebook>=4.4.1->widgetsnbextension>=3.4.0->ipynbwidgets) (18.0.1)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.6/site-packages (from notebook&gt
```

To initialize this nbextension in the browser every time the notebook (or other app) loads:

```
jupyter nbextension enable fileupload --user --py
```

Whew! That was a lot. All of the installs and imports for your word cloud script and uploader widget have been completed.

IMPORTANT! If this was your first time running the above cell containing the installs and imports, you will need to save this notebook now. Then under the File menu above, select Close and Halt. When the notebook has completely shut down, reopen it. This is the only way the necessary changes will take effect.

To upload your text file, run the following cell that contains all the code for a custom uploader widget. Once you run this cell, a "Browse" button should appear below it. Click this button and navigate the window to locate your saved text file.

```
In [2]: def _upload():

    _upload_widget = fileupload.FileUploadWidget()

    def _cb(change):
        global file_contents
        decoded = io.StringIO(change['owner'].data.decode('utf-8'))
        filename = change['owner'].filename
        print('Uploaded "{}" ({:.2f} kB)'.format(
            filename, len(decoded.read()) / 2 **10))
        file_contents = decoded.getvalue()

    _upload_widget.observe(_cb, names='data')
    display(_upload_widget)

_upload()
```

Uploaded 'me.txt' (0.04 kB)

The uploader widget saved the contents of your uploaded file into a string object named `file_contents` that your word cloud script can process. This was a lot of preliminary work, but you are now ready to begin your project.

Write a function in the cell below that iterates through the words in `file_contents`, removes punctuation, and counts the frequency of each word. Oh, and be sure to make it ignore word case, words that do not contain all alphabets and boring words like "and" or "the". Then use it in the `generate_from_frequencies` function to generate your very own word cloud!

Hint: Try storing the results of your iteration in a dictionary before passing them into wordcloud via the `generate_from_frequencies` function

```

In [3]: def calculate_frequencies(file_contents):
# Here is a list of punctuation and uninteresting words you can use to process your text
punctuations = '''(){}:;'\",<./?@#%&*~_!'''
uninteresting_words = ["the", "a", "to", "if", "is", "it", "of", "and", "or", "an", "as", "i", "me", "my",
"we", "your", "our", "you", "yours", "he", "she", "him", "his", "her", "hers", "its", "they", "them",
"their", "what", "which", "who", "whom", "this", "that", "am", "are", "was", "were", "be", "been", "being",
"has", "had", "do", "does", "died", "but", "at", "by", "with", "from", "here", "when", "where", "how",
"any", "both", "each", "few", "more", "some", "such", "no", "nor", "too", "very", "can", "will", "just"]

# LEARNER CODE START HERE
refined_text = ""
frequency_count = {}

#removing punctuation s and uninteresting words
for char in file_contents:
    if char not in punctuations:
        refined_text += char

#splitting the text into word list
word_list = refined_text.split()

#removing uninteresting words
for word in word_list:
    if word.lower() not in uninteresting_words:
        if word in frequency_count:
            frequency_count[word] += 1
        else:
            frequency_count[word] = 1

#wordcloud
cloud = wordcloud.WordCloud()
cloud.generate_from_frequencies(frequency_count)
return cloud.to_array()

```

If you have done everything correctly, your word cloud image should appear after running the cell below. (Fingers crossed!)

```
In [4]: # Display your wordcloud image

myimage = calculate_frequencies(file_contents)
plt.imshow(myimage, interpolation = 'nearest')
plt.axis('off')
plt.show()
```

The logo for 'Crash Course Automation Python' features the words 'Automation' and 'Python' in a light blue font, with 'Crash Course' in a larger, bold, light blue font in the center. The background is a solid black rectangle.

If your word cloud image did not appear, go back and rework your `calculate_frequencies` function until you get the desired output. Definitely check that you passed your frequency count dictionary into the `generate_from_frequencies` function of `wordcloud`. Once you have correctly displayed your word cloud image, you are all done with this project. Nice work!