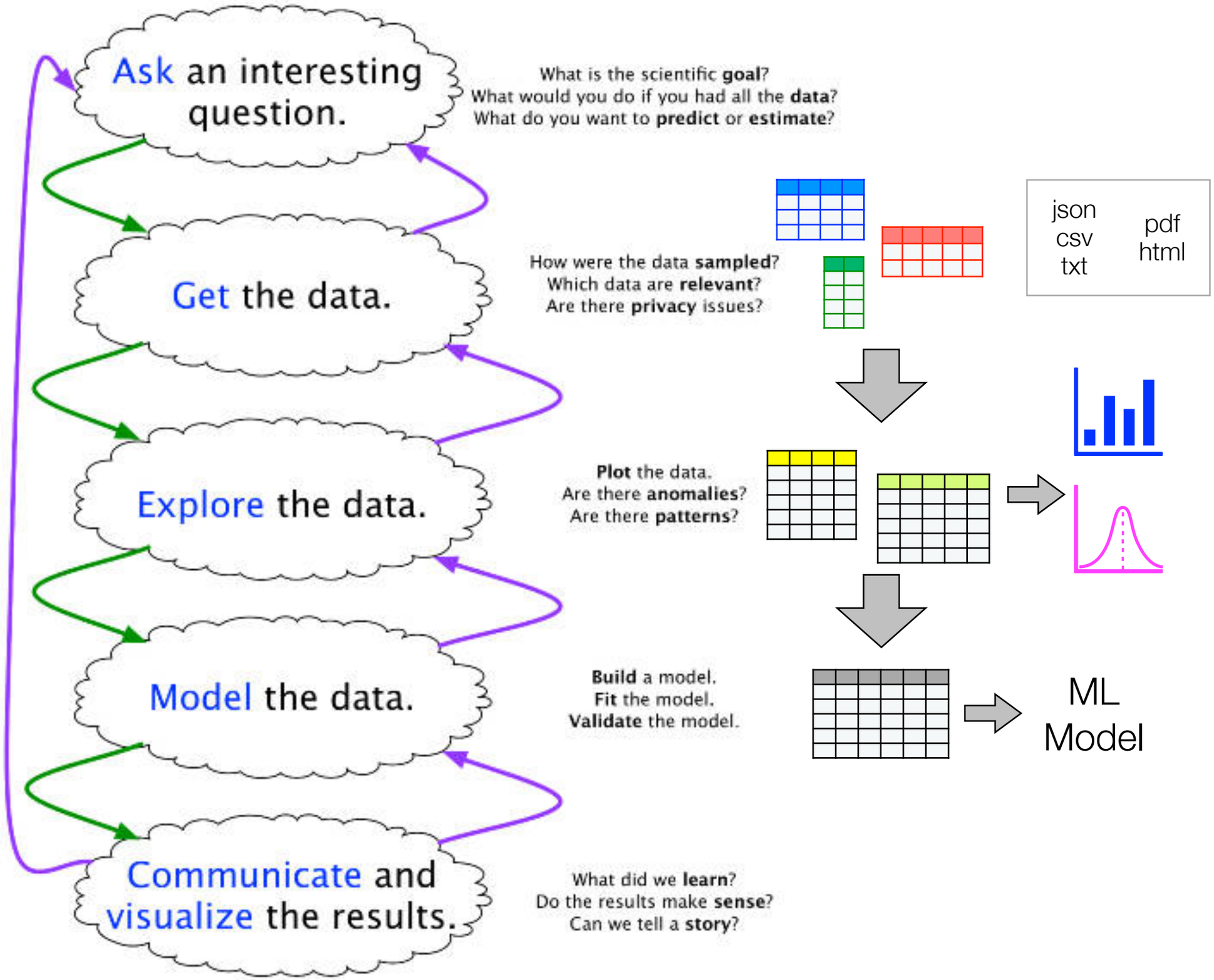
2110446 - Data Science and Data Engineering

Data Extraction

Asst.Prof. Natawut Nupairoj, Ph.D.
Department of Computer Engineering
Chulalongkorn University
natawut.n@chula.ac.th

CHULA ENGINEERING
Foundation toward Innovation

The Data Science Process



Joe Blitzstein and Hanspeter Pfister, created for the Harvard data science course http://cs109.org/.

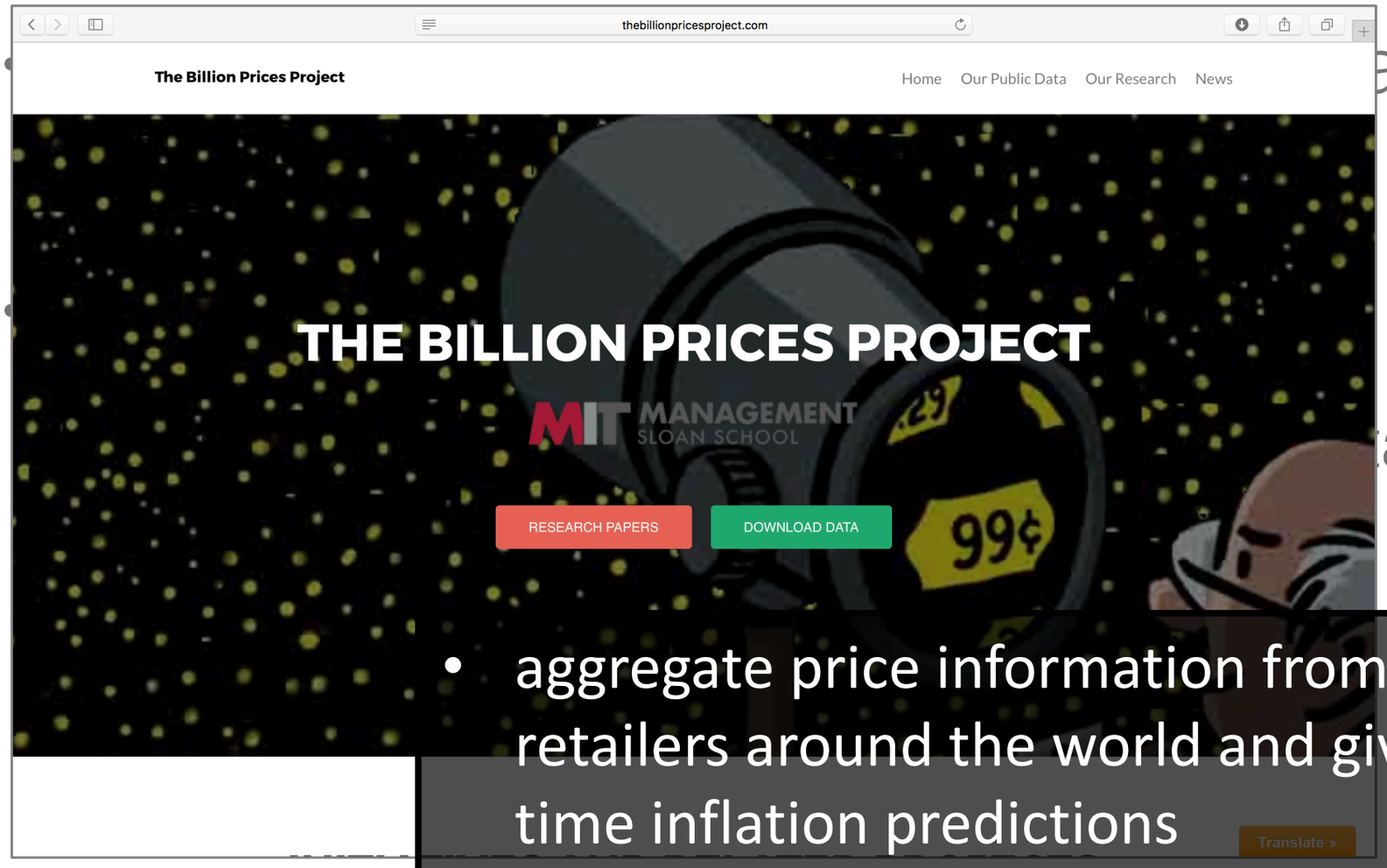
Source: https://www.kdnuggets.com/2016/03/data-science-process-rediscovered.html

Internet Data Sources

- Internet provides lots of data sources
  - Financial
  - Weather
  - Sports
  - News
- Google creates business by utilizing Internet information
- Most data sources are human-readable, it is also possible to extract information systematically with program

CHULA ENGINEERING
Foundation toward Innovation

MIT's Billion Prices Project



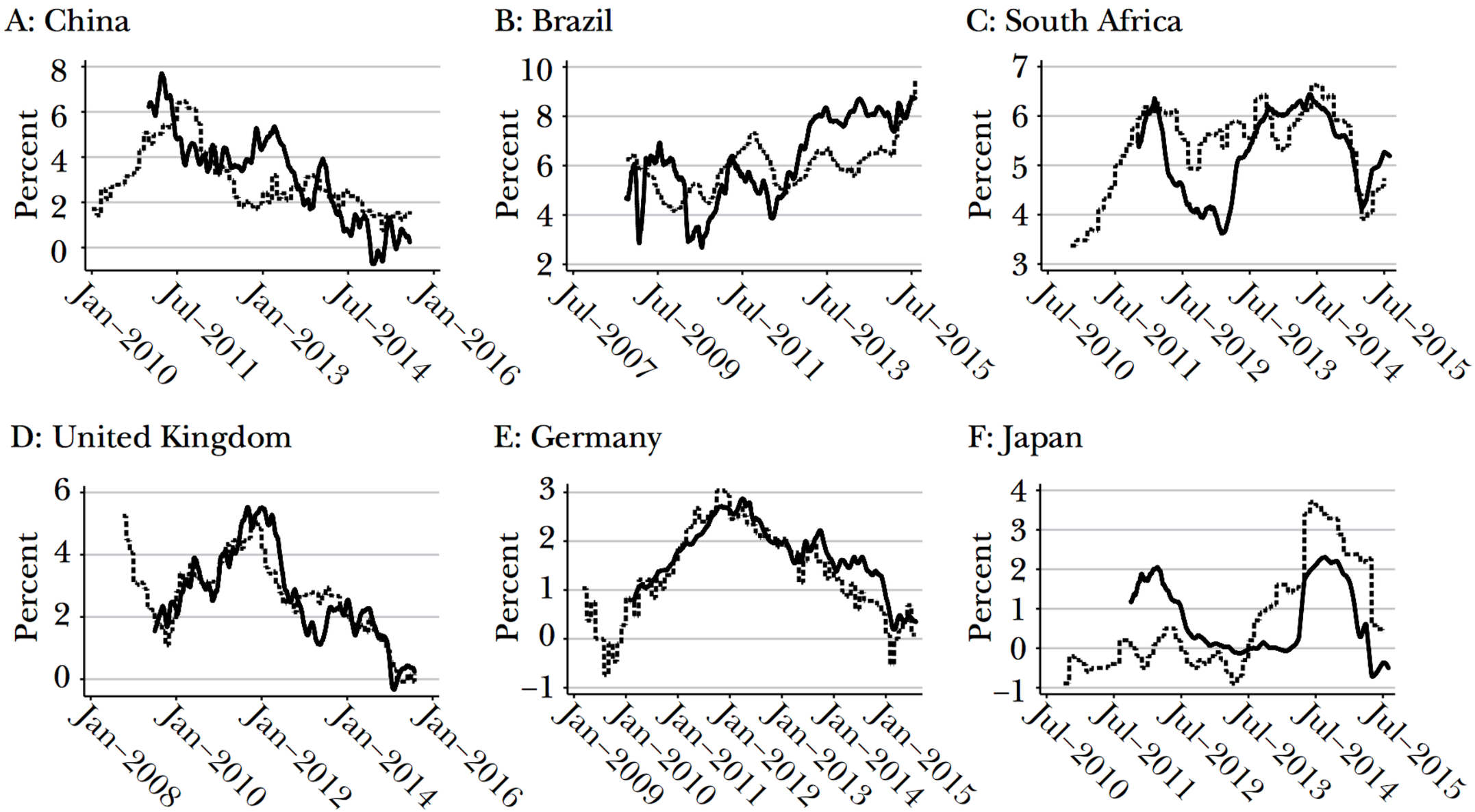
ata structures and

a with

- aggregate price information from multitude of online retailers around the world and gives real time inflation predictions
- monitor daily price fluctuations of ~5 million items sold by ~300 online retailers in more than 70 countries

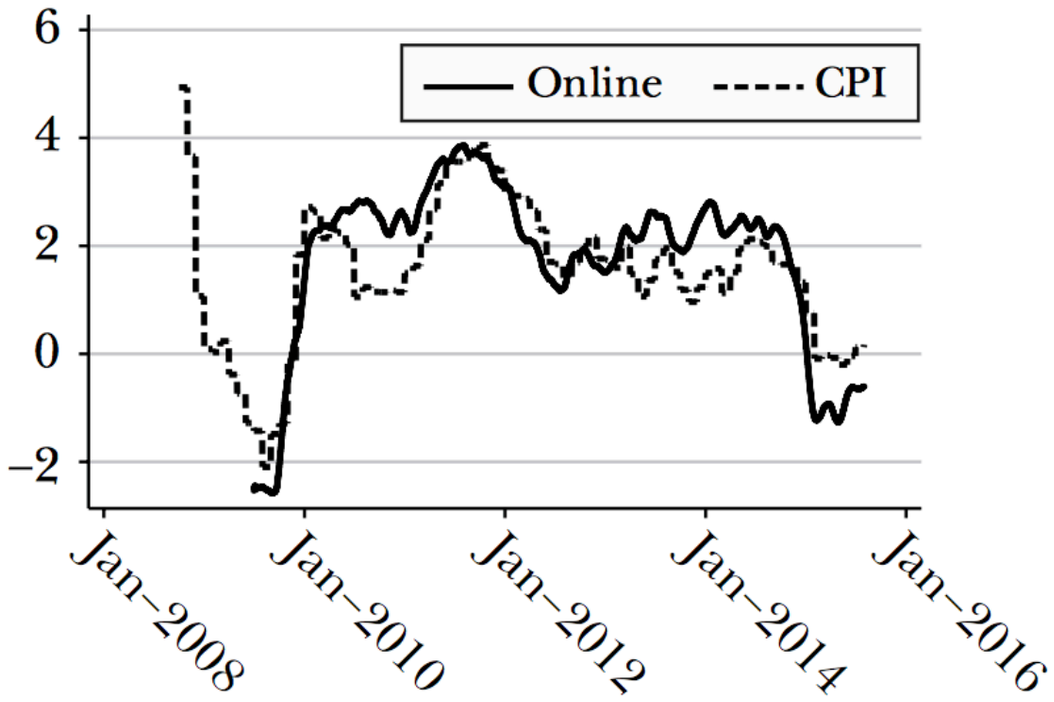
Cavallo, A., & Rigobon, R. (2016). The billion prices project: Using online prices for measurement and research. *The Journal of Economic Perspectives*, *30*(2), 151-178.

Annual Inflation Rate: Online vs. CPI

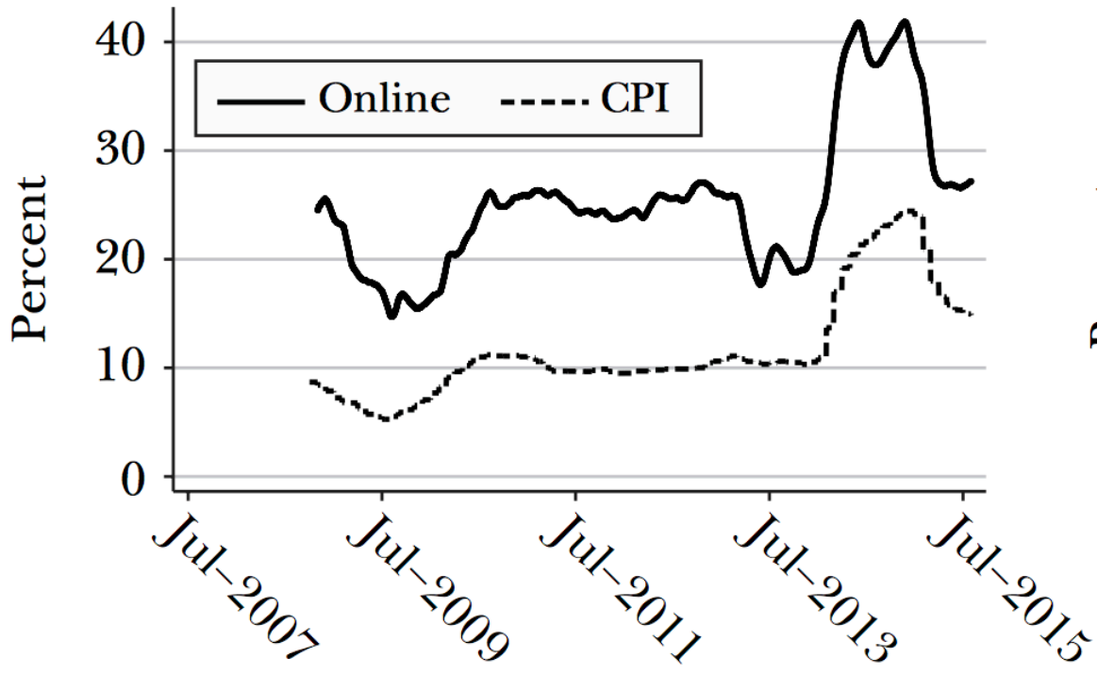


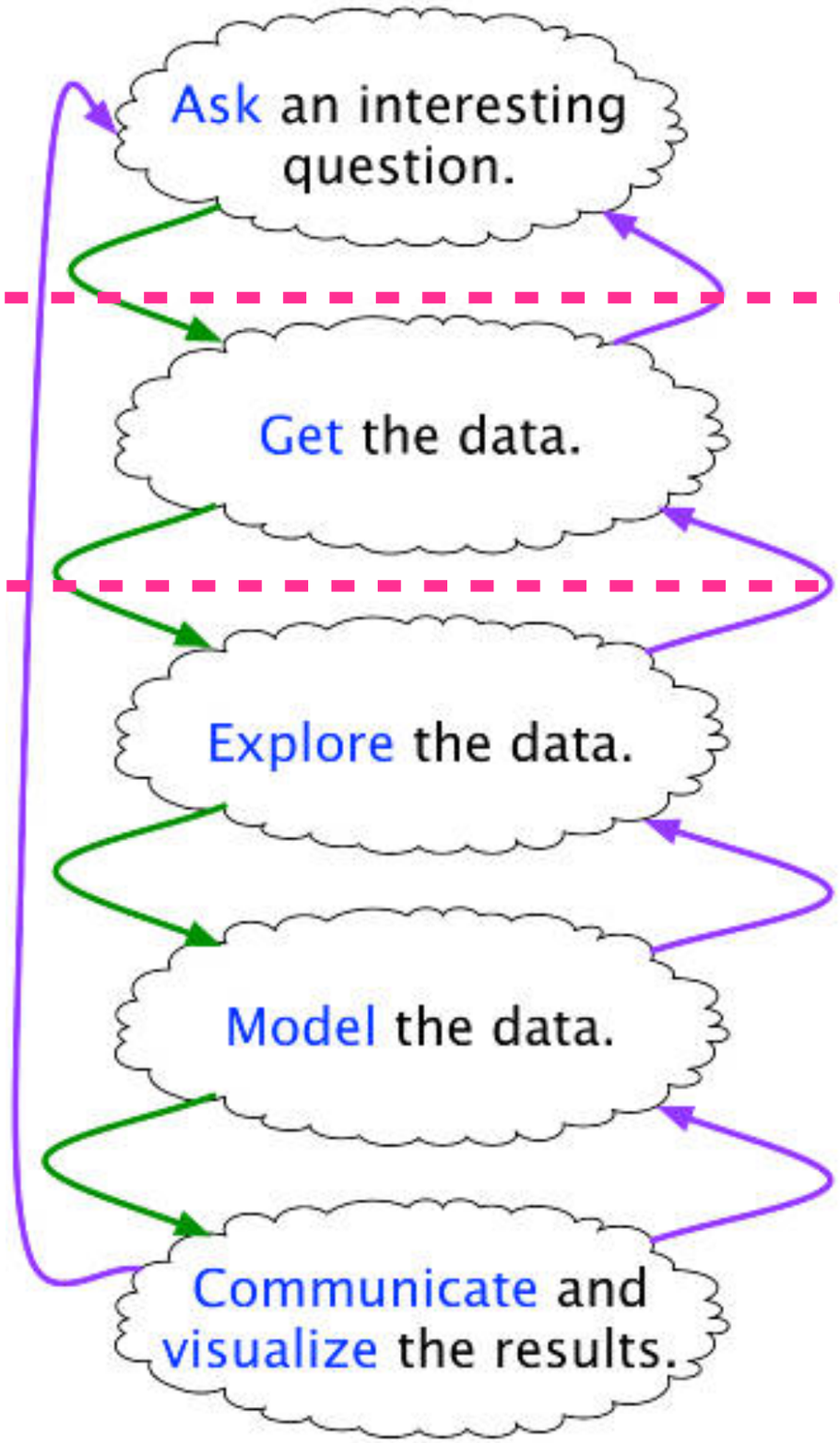
Annual Inflation Rate: Online vs. CPI

USA



Argentina





Scrapy
Requests
BeautifulSoup
Selenium

pandas
statsmodels
NLTK
matplotlib
seaborn
folium
Spark

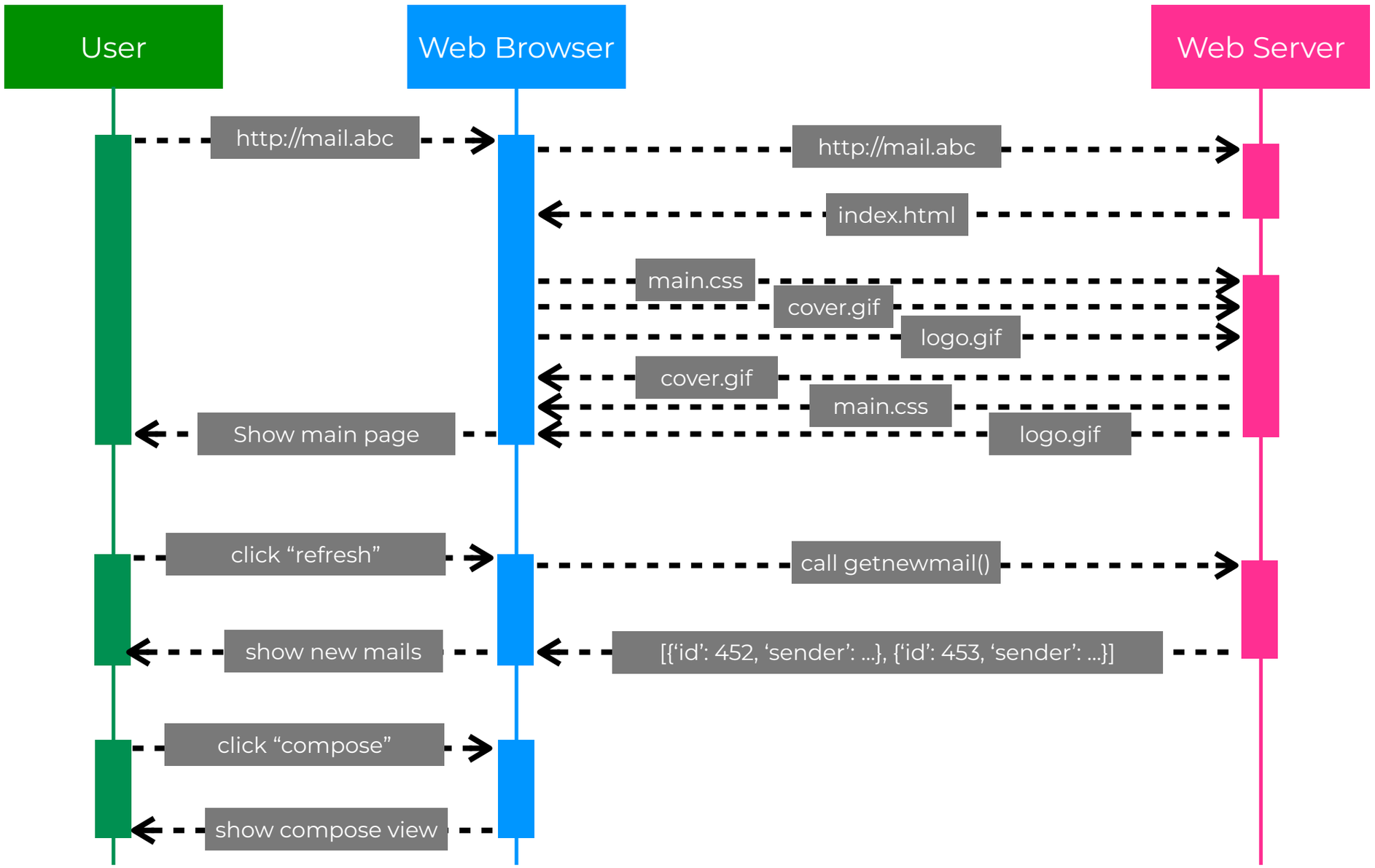
scikit-learn
Theano
Tensorflow
SparkML

Seaborn
Bokeh
Basemap
NetworkX

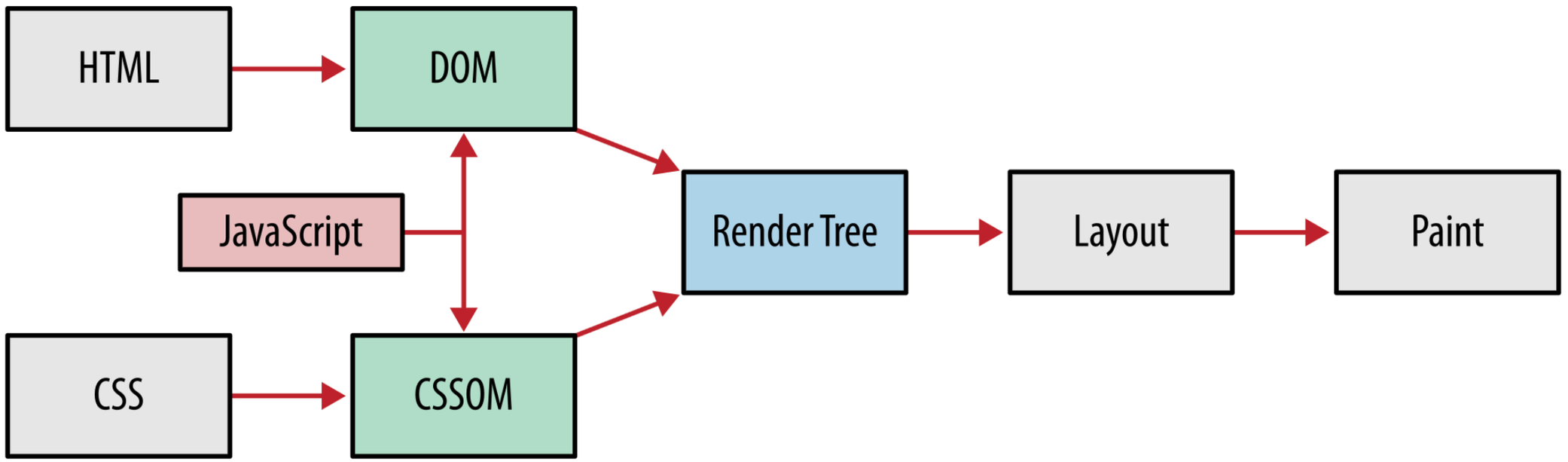
Data Extraction Steps

- Data Accessing
  - Reading a file
  - Issue a HTTP request
  - Crawling a web site using Scrappy
  - Call REST API using Request
  - Call Third-Party API using compatible library
- Data Parsing
  - Parsing text using regular expression
  - Parsing docx using python-docx
  - Parsing excel using Pandas
  - Parsing HTML page using BeautifulSoup
  - Parsing JSON string using JSON

Understand Web Page Mechanism



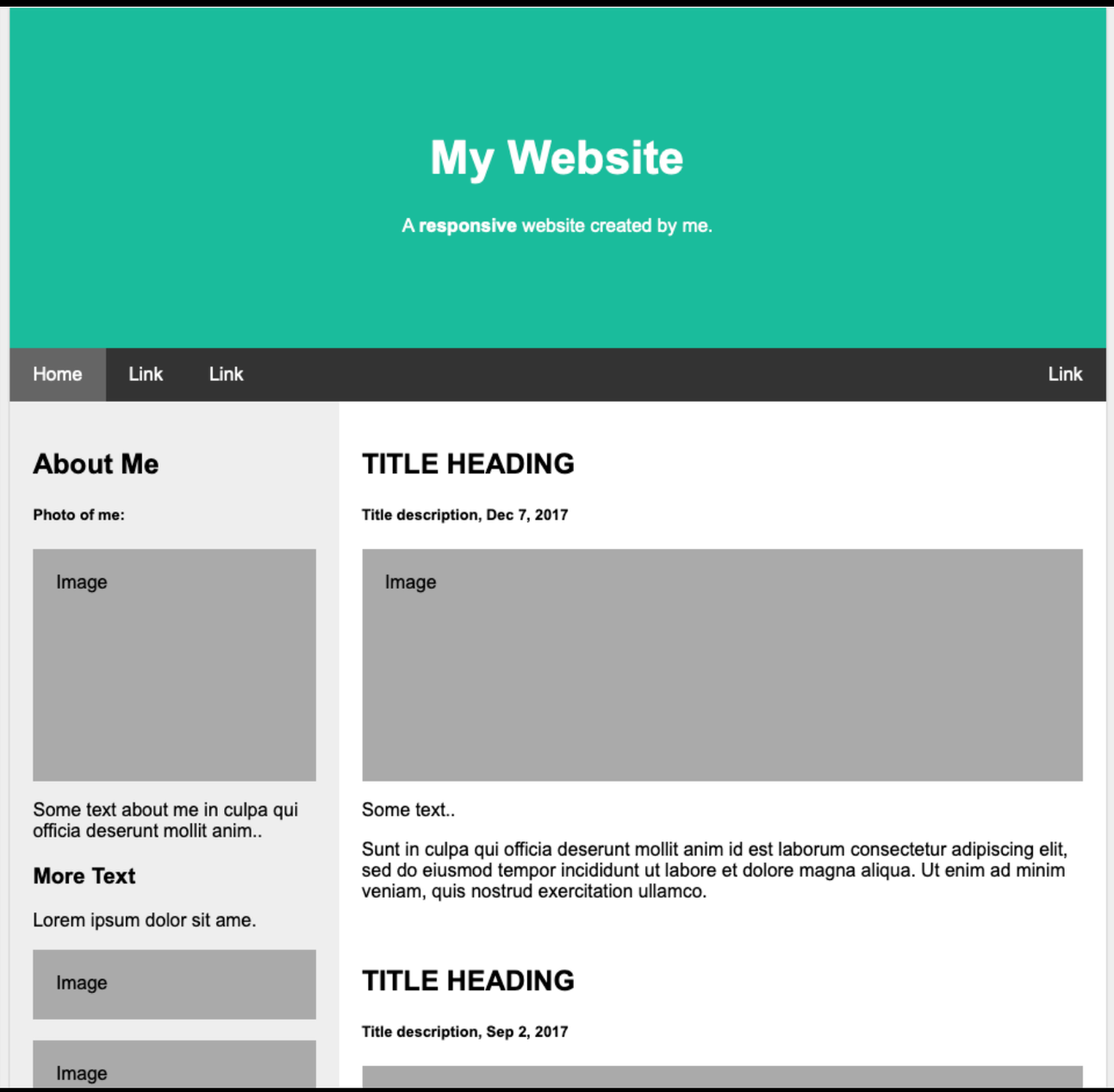
Understand Web Page Mechanism



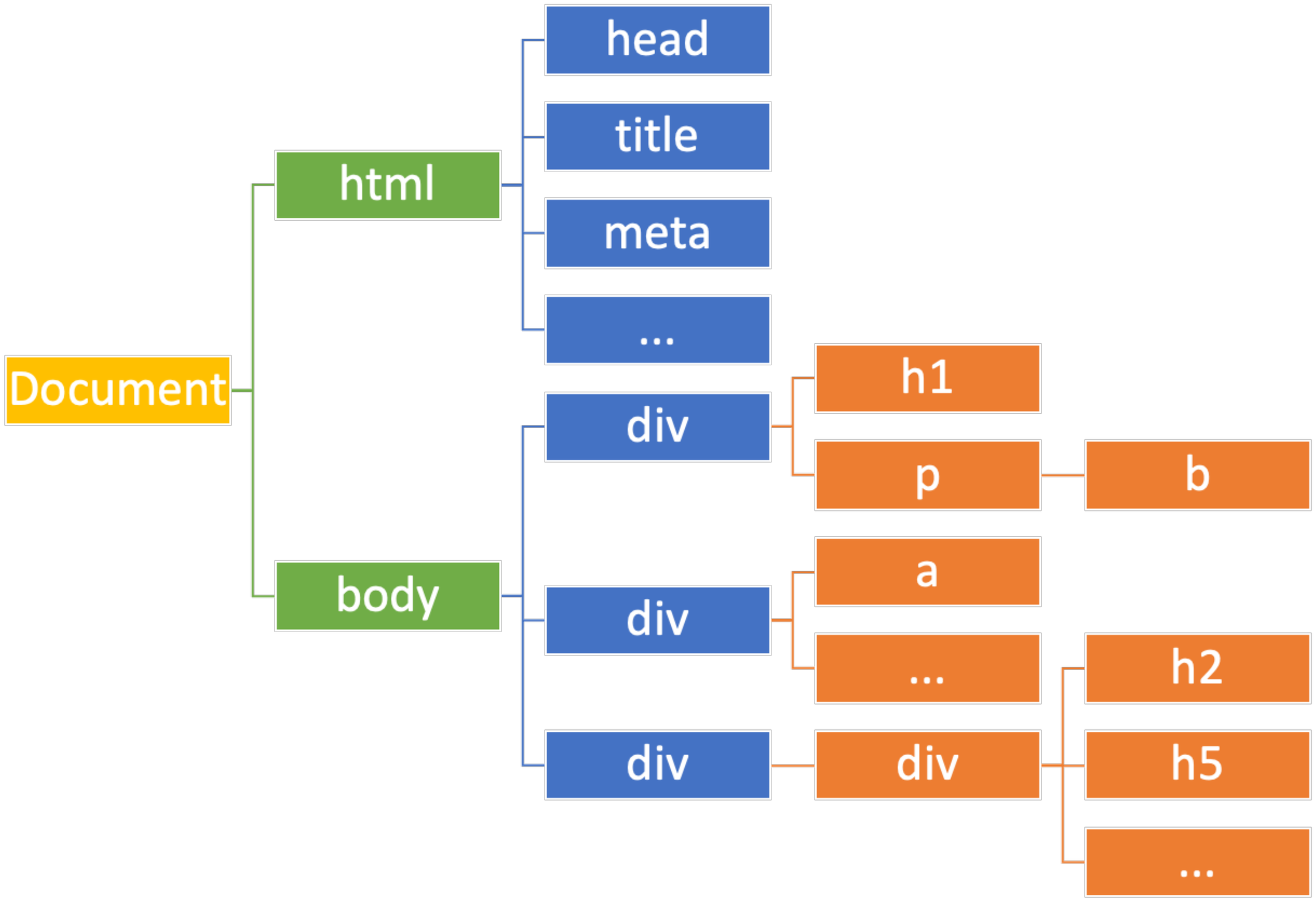
Source: https://hpbn.co/primer-on-web-performance/

https://www.w3schools.com/howto/tryit.asp?filename=tryhow_css_example_website

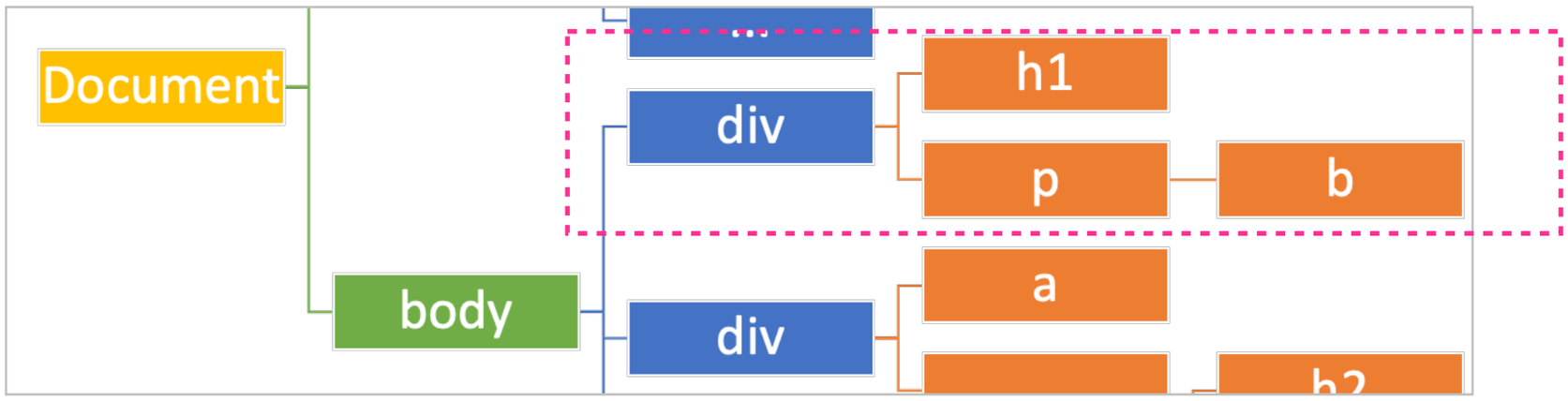
```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Page Title</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

/* Style the body */
body {
  font-family: Arial, Helvetica, sans-serif;
  margin: 0;
}

/* Header/logo Title */
.header {
  padding: 80px;
  text-align: center;
  background: #1abc9c;
  color: white;
}

/* Increase the font size of the heading */
.header h1 {
  font-size: 40px;
}

/* Sticky navbar - toggles between relative and fixed, depending on the scroll position. It is
positioned relative until a given offset position is met in the viewport - then it "sticks" in place
(like position:fixed). The sticky value is not supported in IE or Edge 15 and earlier versions.
However, for these versions the navbar will inherit default position */
.navbar {
  overflow: hidden;
  background-color: #333;
  position: sticky;
  position: -webkit-sticky;
  top: 0;
}

/* Style the navigation bar links */
.navbar a {
  float: left;
  display: block;
  color: white;
  text-align: center;
  padding: 14px 20px;
  text-decoration: none;
}
```



```
<body>

<div class="header">
  <h1>My Website</h1>
  <p>A <b>responsive</b> website created by me.</p>
</div>

<div class="navbar">
  <a href="#" class="active">Home</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#" class="right">Link</a>
</div>

<div class="row">
  <div class="side">
    <h2>About Me</h2>
    <h5>Photo of me:</h5>
    <div class="fakeimg" style="height:200px;">Image</div>
    <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
    <h3>More Text</h3>
    <p>Lorem ipsum dolor sit ame.</p>
    <div class="fakeimg" style="height:60px;">Image</div><br>
    <div class="fakeimg" style="height:60px;">Image</div>
  </div>
  <div class="main">
    <h2>TITLE HEADING</h2>
    <h5>Title description, Dec 7, 2017</h5>
    <div class="fakeimg" style="height:200px;">Image</div>
    <p>Some text..</p>
    <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed
do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco.</p>
    <br>
    <h2>TITLE HEADING</h2>
    <h5>Title description, Sep 2, 2017</h5>
    <div class="fakeimg" style="height:200px;">Image</div>
    <p>Some text..</p>
    <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed
do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco.</p>
  </div>
</div>

<div class="footer">
  <h2>Footer</h2>
</div>

</body>
```


Document
html
head
title
meta
...
body
div
h1
p
b
div
a
...
div
div
h2
h5
...

DOM Node Example




```
<div class="header">
  <h1>My Website</h1>
  <p>A <b>responsive</b> website created by me.</p>
</div>
```

- Node ‘div’ has 2 children and 1 attribute (key=‘class’, value=‘header’)

CHULA ENGINEERING
Foundation toward Innovation

Parsing A Simple Webpage

- Applicable for a simple page only
  - No content rendered by javascript
  - Parse HTML with BeautifulSoup
- BeautifulSoup allows
  - Parsing HTML
  - Accessing and Navigating the DOM tree
  - Getting information from nodes

mentation »

Beautiful Soup Documentation

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

These instructions illustrate all major features of Beautiful Soup 4, with examples. I show you what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.

This document covers Beautiful Soup version 4.9.1. The examples in this documentation should work the same way in Python 2.7 and Python 3.2.

You might be looking for the documentation for Beautiful Soup 3. If so, you should know that Beautiful Soup 3 is no longer being developed and that support for it will December 31, 2020. If you want to learn about the differences between Beautiful Soup 3 see Porting code to BS4.

This documentation has been translated into other languages by Beautiful Soup users:

- 这篇文档当然还有中文版.
- このページは日本語で利用できます(外部リンク)
- 이 문서는 한국어 번역도 가능합니다.
- Este documento também está disponível em Português do Brasil.
- Эта документация доступна на русском языке.

Getting help

If you have questions about Beautiful Soup, or run into problems, send mail to the discussi
lem involves parsing an HTML document, be sure to mention what the diagnose() fun
document.

Quick Start

Here's an HTML document I'll be using as an example throughout this document. It's part
Wonderland:

```
html_doc = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters; and their names
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>

<p class="story">...</p>
"""
```

Running the "three sisters" document through Beautiful Soup gives us a `BeautifulSoup` ob
the document as a nested data structure:

localhost

jupyter 1 - Basic Webpage Scraping (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

Basic Webpage Scraping

Webpage scraping consists of two steps: crawling and parsing. In this tutorial, we focus on parsing HTML data. Beautifulsoup is a powerful tool to process static HTML. More details can be found at https://www.crummy.com/software/BeautifulSoup/bs4/doc/

To simplify our learning, we will use a simple example from W3Schools: https://www.w3schools.com/howto/tryit.asp?filename=tryhow_css_example_website

In [ ]:
```
import sys
IN_COLAB = 'google.colab' in sys.modules
if IN_COLAB:
    !wget https://www.dropbox.com/s/w5khgpro1ym3icg/simple_page.html?dl=
```

In [ ]:
```
with open('simple_page.html') as f:
    html = f.read()
```

In [ ]:
```
html
```

In [ ]:
```
from bs4 import BeautifulSoup
from bs4.element import Tag
from IPython.core.display import HTML
```

In [ ]:
```
soup = BeautifulSoup(html, "lxml")
print(soup.prettify())
```

Example: Extracting references from a wiki page

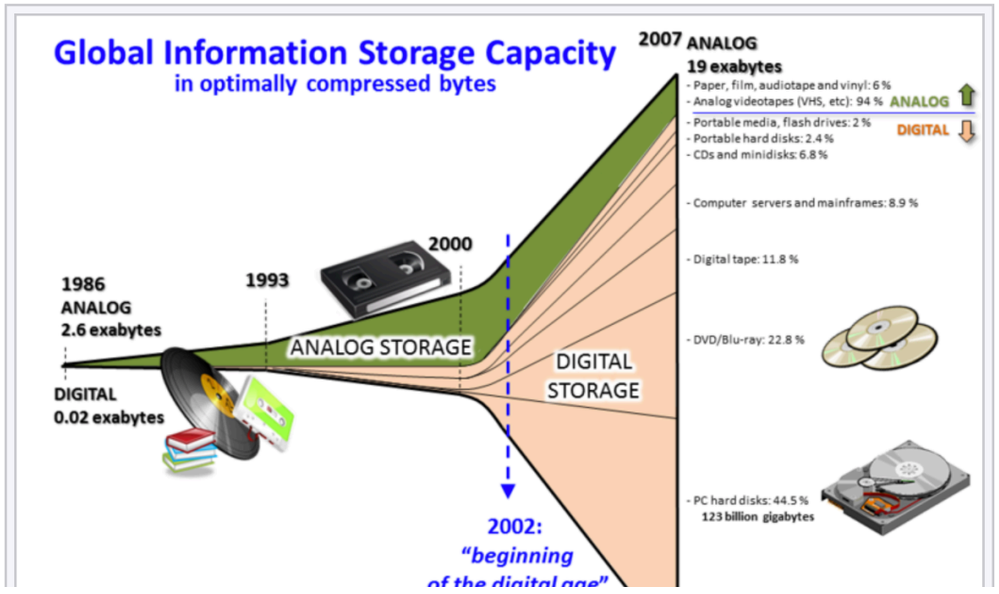
Big data

From Wikipedia, the free encyclopedia

This article is about large collections of data. For the band, see Big Data (band). For buying and selling of personal and consumer data, see Surveillance capitalism.

This article **may contain an excessive number of citations**. Please consider removing references to unnecessary or disreputable sources, merging citations where possible, or, if necessary, flagging the content for deletion. *In particular many references are "spammed" here for promotional purposes. These need to be removed.*. *(November 2019) (Learn how and when to remove this template message)*

Big data is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software. Data with many cases (rows) offer greater statistical power, while data with higher complexity (more attributes or columns) may lead to a higher false discovery rate.[2] Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating, information privacy and data source. Big data was



Big data is a buzzword and a "vague term",[197][198] but at the same time an "obsession"[198] with entrepreneurs, consultants, scientists and the media. Big data showcases such as Google Flu Trends failed to deliver good predictions in recent years, overstating the flu outbreaks by a factor of two. Similarly, Academy awards and election predictions solely based on Twitter were more often off than on target. Big data often poses the same challenges as small data; adding more data does not solve problems of bias, but may emphasize other problems. In particular data sources such as Twitter are not representative of the overall population, and results drawn from such sources may then lead to wrong conclusions. Google Translate—which is based on big data statistical analysis of text—does a good job at translating web pages. However, results from specialized domains may be dramatically skewed. On the other hand, big data may also introduce new problems, such as the multiple comparisons problem: simultaneously testing a large set of hypotheses is likely to produce many false results that mistakenly appear significant. Ioannidis argued that "most published research findings are false"[199] due to essentially the same effect: when many scientific teams and researchers each perform many experiments (i.e. process a big amount of scientific data; although not with big data technology), the likelihood of a "significant" result being false grows fast – even more so, when only positive results are published. Furthermore, big data analytics results are only as good as the model on which they are predicated. In an example, big data took part in attempting to predict the results of the 2016 U.S. Presidential Election[200] with varying degrees of success.

See also [edit]

For a list of companies, and tools, see also: Category:Big data.

Information technology portal

- Big data ethics
- Big Data Maturity Model
- Big memory
- Data analysis
- Data curation
- Data defined storage
- Data journalism
- Data lineage
- Data philanthropy
- Data quality
- Data science
- Datafication
- Data warehouse
- In-memory processing
- List of big data companies
- Small data
- Statistics
- Surveillance capitalism
- Urban informatics

References [edit]

1. ^ "The World's Technological Capacity to Store, Communicate, and Compute Information". *MartinHilbert.net*. Retrieved 13 April 2016.
2. ^ Breur, Tom (July 2016). "Statistical Power Analysis and the contemporary "crisis" in social sciences". *Journal of Marketing Analytics*. **4** (2–3): 61–65. doi:10.1057/s41270-016-0001-3. ISSN 2050-3318.
3. ^ Laney, Doug (2001). "3D data management: Controlling data volume, velocity and variety". *META Group Research Note*. **6** (70).
4. ^ Goes, Paulo B. (2014). "Design science research in top information systems journals". *MIS Quarterly: Management Information Systems*. **38** (1): –.
5. ^ Marr, Bernard (6 March 2014). "Big Data: The 5 Vs Everyone Must Know".
6. ^ boyd, dana; Crawford, Kate (21 September 2011). "Six Provocations for Big Data". *Social Science Research Network: A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society*. doi:10.2139/ssrn.1926431

97. ^ "China: Big Data Fuels Crackdown in Minority Region: Predictive Policing Program Flags Individuals for Investigations, Detentions". *hrw.org*. Human Rights Watch. February 26, 2018. Retrieved August 4, 2018.
98. ^ "News: Live Mint". *Are Indian companies making enough sense of Big Data?*. Live Mint. 23 June 2014. Retrieved 22 November 2014.
99. ^ "Startup nation turns big data nation — a look at Israel's top big data startups". *analyticsindiamag.com*. Retrieved 2018-02-28.
100. ^ "Israeli startup uses big data, minimal hardware to treat diabetes". Retrieved 2018-02-28.
101. ^ "Survey on Big Data Using Data Mining" (PDF). International Journal of Engineering Development and Research. 2015. Retrieved 14 September 2016.
102. ^ "Recent advances delivered by Mobile Cloud Computing and Internet of Things for Big Data applications: a survey". International Journal of Network

See also [edit]

For a list of companies, and tools, see also: Category:Big data.

- Big Data Maturity Model
- Big memory
- Data defined storage
- Data journalism
- Data science
- Datafication
- List of big data companies
- Small data

Information technology portal

```
▶ <p>…</p>
▼ <h2>
    <span class="mw-headline" id="See_also">See also</span>
  ▶ <span class="mw-editsection">…</span>
  </h2>
▶ <div role="navigation" aria-label="Portals" class="noprint portal plainlist tright" style="margin:0.5em 0 0
▶ <div role="note" class="hatnote navigation-not-searchable selfref">…</div>
▼ <div class="div-col columns column-width" style="-moz-column-width: 15em; -webkit-column-width: 15em; colun
  ▼ <ul>
    ▼ <li>
        <a href="/wiki/Big_Data_Maturity_Model" title="Big Data Maturity Model">Big Data Maturity Model</a>
      </li>
    ▼ <li>
        <a href="/wiki/Big_memory" title="Big memory">Big memory</a>
      </li>
    ▶ <li>…</li>
    ▶ <li>…</li>
    ▶ <li>…</li>
    ▶ <li>…</li>
    ▶ <li>…</li>
    ▶ <li>…</li>
    ▶ <li>…</li>
```

h2
- span class=“mw-headline” id=“See_also”
- …

…
div class=“div-col …”
- ul
  - li
    - a title=“Big Data Maturity Model”

localhost

jupyter 2 - Wikipedia page data extraction (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Markdown

Wikipedia page data extraction

In this tutorial, we will learn how to extract a static page and convert it into useful information.

We first get a wikipeidia page using requests.

In [ ]:
```
import requests
import re
from bs4 import BeautifulSoup
```

In [ ]:
```
bigdata = requests.get('https://en.wikipedia.org/wiki/Big_data')
```

In [ ]:
```
len(bigdata.text)
```

In [ ]:
```
bigdata.text
```

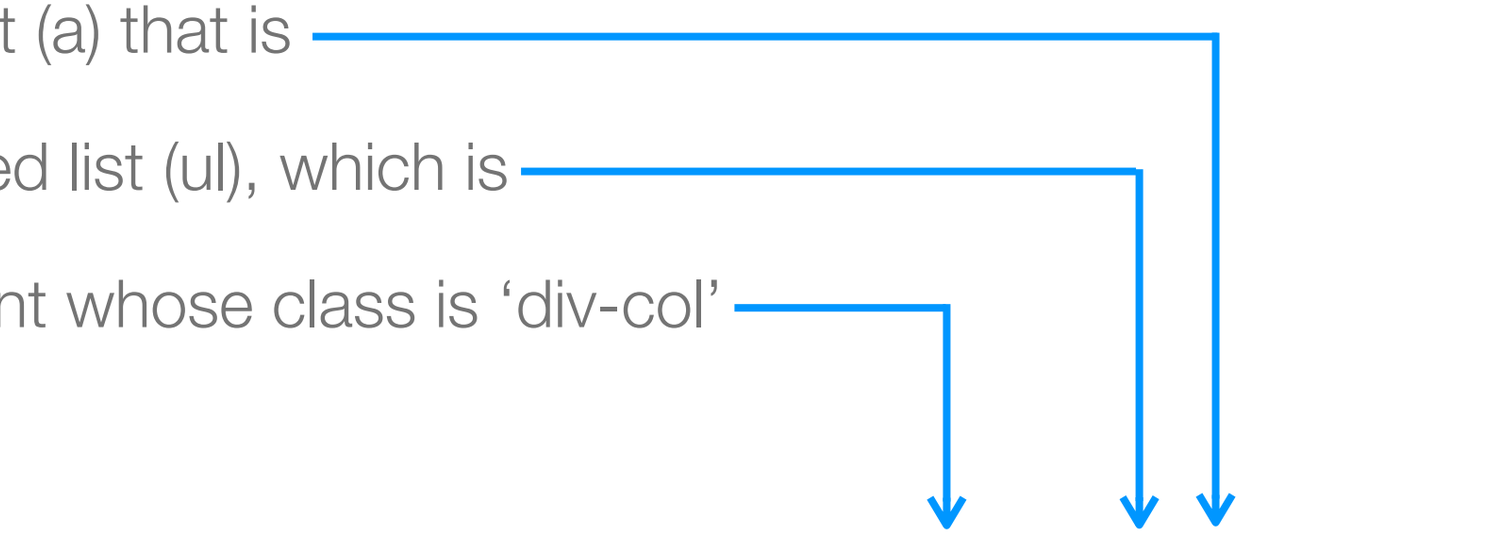
Parsing a wikipedia page

In [ ]:
```
soup = BeautifulSoup(bigdata.text, "lxml")
print(soup.prettify())
```

In [ ]:
```
soup.title.string
```

Understand CSS Selector Syntax

- CSS Selector is a powerful way to navigate DOM tree and locate DOM node(s)
- Find node(s) by tag, id, class, attributes, states (active, focus, hover, etc.), and much more
- Support hierarchical structure e.g. descendant, sibling, order of children, etc.
- In the example, we want to select
  - an anchor element (a) that is
  - inside an unordered list (ul), which is
  - inside a div element whose class is ‘div-col’



```
a_list = soup.select('div.div-col ul a')
```

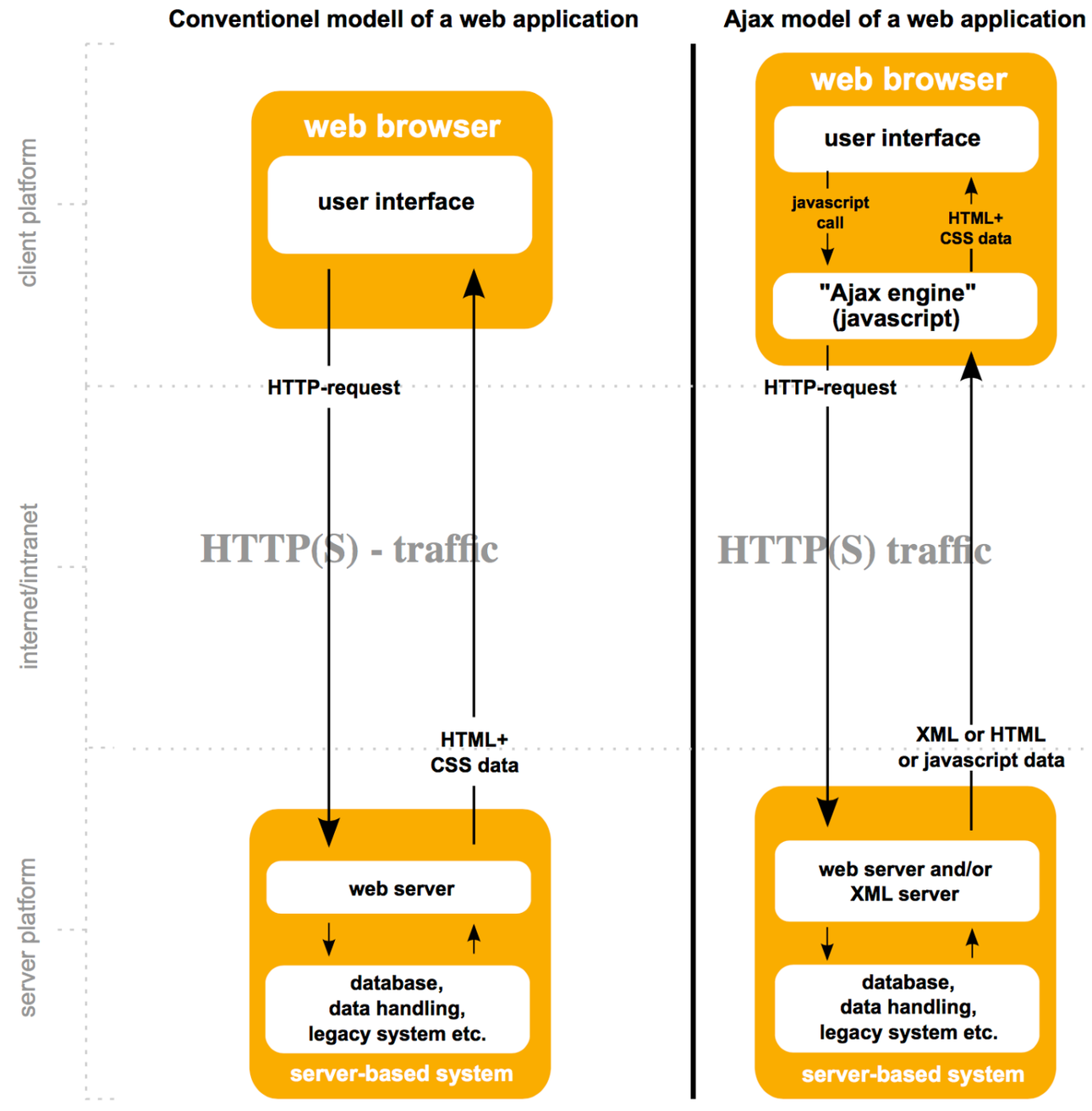
Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
.class1.class2	.name1.name2	Selects all elements with both *name1* and *name2* set within its class attribute
.class1 .class2	.name1 .name2	Selects all elements with *name2* that is a descendant of an element with *name1*
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element.class	p.intro	Selects all <p> elements with class="intro"
element,element	div, p	Selects all elements and all <p> elements
element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element
element+element	div + p	Selects the first <p> element that is placed immediately after elements
element1~element2	p ~ ul	Selects every element that is preceded by a <p> element
[*attribute*]	[target]	Selects all elements with a target attribute
[*attribute=value*]	[target=_blank]	Selects all elements with target="_blank"
[*attribute~=value*]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[*attribute|=value*]	[lang|=en]	Selects all elements with a lang attribute value equal to "en" or starting with "en-"
[*attribute^=value*]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"

https://www.w3schools.com/cssref/css_selectors.asp

CHULA ENGINEERING
Foundation toward Innovation

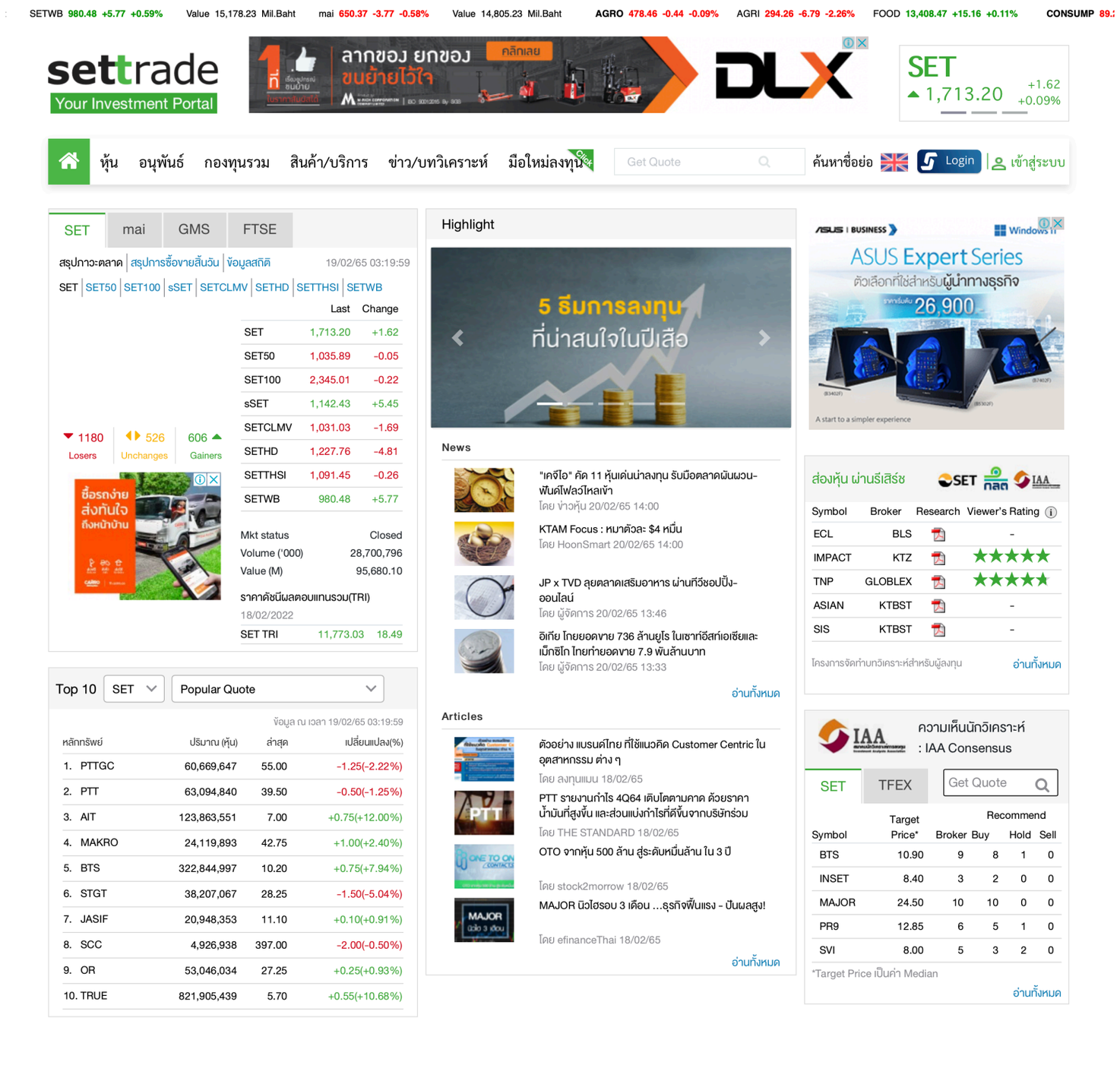
Extracting AJAX-based Webpage

- Modern web applications are AJAX-based e.g. SPA
- Typical web request approach does not work



Source: https://en.wikipedia.org/wiki/Ajax_(programming)

https://www.settrade.com/settrade/home



CHULA ENGINEERING
Foundation toward Innovation

Filter Full URL | All | Document | CSS | Image | Font | JS | XHR/Fetch | Other | Group Media Requests

Name	Domain	Type	Transfer Size	Time	10.00s	20.00s
home	www.settrade.com	document	—	—		
publishertag.prebid.113.js	static.criteo.net	js	—	—		
title-iaa.png	www.settrade.com	png	—	—		
hqdefault.jpg	i.ytimg.com	jpg	—	—		
myanpix.jpg	www.settrade.com	jpg	—	—		
sa_vivobook_pro.jpg	s.isanook.com	jpg	—	—		
amstock.js	www.amcharts.com	js	—	—		
analytics.js	www.google-analytics.com	js	—	—		
title-oppday.png	www.settrade.com	png	—	—		
676504719841639	connect.facebook.net	js	—	—		
amcharts.js	www.amcharts.com	js	—	—		
lsx.jpg	www.settrade.com	jpg	—	—		
142x157_Click2Win.jpg	www.settrade.com	jpg	—	—		
hqdefault.jpg	i.ytimg.com	jpg	—	—		
prebid_2022_2_18_2_14_44.js	anymind360.com	js	—	—		
flagCNY.png	www.settrade.com	png	—	—		
13_sa.jpg	s.isanook.com	jpg	—	—		
tr	www.facebook.com	gif	—	—		
vni.jpg	www.settrade.com	jpg	—	—		
KEX.png	www.set.or.th	png	—	—		
get_dynamic_configuration	api.livechatinc.com	js	—	—		
highlightImage.gif	www.settrade.com	gif	—	—		
topix.jpg	www.settrade.com	jpg	—	—		
x70ser.jpg	s.isanook.com	jpg	—	—		
361539448117743	connect.facebook.net	js	—	—		
39x39_nivate.jpg	www.settrade.com	jpg	—	—		
ads	securepubads.g.doubleclick.net	txt	11.06 KB	1.38s		
polyfills-es2015.9c90355084262abbe1...	www.settrade.com	js	—	—		
flagGBP.png	www.settrade.com	png	—	—		
flagCAD.png	www.settrade.com	png	—	—		
flagNZD.png	www.settrade.com	png	—	—		
highlightImage.gif	www.settrade.com	gif	—	—		
facebook-module-pw.js	www.settrade.com	js	—	—		
1-es2015.50291e34be021e37541a.js	www.settrade.com	js	—	—		
flagINR.png	www.settrade.com	png	—	—		
idx.jpg	www.settrade.com	jpg	—	—		
highlightImage.gif	www.settrade.com	gif	—	—		
get_configuration	api.livechatinc.com	js	—	—		
random_22.jpg	www.settrade.com	jpg	—	—		
highlightImage.gif	www.settrade.com	gif	—	—		
flagUSD.png	www.settrade.com	png	—	—		
light.js	www.amcharts.com	js	—	—		
serial.js	www.amcharts.com	js	—	—		
d0004757.js	lvs.truehits.in.th	js	—	—		
pdpa-script.js	www.settrade.com	js	—	—		
39x39_cfp.jpg	www.settrade.com	jpg	—	—		

CHULA ENGINEERING
Foundation toward Innovation

Filter Full URL | All | Document | CSS | Image | Font | JS | XHR/Fetch | Other | Group Media Requests

Name	Domain	Type	Transfer Size	Time
ads	securepubads.g.doubleclick.net	xhr	—	—
view	securepubads.g.doubleclick.net	fetch	177 B	127ms
activeview	pagead2.googlesyndication.com	fetch	—	1.39ms
view	securepubads.g.doubleclick.net	fetch	177 B	46.3ms
activeview	pagead2.googlesyndication.com	fetch	108 B	168ms
info	api.settrade.com	xhr	1.49 KB	153ms
json	gum.criteo.com	xhr	727 B	178ms
IntradayIndexChartDataServlet	www.settrade.com	xhr	14.22 KB	81.5ms
sodar	pagead2.googlesyndication.com	xhr	10.34 KB	201ms
settrade-mlprediction	xw0vq7mhbi.execute-api.ap-so...	xhr	411 B	426ms
json	gum.criteo.com	fetch	498 B	80.6ms
v1	test.collector.set.or.th	xhr	—	1.3min
v1	collector.set.or.th	xhr	282 B	51.3ms
info	api.settrade.com	xhr	748 B	52.6ms
info	api.settrade.com	xhr	1.21 KB	23.5ms
info	api.settrade.com	xhr	1.21 KB	34.3ms
info	api.settrade.com	xhr	746 B	54.7ms
info	api.settrade.com	xhr	1.22 KB	58.5ms
info	api.settrade.com	xhr	1.22 KB	99.2ms
info	api.settrade.com	xhr	754 B	40.5ms
info	api.settrade.com	xhr	1.21 KB	87.8ms
info	api.settrade.com	xhr	1.22 KB	204ms
info	api.settrade.com	xhr	766 B	49.7ms
info	api.settrade.com	xhr	1.21 KB	283ms
info	api.settrade.com	xhr	1.22 KB	56.4ms

CHULA ENGINEERING
Foundation toward Innovation

Filter Full URL | All | Document | CSS | Image | Font | JS | XHR/Fetch | Other | Group Media Requests

Name

- ads
- view
- activeview
- view
- activeview
- info
- json
- IntradayIndexChartDataServlet
- sodar
- settrade-mlprediction
- json
- v1
- v1
- info
- info
- info
- info
- info
- info
- info
- info
- info
- info
- info
- info
- info

Preview | Headers | Cookies | Sizes | Timing | Security


```
{
    "market_name": "SET",
    "market_display_name": "SET",
    "market_status": "Closed",
    "datetime": "19/02/2022 03:19:59",
    "gainer_amount": 606,
    "gainer_volume": 9794504600,
    "unchange_amount": 526,
    "unchange_volume": 3676836059,
    "loser_amount": 1180,
    "loser_volume": 13345028400,
    "index": [
        {
            "index_name": "SET",
            "index_display_name": "SET",
            "market": "SET",
            "prior": 1711.58,
            "last": 1713.2,
            "change": 1.62,
            "percent_change": 0.0946,
            "high": 1716.14,
            "low": 1703.6,
            "total_volume": 28700796470,
            "total_value": 95680097808.21,
            "flag_url": null
        },
        {
            "index_name": "SET50",
            "index_display_name": "SET50",
            "market": "SET",
            "prior": 1035.94,
            "last": 1035.89,
            "change": -0.05,
            "percent_change": -0.0048,
            "high": 1039.55,
            "low": 1029.61,
            "total_volume": 3001043985,
            "total_value": 58374428219.67,
            "flag_url": null
        },
        {
            "index_name": "SET100",
            "index_display_name": "SET100",
            "market": "SET",
            "prior": 2345.23,
            "last": 2345.01,
            "change": -0.22,
            "percent_change": -0.0093,
            "high": 2352.55,
            "low": 2331.79,
            "total_volume": 3986212374,
            "total_value": 68380195413.62,
            "flag_url": null
        },
```


CHULA ENGINEERING
Foundation toward Innovation

Filter Full URL | All | Document | CSS | Image | Font | JS | XHR/Fetch | Other | Group Media Requests

Name

- ads
- view
- activeview
- view
- activeview
- info
- json
- IntradayIndexChartDataServlet
- sodar
- settrade-mlprediction
- json
- v1
- v1
- info
- info
- info
- info
- info
- info
- info
- info
- info
- info
- activeview
- info
- info
- info
- info
- info
- info

Preview | Headers | Cookies | Sizes | Timing | Security

Summary

URL: https://api.settrade.com/api/market/SET/info
Status: 200
Source: Network
Address: 45.60.48.141:443
Initiator: polyfills-es2015.9c90355084262abbe166.js:1:35749

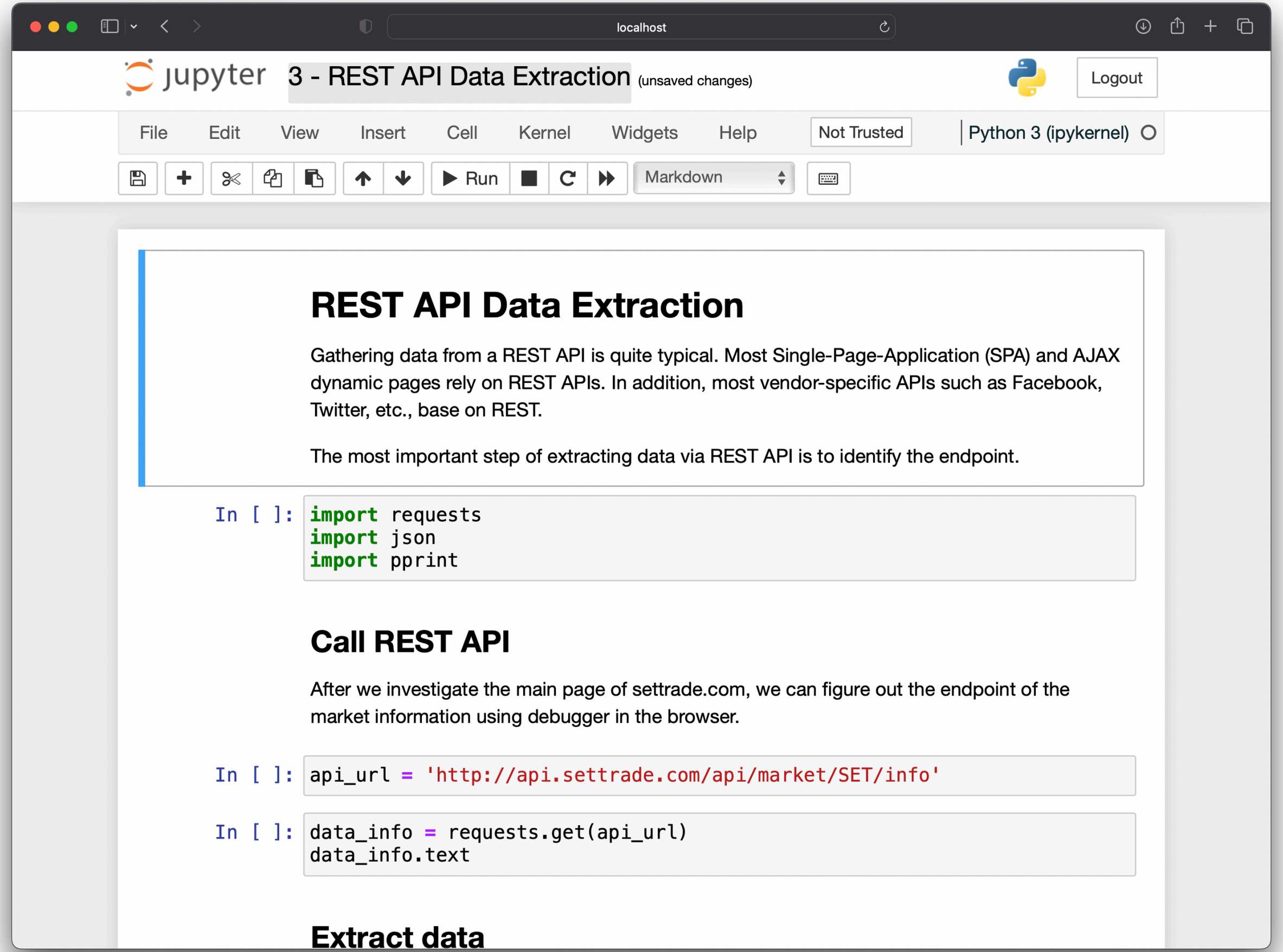
Request

:method: GET
:scheme: https
:authority: api.settrade.com
:path: /api/market/SET/info
Accept: application/json, text/plain, */*
Origin: https://www.settrade.com
Accept-Encoding: gzip, deflate, br
Host: api.settrade.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.3 Safari/605.1.15
Accept-Language: en-CA,en-US;q=0.9,en;q=0.8
Referer: https://www.settrade.com/
Connection: keep-alive

Response

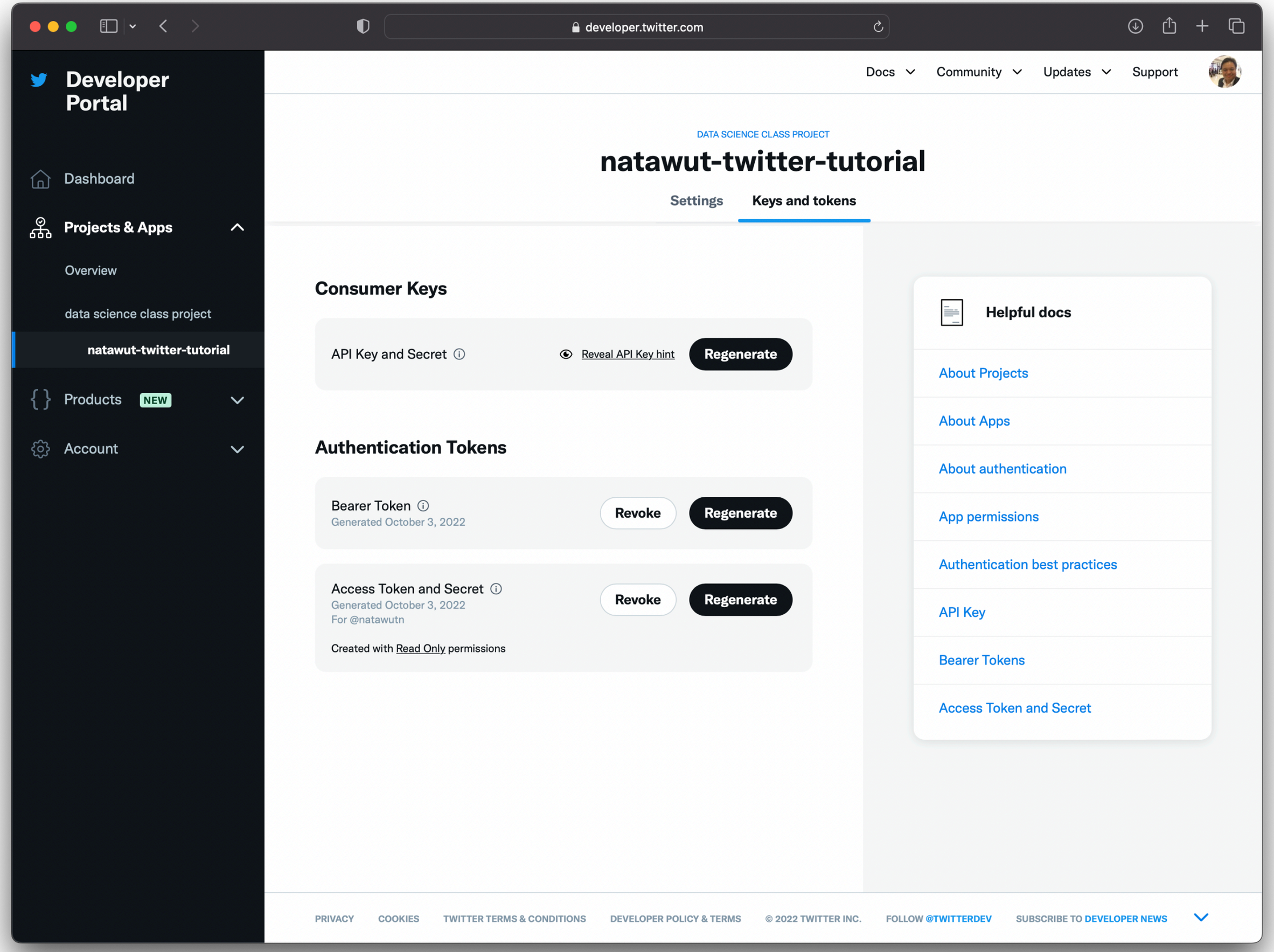
:status: 200
Content-Type: application/json;charset=UTF-8
Access-Control-Allow-Origin: https://www.settrade.com
Set-Cookie: __setuid=CvoYFWIR8KPCRDUMHru2Ag==; domain=.settrade.com; path=/
Set-Cookie: visid_incap_1923491=bFXEbHESR9Kd60Wyql/qj6LwEWIAAAAAQUIPAAAAABbJ8XqwY+XTz44tMej8Cuw; expires=Sun, 19 Feb 2023 17:20:53 GMT; HttpOnly; path=/; Domain=.settrade.com
Set-Cookie: nlbi_1923491_1898525=2GscUDcDHk6XfAVOiwqWKQAAAABl4nnMy2nM2yGeQEsqkUhx; path=/; Domain=.settrade.com
Set-Cookie: incap_ses_1526_1923491=l/XiXInjEh7yvhcy/3AtFaLwEWIAAAAAk+fVQe6ptLO3269Mc53mlg==; path=/; Domain=.settrade.com
Access-Control-Allow-Methods: GET, POST, OPTIONS, DELETE
Content-Encoding: gzip
Access-Control-Allow-Headers: access-control-allow-headers,access-control-allow-origin,x-api-token,x-api-userref,Content-Type,x-api-version,x-device-token,x-device-platform,x-api-system,authorization
Date: Sun, 20 Feb 2022 07:41:23 GMT
Vary: Accept-Encoding
x-cache-status: HIT
x-application-context: settrade-api:production
Server: nginx
x-cdn: Imperva
x-iinfo: 16-55803786-55803787 PNNN RT(1645342882866 0) q(0 0 0 -1) r(0 0) U5

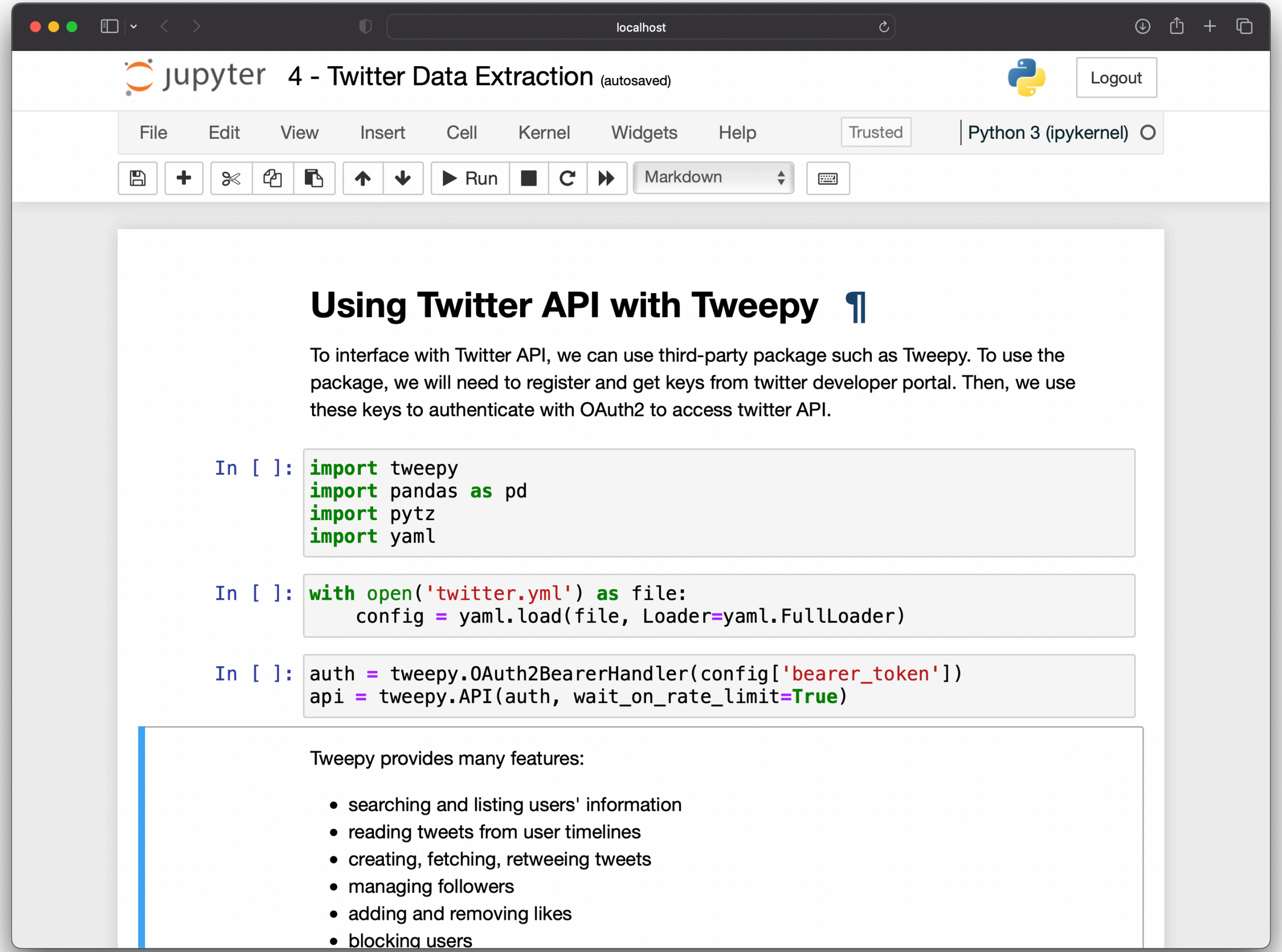
```
(base) ➜  ~ curl https://api.settrade.com/api/market/SET/info | python -m json.tool
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2208    0  2208    0     0    345      0 --:--:--  0:00:06 --:--:--   507
{
    "market_name": "SET",
    "market_display_name": "SET",
    "market_status": "Closed",
    "datetime": "19/02/2022 03:19:59",
    "gainer_amount": 606,
    "gainer_volume": 9794504600.0,
    "unchange_amount": 526,
    "unchange_volume": 3676836059.0,
    "loser_amount": 1180,
    "loser_volume": 13345028400.0,
    "index": [
        {
            "index_name": "SET",
            "index_display_name": "SET",
            "market": "SET",
            "prior": 1711.58,
            "last": 1713.2,
            "change": 1.62,
            "percent_change": 0.0946,
            "high": 1716.14,
            "low": 1703.6,
            "total_volume": 28700796470.0,
            "total_value": 95680097808.21,
            "flag_url": null
        },
        {
            "index_name": "SET50",
            "index_display_name": "SET50",
            "market": "SET",
            "prior": 1035.94,
            "last": 1035.89,
            "change": -0.05,
```


localhost
jupyter
3 - REST API Data Extraction
(unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
REST API Data Extraction
Gathering data from a REST API is quite typical. Most Single-Page-Application (SPA) and AJAX dynamic pages rely on REST APIs. In addition, most vendor-specific APIs such as Facebook, Twitter, etc., base on REST.
The most important step of extracting data via REST API is to identify the endpoint.
In [ ]:
import requests
import json
import pprint
Call REST API
After we investigate the main page of settrade.com, we can figure out the endpoint of the market information using debugger in the browser.
In [ ]:
api_url = 'http://api.settrade.com/api/market/SET/info'
In [ ]:
data_info = requests.get(api_url)
data_info.text
Extract data

Twitter Data Extraction

- To get information from social network, we need to use specific API
  - Facebook Graph API
  - Instagram API
  - Twitter API
- Twitter API
  - REST API with HTTP request and JSON response
  - Request token for access authorization
  - Use third-party package to simplify API requests


developer.twitter.com
Developer Portal
Dashboard
Projects & Apps
Overview
data science class project
natawut-twitter-tutorial
Products NEW
Account
Docs
Community
Updates
Support
DATA SCIENCE CLASS PROJECT
natawut-twitter-tutorial
Settings
Keys and tokens
Consumer Keys
API Key and Secret
Reveal API Key hint
Regenerate
Authentication Tokens
Bearer Token
Generated October 3, 2022
Revoke
Regenerate
Access Token and Secret
Generated October 3, 2022
For @natawutn
Revoke
Regenerate
Created with Read Only permissions
Helpful docs
About Projects
About Apps
About authentication
App permissions
Authentication best practices
API Key
Bearer Tokens
Access Token and Secret
PRIVACY
COOKIES
TWITTER TERMS & CONDITIONS
DEVELOPER POLICY & TERMS
© 2022 TWITTER INC.
FOLLOW @TWITTERDEV
SUBSCRIBE TO DEVELOPER NEWS


localhost
jupyter 4 - Twitter Data Extraction (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown
Using Twitter API with Tweepy ¶
To interface with Twitter API, we can use third-party package such as Tweepy. To use the package, we will need to register and get keys from twitter developer portal. Then, we use these keys to authenticate with OAuth2 to access twitter API.
In [ ]:
import tweepy
import pandas as pd
import pytz
import yaml
In [ ]:
with open('twitter.yml') as file:
config = yaml.load(file, Loader=yaml.FullLoader)
In [ ]:
auth = tweepy.OAuth2BearerHandler(config['bearer_token'])
api = tweepy.API(auth, wait_on_rate_limit=True)
Tweepy provides many features:
searching and listing users' information
reading tweets from user timelines
creating, fetching, retweeing tweets
managing followers
adding and removing likes
blocking users

Extract Complicated Webpage with Selenium

- For complicated webpage, we can use Selenium to drive a web browser to access that webpage to extract some data
  - Navigating
  - Locating Elements using CSS and Path
  - Access info and attributes of page objects
- Checkout https://www.selenium.dev/documentation/ for more information

Data Extraction with Selenium

In this tutorial, we discuss how to use Selenium to extract data from the web. Please see https://selenium-python.readthedocs.io for more details.

Installation

Before using selenium, we will have to install a webdriver of your choice. It can be Chrome or Firefox. Once installed, you will need to know the location of the drive as it will be used as a parameter to start a browser. To install the driver, just install python helper package chromedriver_autoinstaller.

```
pip install chromedriver_autoinstaller
```

We also have to install selenium package.

```
pip install selenium
```

In [ ]:
```
from selenium import webdriver
import chromedriver_autoinstaller
import time
import os
```

In [ ]:
```
chromedriver_autoinstaller.install()
```

Recommended Resources

- http://python.gotrained.com/python-json-api-tutorial/
- https://www.analyticsvidhya.com/blog/2015/10/beginner-guide-web-scraping-beautiful-soup-python/
- https://www.dataquest.io/blog/web-scraping-tutorial-python/
- https://www.datacamp.com/courses/importing-data-in-python-part-2
- https://code.tutsplus.com/tutorials/the-30-css-selectors-you-must-memorize--net-16048

Assignment

- Extract information about “วันพระ” ในระยะเวลา 3 ปีจาก
  - https://www.myhora.com/ปฏิทิน/วันพระ-พ.ศ.2565.aspx
  - https://www.myhora.com/ปฏิทิน/วันพระ-พ.ศ.2566.aspx
  - https://www.myhora.com/ปฏิทิน/วันพระ-พ.ศ.2567.aspx
- Count the distribution of number of week days that are “วันพระ” for all three years and answer the following questions
  - How many วันพระ in total (of 3 years)?
  - How many days in total (of 3 years) that วันพระ is Monday?
  - Which day of the week that has the minimum number of วันพระ? Which has the maximum?
- Note that you can use dateparse package to parse Thai date. There is an example shown in the assignment template