Activity

- Use the harry-potter dataset
- Explore the pattern
- Identify the cleaning points
- Conduct text cleaning
- Explore the word and character distribution
- Do you see different patterns from what you observed in the morning?

- Use the harry-potter dataset

In [2]:
```
import nltk
from nltk.tokenize import word_tokenize
```

In [3]:
```
f = open('harry_7books.txt', encoding='utf8')
raw = f.read()
```

In [4]:
```
tokens = word_tokenize(raw)
```

In [32]:
```
tokens[10:20]
```

Out[32]:
```
['rowling',
 'illustrations',
 'by',
 'mary',
 'grandpré',
 'arthur',
 'a.',
 'levine',
 'books',
 'an']
```

- Explore the pattern

In [33]:
```
import re

[w for w in tokens if re.search('ed$', w)][10:20]
```

Out[33]:
```
['roared',
 'tried',
 'bored',
 'used',
 'snarled',
 'fried',
 'exchanged',
 'tried',
 'drowned',
 'drooped']
```

In [34]:
```
[w for w in tokens if re.search(',|©|"', w)][10:20]
```

Out[34]:
```
['rowling',
 'illustrations',
 'by',
 'mary',
 'grandpré',
 'arthur',
 'a.',
 'levine',
 'books',
 'an']
```

In [8]:
```
[w for w in tokens if re.search('^[0-9]{1}-[0-9]{3}', w)]
```

Out[8]:
```
['0-439-06486-4',
 '0-545-02936-8',
 '0-439-13959-7',
 '0-439-78454-9',
 '0-439-35806-X',
 '0-439-13635-0',
 '0-590-35340-3']
```

In [35]:
```
[w for w in tokens if re.search('^[0-9]', w)][10:20]
```

Out[35]:
```
['1999', '98-46370', '60', '59', '58', '57', '56', '55', '54', '53']
```

In [36]:
```
[w for w in tokens if re.search('\W+\D', w)][10:20]
```

Out[36]:
```
['mat\xadter',
 'sleep\xading',
 'four-poster',
 'four\xadteen',
 'top-of-the-line',
 'sum\xadmer',
 'wiz\xadards',
 'horse-faced',
 'bril\xadliant',
 'jet-black']
```

In [37]:
```
[w for w in tokens if re.search('\xad', w)][10:20]
```

Out[37]:
```
['un\xadderstanding',
 'some\xadthing',
 'ig\xadnore',
 'ca\xadreer',
 'un\xadcle',
 'talk\xading',
 'wel\xadcome',
 'sim\xadpering',
 'compli\xadments',
 'petu\xadnia']
```

In [38]:
```
[w for w in tokens if re.search('\.+$', w)][10:20]
```

Out[38]:
```
['.', 'inc.', '.', 'inc.', 'bros.', 'j.', 'k.', '.', 'inc.', '.']
```

In [39]:
```
[w for w in tokens if re.search('^[a-zA-Z]-', w)][10:20]
```

Out[39]:
```
['c-cat',
 'm-millicent',
 'm-must',
 'p-potion',
 'u-bend',
 'd-did',
 'v-very',
 'b-breakfast',
 'c-couldn',
 's-swear']
```

In [40]:
```
[w for w in tokens if re.search('\.$', w)][10:20]
```

Out[40]:
```
['.', 'inc.', '.', 'inc.', 'bros.', 'j.', 'k.', '.', 'inc.', '.']
```

- Identify the cleaning points
- Conduct text cleaning

In [41]:
```
tokens = [w.lower() for w in tokens]
tokens[10:20]
```

Out[41]:
```
['rowling',
 'illustrations',
 'by',
 'mary',
 'grandpré',
 'arthur',
 'a.',
 'levine',
 'books',
 'an']
```

In [16]:
```
from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)
```

```
{'were', "mightn't", 'once', "aren't", 'll', 'under', 'above', "couldn't", 'ain', "it's", 'in', 'how', 'off', 're', "didn't", 'y', 'than', 'has', 'such', 'your', 'are', 'over', "wasn't", "isn't", 'should', 'that', "didn't", "shouldn't", 'why', 'can', 'his', 'mightn', 'am', 'against', 'when', 'we', 'some', 'through', 't', 'must', 'but', 'won', 'me', 'was', 'a', 'again', 'up', 'what', 'not', 'out', "hadn't", 'there', 'down', 'into', 'whom', 'shan', 'him', "you've", "you'll", 'only', 'until', 'having', 'herself', "couldn't", 'between', 'haven', "needn't", 'these', "doesn't", 'be', "needn", 'no', 'my', 'after', 'have', 'of', 'will', "doesn", "hasn't", "don't", "at", "whole", "all", 'own', 'or', 'from', 'being', 'for', "ma", "you're", "don't", 'just', 'yours', 'nor', 'yourself', 'their', "mustn", 'each', 'the', 'theirs', 'its', 'weren't", 'she', 'other', 'few', 'he', 'here', 'i', "has n't", "if", "ve", 'ourselves', 'with', 'shouldn', 'did', 'now', 'who', 'wouldn', 'it', 'do', 'which', 'themselves', 'her', 'very', 'too', 'those', 'then', 'them', 'o', 'during', 'further', 'as', 'on', 'any', 'hasn', 'does', 'to', 'an', "you'd", 'more', "wouldn't", 'they', 'm', 'hers', 'yourselves', 'where', 'been', 'because', "won't", "mustn't", 'you', 'itself', 'ours', 'by', 'same', "shan't", 'himself', 'before', 'and', "she's", 'so', 'below', 'weren', 's', 'about', 'd', "should've", "haven't", 'wasn', "that'll", 'had', 'is', 'myself', 'aren', "hadn't", 'out', "isn't", 'doing', 'this'}
```

In [17]:
```
filtered_sent=[]
for w in tokens:
    if w not in stop_words:
        filtered_sent.append(w)
```

In [42]:
```
filtered_sent[10:20]
```

Out[42]:
```
['arthur',
 'a.',
 'levine',
 'books',
 'imprint',
 'scholastic',
 'press',
 'sean',
 'p.',
 'f.']
```

In [19]:
```
from nltk.stem.wordnet import WordNetLemmatizer
lem = WordNetLemmatizer()

words = []
for w in filtered_sent:
    words.append(lem.lemmatize(w))
```

In [43]:
```
words[10:20]
```

Out[43]:
```
['arthur',
 'a.',
 'levine',
 'book',
 'imprint',
 'scholastic',
 'press',
 'sean',
 'p.',
 'f.']
```

In [44]:
```
clean_text = [w for w in words if re.findall('^[a-zA-Z]-', w)] +[w for w in words if re.findall('\xad', w)]+[w

clean_text[10:20]
```

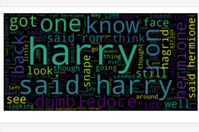
Out[44]:
```
['c-cat',
 'm-millicent',
 'm-must',
 'p-potion',
 'u-bend',
 'd-did',
 'v-very',
 'b-breakfast',
 'c-couldn',
 's-swear']
```

In [ ]:
```
new_list = [w for w in words if w not in clean_text]

new_list[10:20]
```

- Explore the word and character distribution
- Do you see different patterns from what you observed in the morning?

In [28]:
```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

#convert list to string and generate
unique_string=(" ").join(new_list)
wordcloud = WordCloud(width = 1000, height = 500).generate(unique_string)
```

In [29]:
```
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



In [ ]:
```

```