Simple Web Scraping Introduction

by *Cheenar Gupte*

This is my first attempt at creating some kind of LinkedIn content. I've been seeing some of these Jupyter Notebook style posts and thought the format was perfect for something like this. Please let me know if you found any of this helpful and/or have suggestions.

Web Scraping?

If you're reading this you probably have some idea of what web scraping is or what you can do with web scraping, but, if not, web scraping is automating the process of visiting a website with a bot and extracting information.

A lot of companies provide means of accessing data of interest through APIs (application programming interfaces) which essentially is like being able to phone up a company and access some subset of the database without being given explicit access to the database. Think of it as a middle-man that, if possible, you're trying to ask the least amount of things to get the most amount of data.

Web scraping allows statisticians to ask new questions and give better answers.

Libraries

Requests

Let's get to the code. The first thing is we need to import some libraries that will make our lives a lot easier. The first, *requests*, will allow us to make web requests like GET, POST, PUT, etc.

Web requests are the foundation for how the web works. The higher level abstraction is simply when a web browser tries to navigate to a website, it sends a web request to the web server and the server responds with some HTML,CSS,JS,etc that the web browser can then interpret and display.

The requests library is a powerful tool but we will focus on using just GET requests.

If this cell returns an error, make sure to do `pip install requests`

In [2]:
```
import requests # imports the request module
```

BeautifulSoup

Another really powerful tool for web scraping. When we make a web request, the web server will send us back raw HTML and CSS. BeautifulSoup handles all of the parsing and gives us an object hierarchy to descend through. We will focus on using a few vital ones for simple web scraping.

If this cell returns an error, make sure to do `pip install bs4`

In [3]:
```
from bs4 import BeautifulSoup # BeautifulSoup is defined in the bs4 modu
le, this is a way of
#directly accessing it without having to do bs4.BeautifulSoup
```

Website of Interest

For this simple scraping tutorial, I've decided to use the http://dailycal.org (http://dailycal.org) to scrape news clips. The first page we will be working on is http://www.dailycal.org/section/news/ (http://www.dailycal.org/section/news/)

Building Our Toolbox: Getting the Raw Data Ready

First thing we want to be able to do with this web page is detect the links on it, but we need to get the webpage's data in an interactable setting. To do this, we will use the modules we imported early on to navigate to the webpage and parse the data.

In [6]:
```
web_url = "http://www.dailycal.org/section/news/" # this is the web page
that we will be navigating to

raw_page_data = requests.get(web_url) # creates a GET request to the web
page, will return the HTML
print("Request Status Code: " + str(raw_page_data.status_code)) # HTTP S
tatus Codes tell you if the server accepted/denied the request, 200 is a
successful request
```

```
Request Status Code: 200
```

Now we will parse the code using BeautifulSoup. To do this, we will need an XML parser installed -- luckily your python configuration probably already bundles one. In my experience, configuring the BeautifulSoup object without any explicit parser should work in most instances because the system automatically detects one, however, if an error occurs, you can do: `pip install lxml`

In [7]:
```
soup = BeautifulSoup(raw_page_data.content) # soupifies the Object, auto
matically finds XML parser
#soup = BeautifulSoup(raw_page_data.content, "lxml") # Explcitly states
 the XML parser
```

Soupified Object

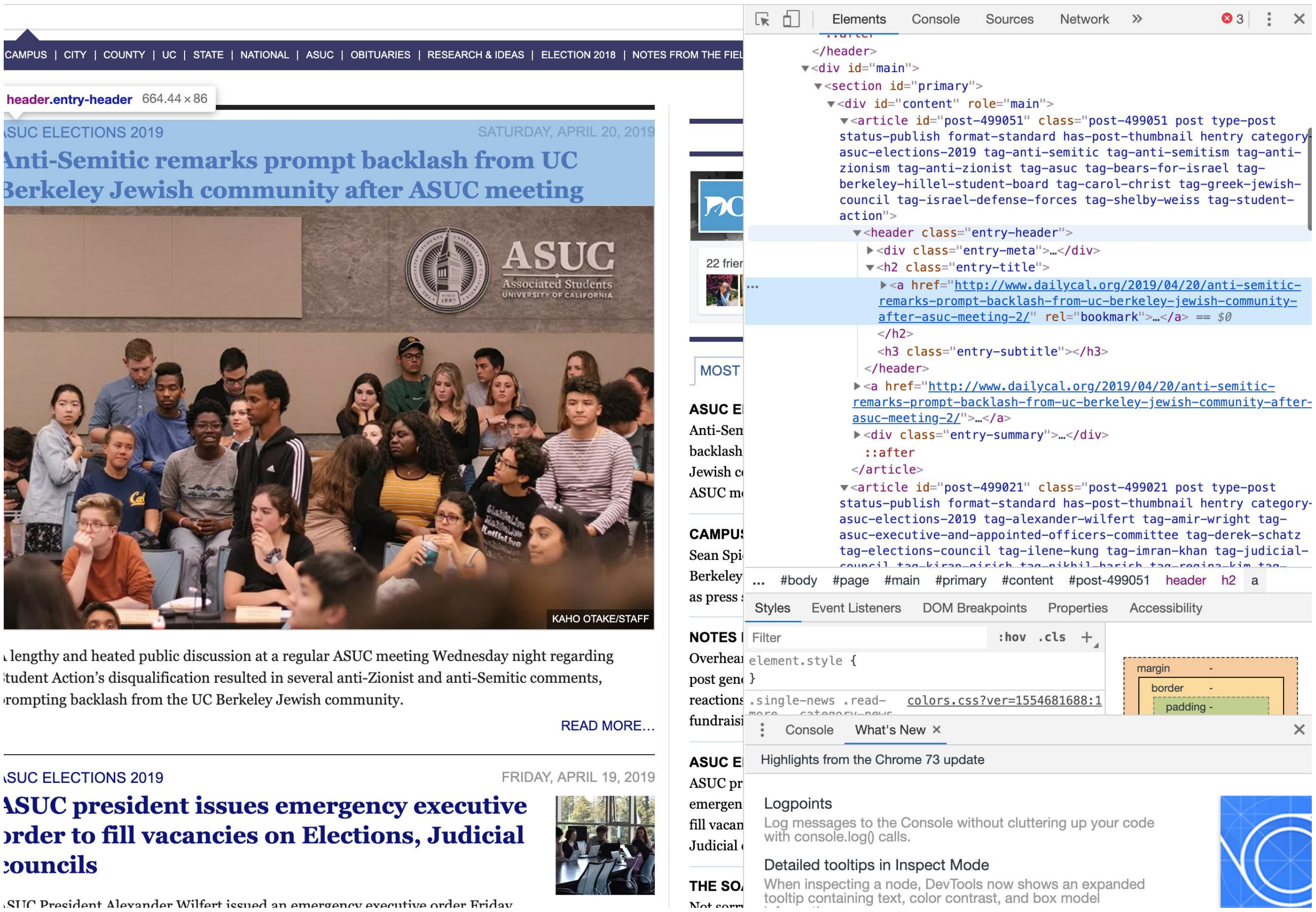
Now we have put our `raw_page_data` into a soup object. This object can be interfaced with to access different parts of the DOM (Document-Object-Model) and extract relevant data. We can explore some of the useful functions now

In [12]:
```
# find() -> finds first instance of an HTML tag
# find("tag", {"attrs":"value"})
# <tag class="title"> ==> find("tag", {"class":"title"})
section_title = soup.find("h2", {"class":"section-title"})

# find_all() -> finds all the instances of a tag, recusively
# find_all("tag", {"attrs":"value"})
relevant_news_articles = soup.find_all("a", {"rel":"bookmark"})

print(section_title.text)
print([x.text for x in relevant_news_articles])
```

```
News
['Anti-Semitic remarks prompt backlash from UC Berkeley Jewish communit
y after ASUC meeting', 'ASUC president issues emergency executive order
to fill vacancies on Elections, Judicial councils', '9 ASUC officials r
esign, students protest in response to Student Action candidates’ disqu
alification from 2019 elections', 'UC workers union calls for speakers
to boycott UC commencements', 'Sean Spicer speaks at UC Berkeley, looks
back on time as press secretary', 'Free menstrual products now availabl
e on Moffitt’s 4th, 5th floors', 'UC Berkeley undergraduates launch sou
ndproof karaoke pod startup', 'Panelists, study expose culture of anxie
ty at UC Berkeley', 'UC Berkeley lecturer sends memo to governor, corre
lating wildfires with rising utility prices', 'Student Action has been
disqualified not 1, but 3 times']
```

With our example, we already have a powerful way to detect links. But how did I know what the tag and attributes of the relevant information were? An essential component of the scraping toolbox is the Developer Console in Chrome or Firefox.

If you right click on the webpage and click `Inspect` you will be able to see the entire page's source code and, conveniently, use the pointer tool at the top left of the console to visually select elements of interest and view their tag specifications.


CAMPUS | CITY | COUNTY | UC | STATE | NATIONAL | ASUC | OBITUARIES | RESEARCH & IDEAS | ELECTION 2018 | NOTES FROM THE FIEL
header.entry-header 664.44 × 86
KAHO OTAKE/STAFF
READ MORE...
Elements
Console
Sources
Network
Styles
Event Listeners
DOM Breakpoints
Properties
Accessibility
What's New
Highlights from the Chrome 73 update

SUC ELECTIONS 2019 SATURDAY, APRIL 20, 2019

Anti-Semitic remarks prompt backlash from UC Berkeley Jewish community after ASUC meeting

lengthy and heated public discussion at a regular ASUC meeting Wednesday night regarding tudent Action's disqualification resulted in several anti-Zionist and anti-Semitic comments, rompting backlash from the UC Berkeley Jewish community.

SUC ELECTIONS 2019 FRIDAY, APRIL 19, 2019

ASUC president issues emergency executive order to fill vacancies on Elections, Judicial councils

SUC President Alexander Wilfert issued an emergency executive order Friday

```
      </header>
    ▼<div id="main">
      ▼<section id="primary">
        ▼<div id="content" role="main">
          ▼<article id="post-499051" class="post-499051 post type-post
          status-publish format-standard has-post-thumbnail hentry category-
          asuc-elections-2019 tag-anti-semitic tag-anti-semitism tag-anti-
          zionism tag-anti-zionist tag-asuc tag-bears-for-israel tag-
          berkeley-hillel-student-board tag-carol-christ tag-greek-jewish-
          council tag-israel-defense-forces tag-shelby-weiss tag-student-
          action">
            ▼<header class="entry-header">
              ▶<div class="entry-meta">…</div>
              ▼<h2 class="entry-title">
...             ▶<a href="http://www.dailycal.org/2019/04/20/anti-semitic-
                remarks-prompt-backlash-from-uc-berkeley-jewish-community-
                after-asuc-meeting-2/" rel="bookmark">…</a> == $0
                </h2>
                <h3 class="entry-subtitle"></h3>
              </header>
            ▶<a href="http://www.dailycal.org/2019/04/20/anti-semitic-
            remarks-prompt-backlash-from-uc-berkeley-jewish-community-after-
            asuc-meeting-2/">…</a>
            ▶<div class="entry-summary">…</div>
              ::after
            </article>
          ▼<article id="post-499021" class="post-499021 post type-post
          status-publish format-standard has-post-thumbnail hentry category-
          asuc-elections-2019 tag-alexander-wilfert tag-amir-wright tag-
          asuc-executive-and-appointed-officers-committee tag-derek-schatz
          tag-elections-council tag-ilene-kung tag-imran-khan tag-judicial-
... #body #page #main #primary #content #post-499051 header h2 a
```

```
element.style {
}
.single-news .read-    colors.css?ver=1554681688:1
```

Logpoints
Log messages to the Console without cluttering up your code with console.log() calls.

Detailed tooltips in Inspect Mode
When inspecting a node, DevTools now shows an expanded tooltip containing text, color contrast, and box model

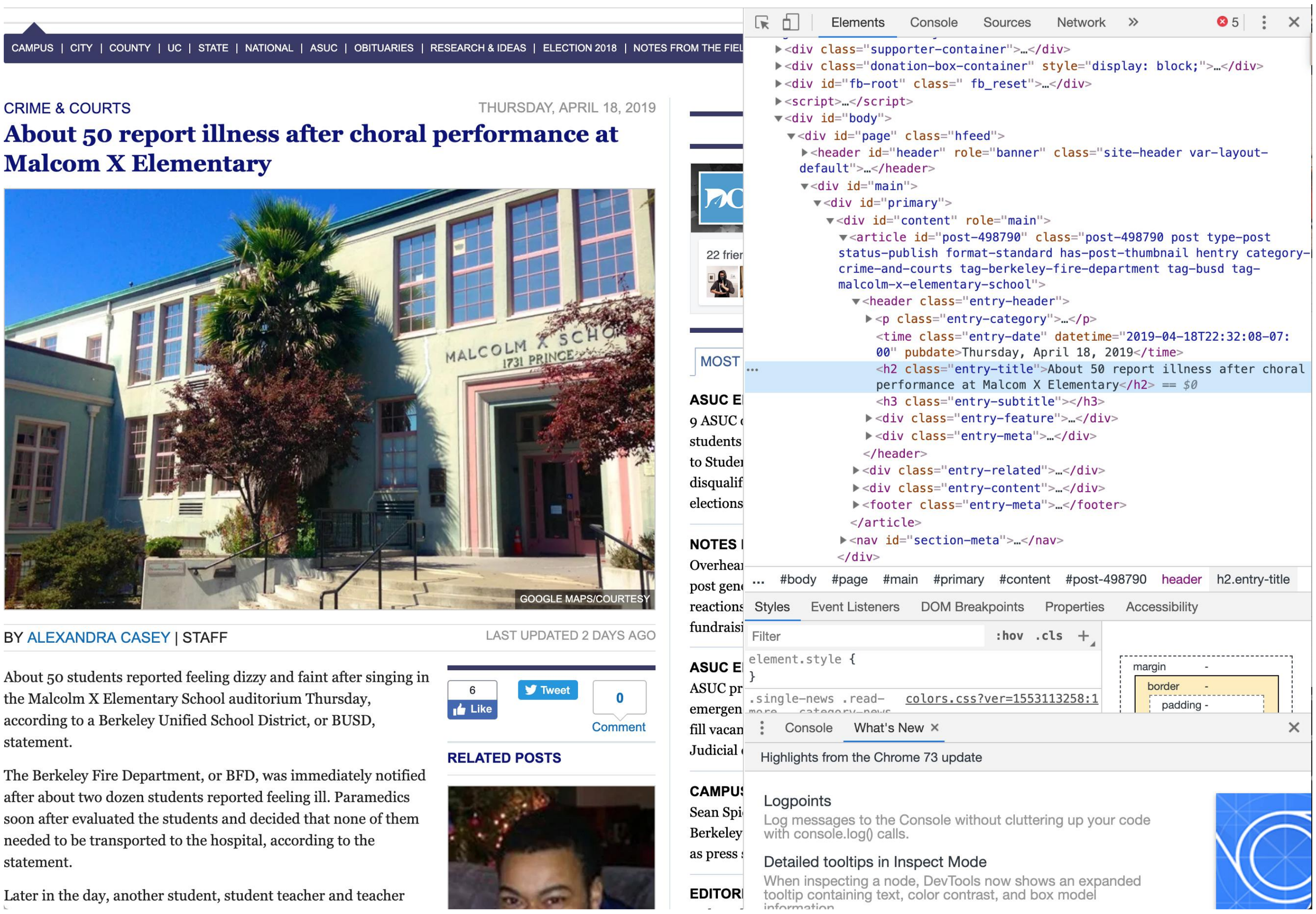
Parsing the Links and Articles

Now we have a list of all the article link objects thanks to our use of the `find_all` function. We want to take this list and for each element in that list, scrape the webpage containing the article information.

At this point, we have already one some exciting web-scraping. We can now create a database of all the links and titles for articles in the DailyCal with essentially the code we have already written!

Article View

Now we are going to the same thing that we did before using the inspect tool, but this time, we are going to click into one of the articles and create a function that will generically scrape the pages. We can then apply this function to all of our link objects.



CRIME & COURTS THURSDAY, APRIL 18, 2019

About 50 report illness after choral performance at Malcom X Elementary

BY ALEXANDRA CASEY | STAFF LAST UPDATED 2 DAYS AGO

About 50 students reported feeling dizzy and faint after singing in the Malcolm X Elementary School auditorium Thursday, according to a Berkeley Unified School District, or BUSD, statement.

The Berkeley Fire Department, or BFD, was immediately notified after about two dozen students reported feeling ill. Paramedics soon after evaluated the students and decided that none of them needed to be transported to the hospital, according to the statement.

Later in the day, another student, student teacher and teacher

RELATED POSTS

```
▶<div class="supporter-container">…</div>
▶<div class="donation-box-container" style="display: block;">…</div>
▶<div id="fb-root" class=" fb_reset">…</div>
▶<script>…</script>
▼<div id="body">
  ▼<div id="page" class="hfeed">
    ▶<header id="header" role="banner" class="site-header var-layout-
    default">…</header>
    ▼<div id="main">
      ▼<div id="primary">
        ▼<div id="content" role="main">
          ▼<article id="post-498790" class="post-498790 post type-post
          status-publish format-standard has-post-thumbnail hentry category-
          crime-and-courts tag-berkeley-fire-department tag-busd tag-
          malcolm-x-elementary-school">
            ▼<header class="entry-header">
              ▶<p class="entry-category">…</p>
                <time class="entry-date" datetime="2019-04-18T22:32:08-07:
                00" pubdate>Thursday, April 18, 2019</time>
...             <h2 class="entry-title">About 50 report illness after choral
                performance at Malcom X Elementary</h2> == $0
                <h3 class="entry-subtitle"></h3>
              ▶<div class="entry-feature">…</div>
              ▶<div class="entry-meta">…</div>
            </header>
          ▶<div class="entry-related">…</div>
          ▶<div class="entry-content">…</div>
          ▶<footer class="entry-meta">…</footer>
          </article>
        ▶<nav id="section-meta">…</nav>
        </div>
```

... #body #page #main #primary #content #post-498790 header h2.entry-title

```
element.style {
}
.single-news .read-     colors.css?ver=1553113258:1
```

Highlights from the Chrome 73 update

Logpoints
Log messages to the Console without cluttering up your code with console.log() calls.

Detailed tooltips in Inspect Mode
When inspecting a node, DevTools now shows an expanded tooltip containing text, color contrast, and box model information.

In [13]:
```python
def scrape_article(article_object):
    url = article_object.attrs["href"] # this allows us to access tag at
tributes like <tag href="url">

    raw_article_source = requests.get(url)
    souped_article = BeautifulSoup(raw_article_source.content)

    title = souped_article.find("h2", {"class":"entry-title"}).text
    date = souped_article.find("time", {"class":"entry-date"}).text
    author = souped_article.find("a", {"rel":"author"}).text
    content = souped_article.find("div", {"class":"entry-content"}).text

    return (title, date, author, content)
```

In [14]:
```
# let's test our function on a sample article
sample_article = relevant_news_articles[0]
sample_scrape = scrape_article(sample_article)

print(sample_scrape)
```

```
('Anti-Semitic remarks prompt backlash from UC Berkeley Jewish communit
y after ASUC meeting', 'Saturday, April 20, 2019', 'Maya Eliahou', '\nA
lengthy and heated public discussion at a regular ASUC meeting Wednesda
y night regarding Student Action’s disqualification resulted in several
anti-Zionist and anti-Semitic comments, prompting backlash from the UC
Berkeley Jewish community.\nIn a joint statement released Thursday even
ing, many Jewish campus groups that signed the statement said they were
“appalled and deeply pained” by the anti-Semitic remarks made at the me
eting — including verbal abuse, harassment and false information, accor
ding to the statement.\n“We, as Jewish students, condemn the events of
April 17th and stand united against the hatred and ignorance that perme
ates our campus,” the statement said. “We ask that the campus community
stand with us, make your voice heard and show that Berkeley truly does
stand united against hate.”\nAttended by more than 200 students, the me
eting was dominated by a public comment period during which students di
scussed the disqualification of Student Action’s entire slate — includi
ng Jewish former senator-elect Shelby Weiss — from the 2019 ASUC electi
ons.\nThroughout the meeting, Jewish students who said they had lost a
source of representation in the ASUC were met with remarks that referen
ced Zionism and the Israel Defense Forces. Some of the comments — notab
ly one from a former ASUC senator who said she was hearing “white tear
s” and “Zionist tears” — were caught on video and shared on Facebook.\n
Many, though not all, campus Jewish student groups signed the statemen
t, including the Berkeley Hillel Student Board, Bears for Israel and th
e Greek Jewish Council. Though some of the participating groups have di
splayed differing political beliefs about Zionism, they all presented a
united front in their disapproval of the events that occurred at the me
eting.\nChancellor Carol Christ sent an email to the campus community F
riday morning addressing “disturbing expressions of bias” at the ASUC m
eeting. The email did not specifically mention the Jewish campus commun
ity or details regarding the incident, but it reaffirmed the campus adm
inistration’s condemnation of bias, including “racism, anti-Semitism an
d other forms of prejudice.”\nContact Maya Eliahou at [email\xa0protect
ed] and follow her on Twitter at @MayaEliahou.\n')
```

Rigging it Together

Now we can convert all of our article objects into parsed data. Since we storing it in a tuple, it should be fairly simple to generate a simple table strucutre to display our data.

In [28]:
```
print("Title,Date,Author,Content(Trunc.)")
print()

for article in relevant_news_articles:
    scraped_data = scrape_article(article)
    print(scraped_data[0])
    print(scraped_data[1])
    print(scraped_data[2])
    print(scraped_data[3][:250]) #truncated the output
    print()
```

```
Title,Date,Author,Content(Trunc.)

Anti-Semitic remarks prompt backlash from UC Berkeley Jewish community
after ASUC meeting
Saturday, April 20, 2019
Maya Eliahou

A lengthy and heated public discussion at a regular ASUC meeting Wednes
day night regarding Student Action’s disqualification resulted in sever
al anti-Zionist and anti-Semitic comments, prompting backlash from the
UC Berkeley Jewish community.
In a j

ASUC president issues emergency executive order to fill vacancies on El
ections, Judicial councils
Friday, April 19, 2019
Amber Tang

ASUC President Alexander Wilfert issued an emergency executive order Fr
iday afternoon in an attempt to quickly fill the vacancies brought abou
t by a flurry of resignations from the Elections and Judicial councils
— an incident that has thrown ASUC o

9 ASUC officials resign, students protest in response to Student Action
candidates’ disqualification from 2019 elections
Thursday, April 18, 2019
Anjali Shrivastava

About 200 students attended Wednesday night’s regular ASUC meeting to b
oth voice their support for and condemnation of the Judicial Council’s
recent controversial ruling that resulted in the retroactive disqualifi
cation of all Student Action candida

UC workers union calls for speakers to boycott UC commencements
Friday, April 19, 2019
Ben Klein

The American Federation of State, County and Municipal Employees, or AF
SCME, Local 3299 has called for speakers to boycott their UC speaking e
ngagements amid ongoing labor disputes over outsourcing and “illegal la
bor practice.”
The call for the boyc

Sean Spicer speaks at UC Berkeley, looks back on time as press secretar
y
Thursday, April 18, 2019
Brandon Yung

When Sean Spicer walked up to the lectern in front of a crowded Evans H
all lecture room, he was met with a round of applause that took him aba
ck: “That was not the introduction I was expecting, but I appreciate i
t.”
The former White House press secr

Free menstrual products now available on Moffitt’s 4th, 5th floors
Thursday, April 18, 2019
Boyce Buchanan
```

```
Free pads and tampons can now be found in Moffitt Library, after the Co
alition for the Institutionalization of Free Menstrual Products, or CIF
MP, worked with the library to secure menstrual products in its fourth-
and fifth-floor bathrooms.
The mens

UC Berkeley undergraduates launch soundproof karaoke pod startup
Thursday, April 18, 2019
Aishwarya Kaimal

UC Berkeley undergraduates Luofei Chen, Noah Adriany and Aayush Tyagi l
aunched Oki Karaoke in September 2018 — a startup that aims to bring As
ia’s soundproof karaoke pods to the United States.
In China last summer, Chen — a freshman in the UC Berkel

Panelists, study expose culture of anxiety at UC Berkeley
Friday, April 19, 2019
Julie Madsen

In an effort to address the mental health climate at UC Berkeley, the B
erkeley Institute for the Future of Young Americans and the Goldman Sch
ool of Public Policy hosted a conversation Thursday titled “A Generatio
n Under Pressure: Talking Mental Hea

UC Berkeley lecturer sends memo to governor, correlating wildfires with
rising utility prices
Thursday, April 18, 2019
Marlena Tavernier-Fine

Steven Weissman, a lecturer at the UC Berkeley Goldman School of Public
Policy, sent a memorandum to Gov. Gavin Newsom on April 10, explaining
the impact of California wildfires on the prices of gas and electric ut
ilities.
In 2018, more than 800,000

Student Action has been disqualified not 1, but 3 times
Wednesday, April 17, 2019
Maya Eliahou

With all 14 of Student Action’s newly elected candidates disqualified a
fter a unanimous ASUC Judicial Council ruling Tuesday, many students ar
e wondering what will come next.
The Judicial Council disqualified all of Student Action’s elected candi
dat
```

Part 2.

With these techniques, anyone with a little Python knowledge can start scraping simple websites and getting the relevant information.

Going on from here, you'd want to start using the `csv` module to start outputing your data so you can clean it for later analysis. This program was capable of scraping only a single page for links and then acquiring the content from it. However, if we wanted to, we could rig a loop to scrape every page from the DailyCal.

I plan on writing another Jupyter Notebook file like this where I rig up a program that is capable of scraping the entire DailyCal and saving it in a format that can be viewed in Excel.

Any questions? cheenar@berkeley.edu