In [1]:
```
# Name of creator
CREATOR_NAME = "Jingheng Wang"
```

This file is Step 1 of Question 2. The previous half of this file is intended to scrape the 20K most frequently used Japanese Words. The latter half is an online demo for query: given a kanji, find those words using that kanji, and scrape the information in the dictionary online. The latter half is not used in answering the question.

Please run the cells following the given order.

In [2]:
```
# Initialization
import numpy as np
import pandas as pd
from bs4 import BeautifulSoup
from requests import get
from time import sleep
from random import randint
```

In [3]:
```
# Two target URLs, one for 1-10K list, another for 10001-20K list
target_url = 'https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/Japanese'
target_url2 = 'https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/Japanese10
001-20000'

raw_html = get(target_url)
soup = BeautifulSoup(raw_html.text, 'html.parser')

# find all <li> tags
raw_wordlist = soup.find('ol').findAll('li')

raw_html2 = get(target_url2)
soup2 = BeautifulSoup(raw_html2.text, 'html.parser')

# find all <li> tags
raw_wordlist2 = soup2.find('ol').findAll('li')

# combine the 20K tags together
raw_wordlist = raw_wordlist + raw_wordlist2[1:]
# THERE IS AN EMPTY LI TAG IN 10K-20K WEBPAGE, SO REMOVE IT
```

In [4]:
```
# Clean the tags and unzip the word itself
def data_clean(x):
    #if (x.findAll('span') is None):
        #print(x)
    if x.span is not None:
        y = x.find('a', class_='extiw')
        if y is not None:
            #SPECIAL COMPANY NAMES, LIKE TOUEI (LINE 3473), SHOUCHIKU(LINE 4258)
            return x.get_text()[:2]
        else:
            #print(x)
            return x.span.a.get_text()
    else:
        return x.get_text()

wordlist = [data_clean(x) for x in raw_wordlist]

wordlist
```

Out[4]:
```
['の',
 'に',
 'する',
 'は',
 'を',
 'が',
 'と',
 '年',
 'で',
 'だ',
 '月',
 'も',
 'から',
 '日',
 '成る',
 'こと',
 '有る',
 'よる',
 'や',
 'など',
 '言う',
 '日本',
 '為',
 'この',
 '人',
 'その',
 'まで',
 'もの',
 'へ',
 '又',
 'これ',
 '行う',
 'よう',
 '出来る',
 '駅',
 '国',
 'より',
 '大学',
 '現在',
 '後',
 'か',
 '線',
 'ぬ',
 '放送',
 '号',
 '軍',
 '無い',
 '部',
 '持つ',
 '所',
 '名',
 '回',
 '世界',
 '時',
 '戦',
 '時代',
 '東京',
 'おく',
 'でも',
 '呼ぶ',
 'その後',
 '会',
 'それ',
 '機',
```

In [5]:
```
# turn to a series. Check if any component is empty
sr = pd.Series(wordlist)
for i in np.arange(20000):
    if sr[i] is None:
        print(i)
```

In [6]:
```
# display the series
sr
```

Out[6]:
```
0          の
1          に
2         する
3          は
4          を
          ..
19995    洋館
19996    相同
19997    美祢
19998    議案
19999    赤身
Length: 20000, dtype: object
```

In [7]:
```
# Write the words into a csv file for scraping
pd.DataFrame({
    'rank':np.arange(1,20001),
    'word':wordlist
}).to_csv('wikitionary_wordlist.csv')
```

Online Scraping Demo

The second part of this notebook works as a demo of word scraping. Please run the functions and initializations first.

The first demo is, for a given kanji, display all words that uses the kanji.

In [8]:
```
# Separate each string to a char list so we can use 'in'
wordlist_sep_char = [list(x) for x in wordlist]

# x is a given kanji
def find_word(x):
    wlist = []
    # for each word in 20K list
    for i, li in enumerate(wordlist_sep_char):
        # if the kanji is in the that word
        if (x in li):
            # append the word to a return list
            wlist.append(wordlist[i])
    return wlist
```

In [9]:
```
# Modify me if you want to change kanji
query_kanji = '夏'
find_word(query_kanji)

# Sample Kanjis
# 夏 (summer)
# 冬 (winter)
# 甘 (sweet)
# 紙 (paper)
# 灯 (light)
```

Out[9]:
```
['夏', '夏季', '夏期', '夏目', '夏場', '初夏', '春夏']
```

The second part contains a more comprehensive task. For each of the words, request the page on dictionary Jisho (jisho.org), and scrape the reading and meaning of them.

This part is an online demo, meaning it will do the requests after we provide the kanji.

In [10]:
```
# Initialization: all furigana/katakana (also called kamei) in Japanese
kamei_list = list("あいうえおかきくけこさしすせそたちつてとなにぬねのはひふへほまみむめもやゆよらりるれろわをんがぎぐげござじずぜぞだぢづでどばびぶべぼパピプペポアイウエロカキクケコサシスセソタチツテトナニヌネノハヒフヘホマミムメモヤユヨラリルレロワヲンガギグゲゴザジズゼゾダヂヅデドバビブベボパピプペポャュョー")

# given the soup, scrape the readings of that word
# Attention: the cleaning process is a bit weird.
#            The front-side programmer of Jisho.org is dealing with the readings in a way extremely not friendly for scraping programs.
#            That's why I need to use the list upon.
#            I'm sorry, but this is hard to explain to a person who doesn't have Japanese learning background.

def get_furigana(soup):
    # The two tags: one for readings, one for text
    furi = soup.find('span',class_='furigana')
    txt = furi.next_sibling.next_sibling
    text_cont = list(str(txt.get_text()).strip())
    furistring = ""
    txtstring = ""
    text_loc = 0
    for f1 in furi.children:
        #print(f1)
        if ((f1 != '\n') & (f1 is not None)):
            t1 = text_cont[text_loc]
            txtstring += t1
            # If the tag is empty, while the text is a kamei, copy that to the reading
            if (f1.get_text() == '') & (t1 in kamei_list):
                furistring += t1
            else:
                furistring += f1.get_text().strip()
            text_loc += 1
    return (furistring, txtstring)
```

In [11]:
```
# given the soup, scrape the meanings in that list

def get_meanings(soup):
    meanings = ""
    raw_meanings = soup.findAll('div', class_='meaning-wrapper')
    for x in raw_meanings:
        flag = x.findAll('span', class_='meaning-definition-section_divider')
        if ((flag is not None) & (flag != [])):
            tag = flag[0]
            # Split the meanings with $ sign
            meanings += tag.get_text() + tag.next_sibling.get_text() + '$'
    return meanings
```

Strategy:

For each word, there can be many readings/meanings, and they are listed in different pages. Since we do not know how many webpages are there for each word, we try to query on the same word, until the status code is 404. In that case, we move to the next word.

There might be cases that the server returns 408 (Timed out). In that case, remade the request.

In [12]:
```
# for a given word, scrape the jisho page and call the previous two functions to ge
t their readings and meanings

def find_word_info(word):
    url_base = 'https://jisho.org/word/'
    wordlist = []
    furigana = []
    meanings = []
    i = 0
    requests = 0
    while (True):

        # Jisho's naming policy for words with different readings/meanings
        if (i == 0):
            url = url_base+word
        else:
            url = url_base+word+'-'+str(i)

        print(url)

        response = get(url)
        requests += 1
        print("Requests Made: {}, status {}".format(requests, response.status_cod
e))
        sleep(0.5)

        i += 1

        if (response.status_code != 200):
            if (response.status_code == 408):
                # Timed Out
                i -= 1
                continue
            if (response.status_code != 404):
                # If that is not 404 nor 408, report this issue
                print("Error: code {} at word {}, i={}".format(response.status_cod
e, word, i-1))
            # break this loop and continue to the next word
            break
        else:
            soup = BeautifulSoup(response.text, 'html.parser')
            furi, txt = get_furigana(soup)
            mean = get_meanings(soup)
            print(txt+' / '+furi+' / '+mean)
            wordlist.append(txt)
            furigana.append(furi)
            meanings.append(mean)

    return (wordlist,furigana,meanings)
```

In [13]:
```
# for a given kanji, find words that uses that kanji, and get their readings/meanin
gs, combining them to a data frame
# This will take some time to complete, as it needs online requests.

# Modify me if you want to change the query kanji
query_kanji = '紙'

# Sample Kanjis
# 夏 (summer)
# 冬 (winter)
# 甘 (sweet)
# 紙 (paper)
# 灯 (light)

w = []
f = []
m = []
for word in find_word(query_kanji):
    print('Start Query:{}'.format(word))
    wlist, flist, mlist = find_word_info(word)
    w = w + wlist
    f = f + flist
    m = m + mlist

word_df = pd.DataFrame({'word':w, 'readings':f, 'meanings':m})
word_df
```

```
Start Query:紙
https://jisho.org/word/紙
Requests Made: 1, status 200
紙 / かみ / 1. paper$2. Paper$
https://jisho.org/word/紙-1
Requests Made: 2, status 200
紙 / し / 1. newspaper$
https://jisho.org/word/紙-2
Requests Made: 3, status 200
紙 / かみ / 1. Kami$
https://jisho.org/word/紙-3
Requests Made: 4, status 404
Start Query:手紙
https://jisho.org/word/手紙
Requests Made: 1, status 200
手紙 / てがみ / 1. letter$2. Letter (message)$
https://jisho.org/word/手紙-1
Requests Made: 2, status 404
Start Query:表紙
https://jisho.org/word/表紙
Requests Made: 1, status 200
表紙 / ひょうし / 1. cover (of a book, magazine, etc.); binding$2. appearing on th
e cover of a magazine$3. Book cover$
https://jisho.org/word/表紙-1
Requests Made: 2, status 404
Start Query:紙幣
https://jisho.org/word/紙幣
Requests Made: 1, status 200
紙幣 / しへい / 1. paper money; note; bill$2. Banknote$
https://jisho.org/word/紙幣-1
Requests Made: 2, status 404
Start Query:製紙
https://jisho.org/word/製紙
Requests Made: 1, status 200
製紙 / せいし / 1. papermaking; paper-making; paper making; paper manufacture$
https://jisho.org/word/製紙-1
Requests Made: 2, status 404
Start Query:用紙
https://jisho.org/word/用紙
Requests Made: 1, status 200
用紙 / ようし / 1. blank form$2. sheets of paper; sheet of paper$
https://jisho.org/word/用紙-1
Requests Made: 2, status 404
Start Query:紙面
https://jisho.org/word/紙面
Requests Made: 1, status 200
紙面 / しめん / 1. space on a page (e.g. in a newspaper)$2. surface of paper$3. le
tter; writings; document$
https://jisho.org/word/紙面-1
Requests Made: 2, status 404
Start Query:白紙
https://jisho.org/word/白紙
Requests Made: 1, status 200
白紙 / はくし / 1. white paper; flyleaf$2. blank paper$3. clean slate; lack of pri
or opinion, positions, etc.$4. scratch; beginning$
https://jisho.org/word/白紙-1
Requests Made: 2, status 404
Start Query:紙上
https://jisho.org/word/紙上
Requests Made: 1, status 200
紙上 / しじょう / 1. on paper; in the newspapers; in a letter$
https://jisho.org/word/紙上-1
Requests Made: 2, status 404
Start Query:和紙
```

Out[13]:

	word	readings	meanings
0	紙	かみ	1. paper2.$Paper$
1	紙	し	1. newspaper$
2	紙	かみ	1. Kami$
3	手紙	てがみ	1. letter2.$Letter(message)$
4	表紙	ひょうし	1. cover (of a book, magazine, etc.); binding$...
5	紙幣	しへい	1. paper money; note; bill2.$Banknote$
6	製紙	せいし	1. papermaking; paper-making; paper making; pa...
7	用紙	ようし	1. blank form2.$sheetsofpaper$; $sheetofpaper$
8	紙面	しめん	1. space on a page (e.g. in a newspaper)$2. su...
9	白紙	はくし	1. white paper; flyleaf2.$blankpaper$3. clea...
10	紙上	しじょう	1. on paper; in the newspapers; in a letter$
11	和紙	わし	1. washi; Japanese paper2.$Washi$

In [ ]: