# Section 1: Effective Webscraping

Masha Krupenkin<sup>1</sup>

April 16, 2019

<sup>&</sup>lt;sup>1</sup>Stanford University

Course website: https://github.com/justingrimmer/tad\_19

- Course website: https://github.com/justingrimmer/tad\_19
- ▶ Please sign up for section on Canvas!

- Course website: https://github.com/justingrimmer/tad\_19
- ▶ Please sign up for section on Canvas!
- ▶ HW1 due on 4/22!

▶ Urllib package = easy to use for simple scraping tasks

- ▶ Urllib package = easy to use for simple scraping tasks
- ► BeautifulSoup parses html

- ► Urllib package = easy to use for simple scraping tasks
- BeautifulSoup parses html
- Many scraping tasks are not simple!

# Difficult Scraping Tasks

# Difficult Scraping Tasks

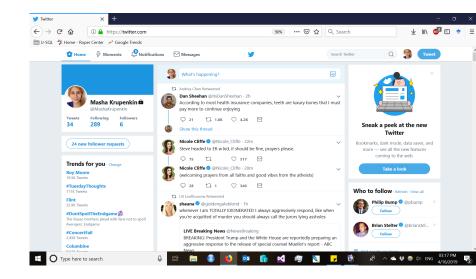
1. Scraping complicated webpages

## Difficult Scraping Tasks

- 1. Scraping complicated webpages
- 2. Scraping a large number of webpages

# Complex Webpages w/ Selenium

# Scraping Complicated Webpages



## Naive approach...

```
*untitled-1 🖾
  1 import urllib.request
  2 from bs4 import BeautifulSoup
  4 with urllib.request.urlopen("https://twitter.com/") as response:
       html = response.read()
 7 soup = BeautifulSoup(html, "html.parser")
 8 print(soup)
```

Urllib package will not give you tweets!

- Urllib package will not give you tweets!
- Infinite scrolling poses additional challenge

- Urllib package will not give you tweets!
- Infinite scrolling poses additional challenge
- Ethics: scrape at your own discretion!

#### Selenium Demo

To the Python!

#### Selenium Install

```
https://selenium-
python.readthedocs.io/installation.html
```

#### Excellent Selenium Tutorial

https://medium.com/@dawranliou/twitter-scraper-tutorial-with-python-requests-beautifulsoup-and-selenium-part-2-b38d849b07fe

▶ **Pro:** Can scrape webpages that hide content

- ▶ **Pro:** Can scrape webpages that hide content
- ▶ **Pro:** Can fill in forms, searches, etc (NO CAPCHA)

- Pro: Can scrape webpages that hide content
- ▶ **Pro:** Can fill in forms, searches, etc (NO CAPCHA)
- ► Con: Slow and inefficient

- Pro: Can scrape webpages that hide content
- ▶ **Pro:** Can fill in forms, searches, etc (NO CAPCHA)
- ► Con: Slow and inefficient
- ▶ Con: Potential to miss data

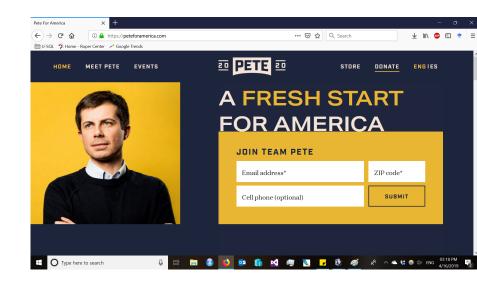
Selenium: Final Verdict

Great for scraping (relatively) small number of highly complex pages

On the other end of the spectrum...

# **Fast Scraping with Scrapy**

# Mayor Pete's Website



## Scrapy Demo

# Command Prompt Microsoft Windows [Version 10.0.17134.706] (c) 2018 Microsoft Corporation. All rights reserved. C:\Users\mkrup>cd C:\Users\mkrup\Dropbox\Misc\_files\Docs\_Grad\_School\Teaching\TAD19 C:\Users\mkrup\Dropbox\Misc\_files\Docs\_Grad\_School\Teaching\TAD19>scrapy startproject mayorpete\_

#### Scrapy Demo

#### Command Prompt

Microsoft Windows [Version 10.0.17134.706]

- (c) 2018 Microsoft Corporation. All rights reserved.
- C:\Users\mkrup>cd C:\Users\mkrup\Dropbox\Misc\_files\Docs\_Grad\_School\Teaching\TAD19
- C:\Users\mkrup\Dropbox\Misc files\Docs Grad School\Teaching\TAD19>cd mayorpete
- C:\Users\mkrup\Dropbox\Misc\_files\Docs\_Grad\_School\Teaching\TAD19\mayorpete>scrapy crawl pete

# **Scrapy Spiders**

To the Python!

# Scrapy Install Guide

https://docs.scrapy.org/en/latest/intro/install.html

# In-depth Scrapy Tutorial

https://docs.scrapy.org/en/latest/intro/tutorial.html

▶ **Pro:** Incredibly fast Python scraper!

- Pro: Incredibly fast Python scraper!
- ▶ **Pro:** Can crawl full websites easily

- ▶ **Pro:** Incredibly fast Python scraper!
- ▶ **Pro:** Can crawl full websites easily
- ► Con: Too efficient use time.sleep()!

- Pro: Incredibly fast Python scraper!
- ▶ **Pro:** Can crawl full websites easily
- ► Con: Too efficient use time.sleep()!
- ► Con: Cannot scrape complex pages

Scrapy: Final Verdict

Great for bulk scraping

► Choose the correct scraper for the problem

- ► Choose the correct scraper for the problem
- Urllib good for beginners

- ► Choose the correct scraper for the problem
- Urllib good for beginners
- Speed vs complexity tradeoff