

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.

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.

```
# Here are all the installs and imports you will need for your word cloud script and uploader widget
```

```
Collecting wordcloud
  Downloading https://files.pythonhosted.org/packages/05/ef/52e4bef8e2e3499f6e96c8ff7e90240b95014134b0624edc4ff8b9f8c/wordcloud-1.8.1-cp36-cp36m-manylinux1_x86_64.whl (366kB)
    100% |#####| 366kB 26.0MB/s ta 0:00:01
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 wordcloud) (3.0.3)
Requirement already satisfied: numpy>=1.6.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib) (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) (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) (2.3.1)
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) (1.10.0)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwisolver>=1.0.1->matplotlib>wordcloud) (40.8.0)
Installing collected packages: wordcloud
```

IMPORTANT! If this was your first time running the above cell containing the installs and imports, you will need 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 affect.

```
In [3]: # This is the uploader widget
```

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

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.

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

```
[In 23]: def calculate_frequencies(file_contents):
# Here is a list of punctuations and interesting words you can use to process your text
punctuations = '''!()-[]{};:'"\,.<./?@#%&*~'''
interesting_words = ["the", "a", "to", "if", "is", "it", "of", "and", "on", "an", "as", "i", "me", "my", \
"we", "our", "ours", "you", "your", "yours", "he", "she", "him", "his", "her", "hers", "its", "they", "them", \
"their", "what", "which", "who", "how", "this", "that", "am", "are", "was", "were", "be", "been", "being", \
"have", "has", "had", "do", "does", "did", "but", "at", "by", "with", "from", "here", "when", "where", "how", \
"all", "any", "both", "each", "few", "more", "some", "such", "no", "nor", "too", "very", "can", "will", "just"]

# LEARNER CODE START HERE
for punctuation in punctuations:
    file_contents = file_contents.replace(punctuation, '')

file_contents = file_contents.lower()

frequencies = {}
words = file_contents.split(' ')

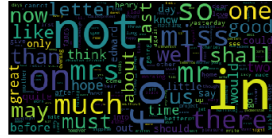
for word in words:
    if word not in interesting_words:
        if word in frequencies.keys():
            frequencies[word] += 1
        else:
            frequencies[word] = 1

#wordcloud
cloud = wordcloud.WordCloud()
cloud.generate_from_frequencies(frequencies)
return cloud.to_array()
```

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

```
In [24]: # 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!