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

This file is intended to scrape the readings/meanings of the whole 20K word list from Jisho (jisho.org). A .py file with temporary saving features should be in the same directory with this notebook. That .py file has the same usage but friendlier if you want to run the program on a server. It really takes **A LONG PERIOD OF TIME** to scrape the whole word list! According to my calculation, in total more than 40K requests are made before the program terminates.

Run all the cells (or the .py file) will generate a "wikiword_table_new.csv" file, containing all word/reading/meaning entries of the 20K word list.

In [2]:
```
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]:
```
df = pd.read_csv('wikitionary_wordlist.csv')
sr = df['word']
```

In [4]:
```
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")
```

In [5]:
```
def get_furigana(soup):
    furi = soup.find('span',class_='furigana')
    #print(list(furi.children))
    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 (f1.get_text() == '') & (t1 in kamei_list):
                furistring += t1
            else:
                furistring += f1.get_text().strip()
            text_loc += 1
    return (furistring, txtstring)
```

In [6]:
```
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 != [])):
            #print(flag)
            tag = flag[0]
            #print(tag)
            #print(tag.next_sibling)
            meanings += tag.get_text() + tag.next_sibling.get_text() + '$'
    return meanings
```

In [8]:
```
wordlist = []
furigana = []
meanings = []

url_base = 'https://jisho.org/word/'

requests = 0
i = 0

for k in np.arange(len(sr)):
    i = 0
    word = sr[k]
#word = '三つ'
#if (True):
    while (True):
        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):
                print("Error: code {} at word {}, i={}".format(response.status_cod
e, word, i-1))
            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)
```

```
Requests Made: 1, status 200
の / の / 1. indicates possessive$2. nominalizes verbs and adjectives$3. substitu
tes for "ga" in subordinate phrases$4. (at sentence-end, falling tone) indicates
a confident conclusion$5. (at sentence-end) indicates emotional emphasis$6. (at
sentence-end, rising tone) indicates question$7. No (kana)$
Requests Made: 2, status 404
Requests Made: 3, status 200
に / に / 1. at (place, time); in; on; during$2. to (direction, state); toward; i
nto$3. for (purpose)$4. because of (reason); for; with$5. by; from$6. as (i.e. i
n the role of)$7. per; in; for; a (e.g. "once a month")$8. and; in addition t
o$9. if; although$10. Ni (kana)$
Requests Made: 4, status 404
Requests Made: 5, status 404
Requests Made: 6, status 200
は / は / 1. topic marker particle$2. indicates contrast with another option (sta
ted or unstated)$3. adds emphasis$4. Ha (kana)$
Requests Made: 7, status 404
Requests Made: 8, status 200
を / を / 1. indicates direct object of action$2. indicates subject of causative
expression$3. indicates an area traversed$4. indicates time (period) over which
action takes place$5. indicates point of departure or separation of action$6. in
dicates object of desire, like, hate, etc.$7. Wo (kana)$
Requests Made: 9, status 404
Requests Made: 10, status 200
が / が / 1. indicates sentence subject (occasionally object)$2. indicates posses
sive (esp. in literary expressions)$3. but; however; still; and$4. regardless o
f; whether (or not)$
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-8-78a16ad0cbde> in <module>
     21         #print(url)
     22 
---> 23         response = get(url)
     24         requests += 1
     25         print("Requests Made: {}, status {}".format(requests, response.s
tatus_code))

c:\python\lib\site-packages\requests\api.py in get(url, params, **kwargs)
     73 
     74     kwargs.setdefault('allow_redirects', True)
---> 75     return request('get', url, params=params, **kwargs)
     76 
     77 

c:\python\lib\site-packages\requests\api.py in request(method, url, **kwargs)
     58     # cases, and look like a memory leak in others.
     59     with sessions.Session() as session:
---> 60         return session.request(method=method, url=url, **kwargs)
     61 
     62 

c:\python\lib\site-packages\requests\sessions.py in request(self, method, url, p
arams, data, headers, cookies, files, auth, timeout, allow_redirects, proxies, h
ooks, stream, verify, cert, json)
    531         }
    532         send_kwargs.update(settings)
--> 533         resp = self.send(prep, **send_kwargs)
    534 
    535         return resp

c:\python\lib\site-packages\requests\sessions.py in send(self, request, **kwarg
s)
    644 
    645         # Send the request
--> 646         r = adapter.send(request, **kwargs)
    647 
    648         # Total elapsed time of the request (approximately)

c:\python\lib\site-packages\requests\adapters.py in send(self, request, stream,
timeout, verify, cert, proxies)
    447                     decode_content=False,
    448                     retries=self.max_retries,
--> 449                     timeout=timeout
    450                 )
    451 

c:\python\lib\site-packages\urllib3\connectionpool.py in urlopen(self, method, u
rl, body, headers, retries, redirect, assert_same_host, timeout, pool_timeout, r
elease_conn, chunked, body_pos, **response_kw)
    670                 body=body,
    671                 headers=headers,
--> 672                 chunked=chunked,
    673             )
    674 

c:\python\lib\site-packages\urllib3\connectionpool.py in _make_request(self, con
n, method, url, timeout, chunked, **httplib_request_kw)
    374         # Trigger any extra validation we need to do.
    375         try:
--> 376             self._validate_conn(conn)
    377         except (SocketTimeout, BaseSSLError) as e:
```

```
In [ ]: pd.DataFrame({
                'word': wordlist,
                'reading': furigana,
                'meaning': meanings
                }).to_csv('wikiword_table_new.csv')
```