

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 <u>Project Gutenberg</u> 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

Collecting wordcloud
Downloading https://files.pythonhosted.org/packages/71/8b/0feea1341a29c645ca9dbc7f453b1101ee0725059977c7fce23e1bd75b83/word
cloud-1.8.2.2.tar.gz (220kB)

100% | 225kB 26.6MB/s ta 0:00:01

Requirement already satisfied: numpy>=1.6.1 in /opt/conda/lib/python3.6/site-packages (from wordcloud) (1.15.4)

Requirement already satisfied: pillow in /opt/conda/lib/python3.6/site-packages (from wordcloud) (5.4.1)

Requirement already satisfied: matplotlib in /opt/conda/lib/python3.6/site-packages (from matplotlib-wordcloud) (0.10.0)

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) (1.0.1)

Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib-wordcloud)

Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib-wordcloud)

(2.8.0)
Requirement already satisfied: six in /opt/conda/lib/python3.6/site-packages (from cycler>=0.10->matplotlib->wordcloud) (1.1
2.0)

Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwisolver>=1.0.1->matplotlib->word cloud) (40.8.0)
Building wheels for collected packages: wordcloud

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.

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 script.

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". I nen 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.

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

```
In [11]: # Display your wordcloud image
myimage = calculate_frequencies(file_contents)
plt.imshow(myimage, interpolation = 'nearest')
plt.axis('off')
plt.show()
```



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!