Contents

0.1 Download Sitemap URLs

```
from lxml import etree

def download_sitemap_urls(sitemap_url):
    # Parse remote sitemap
    tree = etree.parse(sitemap_url)
    # Extract namespace
```

```
    ns = tree.getroot().nsmap[None]
    # Extract URLs using xpath and namespace
    urls = tree.xpath('/s:urlset/s:url/s:loc/text()',
        namespaces={'s': ns})

    return urls
```

0.2 Write to csv

```
import csv

# data is a list of lists [[]]
def write_to_csv(filename, data)
    with open(filename, 'w') as f:
        csv_export = csv.writer(f)
        csv_data = []
        for row in data:
            csv_row = [item.encode("utf-8")
                if isinstance(item, unicode) else str(item)
                for item in row]

            csv_data.append(row)

        csv_export.writerows(csv_data)
        return True

    return False
```

0.3 Create sitemap

0.4 Write to Excel

0.5 Write to Google Spreadsheet

0.6 Check robots.txt

0.7 Bulk robots.txt checker

0.8 Google Webmaster Tools crawl errors

0.9 Find proxies

0.10 Test proxies

0.11 Search Twitter

```
from twitter import *
import json
consumer_key = 'Bm6HALfQPWJmnLATt8PGQu1pF'
consumer_secret =  'avoFUUhHD9OZ4PJga9nK9AKqIkV3C8XxO06xGKf65ocVDX0zJw'
access_token = '373618025-            E662cyYTBNDECgYelj4MZR9Gff6E6gSjohTeqGZe'
access_token_secret =  'GOAhvgzOuRQkVo6v1DXLr1PHTZUjbVmCQwgV8Ch8srCrM'
twitter_api = Twitter(auth=OAuth(access_token,    access_token_secret, consumer_key, consu
WORLD_WOE_ID = 1
US_WOE_ID = 23424977
#world_trends = api.trends.place(_id=WORLD_WOE_ID)
#us_trends = api.trends.place(_id=US_WOE_ID)
#print world_trends
#print
#print us_trends
q = '#seo'
count = 100
search_results = twitter_api.search.tweets(q=q,       count=count)
statuses = search_results['statuses']

Iterate through 5 more batches of results by following the cursor

for _ in range(10):
    print "Length of statuses", len(statuses)
    try:
     next_results = search_results['search_metadata']['next_results']
  except KeyError, e: # No more results when next_results doesn't exist
```

```
        break

Create a dictionary from next_results, which has the following form:
    # ?max_id=313519052523986943&q=NCAA&include_entities=1
   kwargs = dict([ kv.split('=') for kv in next_results[1:].split("&") ])

search_results = twitter_api.search.tweets(**kwargs)
statuses += search_results['statuses']

# Show one sample search result by slicing the list...
print json.dumps(statuses[0], indent=1)
```

0.12 Get data from Google Analytics

0.13 Check website is up

0.14 Get domain information

0.15 Harvest email addresses (Warning)

0.16 Get Moz metrics

0.17 Analyse Logs for Google Bots

0.18 Duplication Test (supplementary index)

0.19 Get Adwords data

0.20 Send emails with gmail

0.21 Create a spider

0.22 Crawl a website

0.23 On-Page SEO Checker