

```

# CHALLENGE 3: Solution
import numpy as np
import matplotlib.pyplot as plt
import wikipedia, random, time, string

# 1. Load the content of the 'hello world' page and print.
search = wikipedia.search('Hello world')
page = wikipedia.WikipediaPage(search[0])
print(page.summary)

# 2. Run frequency analysis on the Oxford page string.
S = wikipedia.search('Oxford')
page = wikipedia.WikipediaPage(S[0])
oxfordText = page.content
freqEL=[8.16,1.49,2.78,4.25,12.70,2.23,2.02,6.09,6.97,0.15,0.77,4.02,2.40,
        6.74,7.50,1.92,0.09,5.98,6.32,9.05,2.75,0.97,2.36,0.15,1.97,0.07]
relFreq = dict.fromkeys(string.ascii_uppercase, 0)

nLetters = 0
for letter in relFreq.keys():
    letterCount = oxfordText.upper().count(letter)
    relFreq[letter] = letterCount
    nLetters = nLetters + letterCount # ignore spaces, punctuation etc.

relFreqOxf = 100*np.array(list(relFreq.values()))/nLetters

plt.bar(np.arange(26)-0.25,freqEL,width=0.3,label='E.L. Freq.')
plt.bar(np.arange(26)+0.25,relFreqOxf,width=0.3,label='\Oxford\')
plt.xticks(np.arange(26),labels=relFreq.keys())
plt.ylabel('Relative Frequency, %'); plt.legend()
plt.show()

# 3. Make a wordcloud of the page of text for Oxford.
from wordcloud import WordCloud; from random import random
wordcloud = WordCloud().generate(oxfordText)
plt.imshow(wordcloud, interpolation='bilinear')
plt.show()

# Extension 1 & 2: Cycle through random pages starting at a page.
S = wikipedia.search('Ham and Cheese Sandwich')
page = wikipedia.WikipediaPage(S[0])
for i in range(10):
    try:
        randomPage = page.links[int(random()*len(page.links))]
        page = wikipedia.WikipediaPage(randomPage); print(randomPage)
    except:
        print('Page Error! try a different link. '); pass

# Extension 3: How long would it take to read wikipedia using this API?
t = time.time()
for i in range(10):
    randomPage = wikipedia.WikipediaPage(wikipedia.random())
timeToRead = (time.time() - t)*(48174651/10)/3600 # in hours.

```