```
In [1]: import requests
        import time
        import pandas as pd
```

```
In [2]: url = 'https://www.reddit.com/r/funny/.json'
```

```
In [3]: headers = {'User-agent': 'As Lama Oui 0.1'}

        res = requests.get(url, headers=headers)

        res.status_code

        the_json = res.json()

        the_json
```

```
Out[3]: {'kind': 'Listing',
         'data': {'modhash': '',
          'dist': 25,
          'children': [{'kind': 't3',
            'data': {'approved_at_utc': None,
             'subreddit': 'funny',
             'selftext': '',
             'author_fullname': 't2_mz3ih',
             'saved': False,
             'mod_reason_title': None,
             'gilded': 0,
             'clicked': False,
             'title': 'Gang Violence Going Down',
             'link_flair_richtext': [],
             'subreddit_name_prefixed': 'r/funny',
             'hidden': False,
             'pwls': 6,
             'link_flair_css_class': None,
             'downs': 0,
             'thumbnail_height': 140,
```

```
In [4]: sorted(the_json.keys())

Out[4]: ['data', 'kind']
```

```
In [5]: the_json['kind']

Out[5]: 'Listing'
```

```
In [6]: sorted(the_json['data'].keys())

Out[6]: ['after', 'before', 'children', 'dist', 'modhash']
```

'children' is where posts are

In [7]:
```
len(the_json['data']['children'])
```

Out[7]:
```
25
```

In [8]:
```
the_json['data']['children'][0]
```

Out[8]:
```
{'kind': 't3',
 'data': {'approved_at_utc': None,
  'subreddit': 'funny',
  'selftext': '',
  'author_fullname': 't2_mz3ih',
  'saved': False,
  'mod_reason_title': None,
  'gilded': 0,
  'clicked': False,
  'title': 'Gang Violence Going Down',
  'link_flair_richtext': [],
  'subreddit_name_prefixed': 'r/funny',
  'hidden': False,
  'pwls': 6,
  'link_flair_css_class': None,
  'downs': 0,
  'thumbnail_height': 140,
  'hide_score': False,
  'name': 't3_b810q4',
```

In [9]:
```
pd.DataFrame(the_json['data']['children'])['data'][0].keys()
```

Out[9]:
```
dict_keys(['approved_at_utc', 'subreddit', 'selftext', 'author_fullname',
'saved', 'mod_reason_title', 'gilded', 'clicked', 'title', 'link_flair_ri
chtext', 'subreddit_name_prefixed', 'hidden', 'pwls', 'link_flair_css_cla
ss', 'downs', 'thumbnail_height', 'hide_score', 'name', 'quarantine', 'li
nk_flair_text_color', 'author_flair_background_color', 'subreddit_type',
'ups', 'domain', 'media_embed', 'thumbnail_width', 'author_flair_template
_id', 'is_original_content', 'user_reports', 'secure_media', 'is_reddit_m
edia_domain', 'is_meta', 'category', 'secure_media_embed', 'link_flair_te
xt', 'can_mod_post', 'score', 'approved_by', 'thumbnail', 'edited', 'auth
or_flair_css_class', 'author_flair_richtext', 'gildings', 'post_hint', 'c
ontent_categories', 'is_self', 'mod_note', 'created', 'link_flair_type',
'wls', 'banned_by', 'author_flair_type', 'contest_mode', 'selftext_html',
'likes', 'suggested_sort', 'banned_at_utc', 'view_count', 'archived', 'no
_follow', 'is_crosspostable', 'pinned', 'over_18', 'preview', 'media_onl
y', 'can_gild', 'spoiler', 'locked', 'author_flair_text', 'visited', 'num
_reports', 'distinguished', 'subreddit_id', 'mod_reason_by', 'removal_rea
son', 'link_flair_background_color', 'id', 'is_robot_indexable', 'report_
reasons', 'author', 'num_crossposts', 'num_comments', 'send_replies', 'wh
itelist_status', 'mod_reports', 'author_patreon_flair', 'author_flair_tex
t_color', 'permalink', 'parent_whitelist_status', 'stickied', 'url', 'sub
reddit_subscribers', 'created_utc', 'media', 'is_video'])
```

In [10]:
```
features = ['subreddit', 'selftext', 'author_fullname', 'title', 'subreddit
```

the four pieces of content we need:

1. The title of the thread
2. the subreddit that the thread conrresponds to
3. the length of time it has been up on Reddit
4. the number of comments on the thread

In [11]:
```
#the id of the last post in this list
the_json['data']['after']
```

Out[11]:
```
't3_b8jyxy'
```

In [12]:
```
[post['data']['name'] for post in the_json['data']['children']]
```

```
...
```

In [13]:
```
url = 'https://www.reddit.com/r/funny/.json?AFTER=t3_b80433'
```

In [14]:
```
param = {'after':'t3_b8jyxy'}
```

In [15]:
```
requests.get(url, params=param, headers=headers)
```

Out[15]:
```
<Response [200]>
```

In [16]:
```
features = ['subreddit', 'selftext', 'author_fullname', 'title',
            'content_categories', 'name','is_self', 'suggested_sort',
            'subreddit_id', 'category','id']
```

In [17]:

```
posts = []
after = None
for i in range(100):
    print(i)
    if after == None:
        params = {}
    else:
        params = {'after':after}
    url = 'https://www.reddit.com/r/funny/.json'
    res = requests.get(url, params=params, headers=headers)
    if res.status_code == 200:
        the_json = res.json()
        for j in range(len(the_json['data']['children'])):
            posts.append({'subreddit': the_json['data']['children'][j]['dat
                          'selftext': the_json['data']['children'][j]['data
                          'author_fullname': the_json['data']['children'][j
                          'title': the_json['data']['children'][j]['data'][
                          'content_categories': the_json['data']['children'
                          'name': the_json['data']['children'][j]['data']['
                          'is_self': the_json['data']['children'][j]['data'
                          'suggested_sort': the_json['data']['children'][j]
                          'subreddit_id': the_json['data']['children'][j]['
                          'category': the_json['data']['children'][j]['data
                          'id': the_json['data']['children'][j]['data']['id
                         })
        after = the_json['data']['after'] #check inside or outside
    else:
        print(res.status_code)
        break
    time.sleep(1)
len(posts)
df = pd.DataFrame(posts)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

```
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
```

```
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

In [19]:

```
len(posts)
```

Out[19]:

```
2477
```

In [20]:

```
funny = pd.DataFrame(posts)
```

In [21]:

```
funny = funny.to_csv('funny_subreddit.csv')
```

In [22]:

```
funny
```

In [ ]: