Cheatography

# Network data mining Cheat Sheet

by gonz95alo via cheatography.com/152363/cs/32833/

| Web scraping with urllib | |
|---|---|
| Import basic functions | `from urllib.request import urlopen, urlretrieve, Request` |
| Request webpage 'example.com' | `raw_request = Request('https://example.com')` |
| Add headers (I) | `raw_request.add_header('User-Agent', 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:78.0) Gecko/20100101 Firefox/78.0')` |
| Add headers (II) | `raw_request.add_header('Accept', 'text/html,application/xhtml+xml,application/xml;q=-0.9,image/webp,/;q=0.8')` |
| Get HTML code as a string | `html = urlopen(raw_request).read().decode("utf-8")` |
| Download file | `urlretrieve(fileURL, 'file_name_in_destination')` |

## Python libraries

| | |
|---|---|
| urllib: URL handling | `import urllib` |
| BeautifulSoup: HTML/XML parser | `from bs4 import BeautifulSoup` |
| Regular expressions: pattern matching | `import re` |
| NetworkX: network analysis | `import networkx as nx` |

## Websites of interest

| | |
|---|---|
| Stanford Network Analysis Project | http://snap.stanford.edu/ |
| Koblenz Network Collection | http://konect.cc/ |
| Network Repository | https://networkrepository.com/ |

## Graph dataset formats

| | | |
|---|---|---|
| GML (.gml) | *Custom structure* | `G = nx.read_gml(path)` |
| Pajek (.net) | *Custom structure* | `G = nx.read_pajek(path)` |
| JSON (.json) | *Custom structure* | `G = nx.read_json(path)` |
| Plain text | `node1 node2 weight` | Manual |
| Multilayer v1 (.) | `layer node1 node2 weight` | Manual |
| Multilayer v2 (.) | `layer1 layer2 node1 node2 weight` | Manual |

Those formats without a specific function are usually easy to parse manually.

## Beautiful Soup HTML parsing

## String slicing tricks

| | |
|---|---|
| Split by character | `'a_string'.split('_')` |
| Join with a character | `'_'.join(['a', 'string'])` |
| Capitalize | `foo.capitalize()` |
| Capitalize, lower, upper | `foo.capitalize(), foo.lower(), foo.upper()` |
| Find, count | `foo.find('e'), foo.count('e')` |
| Replace | `'a_string.replace('string', 'banana')'` |

## Regular Expressions