Scraping with Python

ale rimoldi

Scraping with Python

What is *scraping*

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites.

From Wikipedia

cc-by digboston ↗

What is *scraping*

Web browsing:	Going into the internet for information (. . . or entertainment)
Web scraping:	Looking for data Collecting for data

From Data to Information

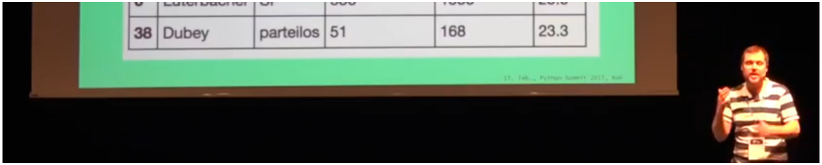
Data is raw, not organized, material which – after being analyzed – can be used to derive information.

Information is for humans.

It's about giving significance and usefulness to data through interpretation.

It has a meaning.

Inspiration: The Swiss Federal Administrative Court PDFs



Barnaby Skinner – Scraping the Federal Administrative Court's Database $\nearrow$, Swiss Python Summit 2017

Starting point:

- The rulings are published as PDFs
- They can be searched through a Web Form
- Does the party affiliation of the judge affect the decisions?

Result:

- Automatically submit the form and get 29'263 verdicts
- "Scrape" the content of the PDFs
- Yes

Inspiration: Data Journalism

Barnaby Skinner ↗ is a Data Journalist -> Data drive story telling

A selection of data driven investigations, feature stories, interviews and other bits and pieces. ↗

- Bundesrichter nicken Urteile oft nur noch ab ↗
  Eine Analyse aller 73'000 Urteile seit 2007 zeigt, wie gross der Einfluss der Gerichtsschreiber auf die Rechtsprechung ist. Hintergrund ist die Beschwerdeflut am obersten Gericht.
- Die Gutheissungsquoten sinken bei allen Richtern ↗
  Der Präsident des Bundesverwaltungsgerichts bezeichnete eine Analyse zu den Asylentscheiden wider besseres Wissen als falsch – zumindest gegen aussen.

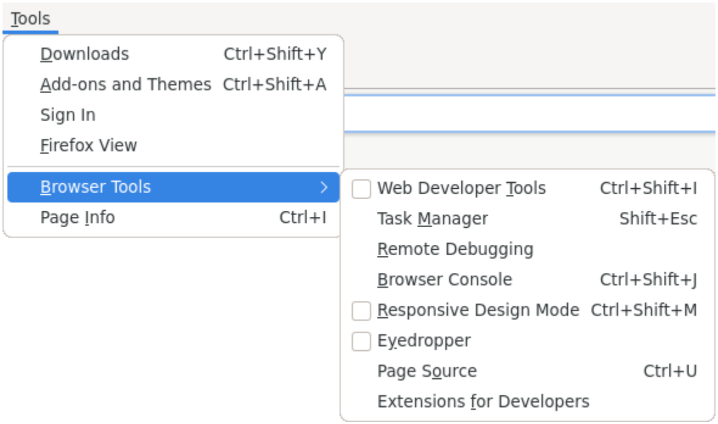
Inspiration: David Kriesel @ ccc

BahnMining - Pünktlichkeit ist eine Zier (David Kriesel) $\nearrow$

Result: you can save money, if you know in advance which train will be delayed by how much.

Behind the Website

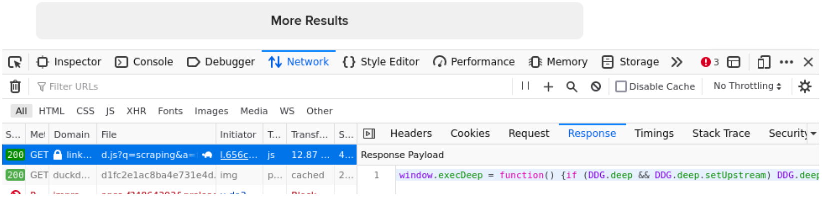
- Page Source (ctrl + u)
- Web Developer tools (F12)
  - Inspect (Q)


Tools
Downloads Ctrl+Shift+Y
Add-ons and Themes Ctrl+Shift+A
Sign In
Firefox View
Browser Tools
Page Info Ctrl+I
Web Developer Tools Ctrl+Shift+I
Task Manager Shift+Esc
Remote Debugging
Browser Console Ctrl+Shift+J
Responsive Design Mode Ctrl+Shift+M
Eyedropper
Page Source Ctrl+U
Extensions for Developers

Example: https://www.pwg.ch/ ↗

Snoop the JSON

- Web Developer tools (F12)
- Network
  - JSON
    - Response



S...	Me	Domain	File	Initiator	T...	Transf...	S...
200	GET	link...	d.js?q=scraping&a=	l.656c...	js	12.87 ...	4...
200	GET	duckd...	d1fc2e1ac8ba4e731e4d.	img	p...	cached	2...

Response Payload

```
window.execDeep = function() {if (DDG.deep && DDG.deep.setUpstream) DDG.deep
```

Example: https://duckduckgo.com ↗

Getting the Website content

Can we *simply* get a web page text with Python?

No

What can we easily get?

The HTML source. And then extract the text from it.

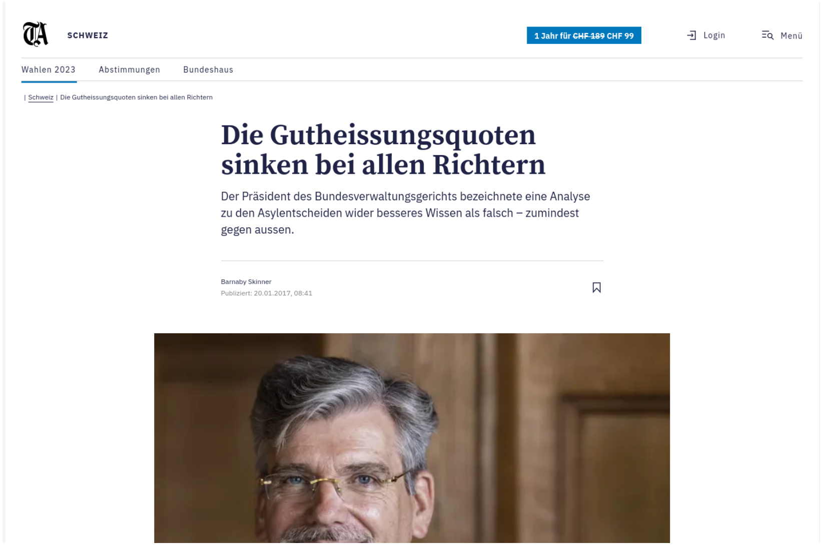
```
$ pip install requests beautifulsoup4
```

```
import requests
from bs4 import BeautifulSoup

r = requests.get('https://www.tagesanzeiger.ch/' + \
    'die-gutheissungsquoten-sinken-bei-allen-' + \
    'richtern-404322581374')
# print(r.text)
soup = BeautifulSoup(r.text)
print(soup.get_text())
```

Getting the Website content

This is what we want . . .



SCHWEIZ

1 Jahr für ~~CHF 189~~ CHF 99

Login

Menü

Wahlen 2023 Abstimmungen Bundeshaus

| Schweiz | Die Gutheissungsquoten sinken bei allen Richtern

Die Gutheissungsquoten sinken bei allen Richtern

Der Präsident des Bundesverwaltungsgerichts bezeichnete eine Analyse zu den Asylentscheiden wider besseres Wissen als falsch – zumindest gegen aussen.

Barnaby Skinner
Publiziert: 20.01.2017, 08:41

Getting the Website content

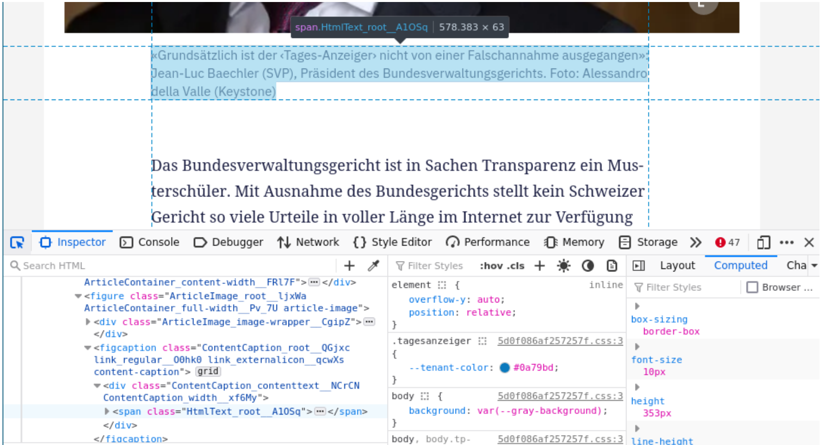
This is what we get . . .

Die Gutheissungsquoten sinken bei allen Richtern | Tages-AnzeigerIhr Browser ist veraltet. Bitte aktualisieren Sie Ihren Browser auf die neueste Version, oder wechseln Sie auf einen anderen Browser wie Chrome, Safari, Firefox oder Edge um Sicherheitslücken zu vermeiden und eine bestmögliche Performance zu gewährleisten.Zum Hauptinhalt springenTages-AnzeigerSchweizLoginMenüWahlen 2023AbstimmungenBundeshaus SchnellzugriffeNeueste ArtikelDas Wichtigste des TagesE-PaperNewsletter & AlertsPodcastsSchaadzeile/WidmerzeileAboshopcarte blanche & AbovorteileThemen & SchwerpunkteStartseiteSchwerpunkteWahlen 2023ZürichStadtRegionGastroZüritipp(s)KurzmeldungenMeine GemeindeSchweiz . . .

https://www.tagesanzeiger.ch/die-gutheissungsquoten-sinken-bei-allen-richtern-404322581374 ↗

Looking for Information in the HTML

- In the browser . . .
- Open the *Web Developer Tools* (F12).
- Use the element picker (ctrl-shift-c).
- Often the HTML tag and the CSS classes help identifying where the interesting information is.



«Grundsätzlich ist der ‹Tages-Anzeiger› nicht von einer Falschannahme ausgegangen»: Jean-Luc Baechler (SVP), Präsident des Bundesverwaltungsgerichts. Foto: Alessandro della Valle (Keystone)

Das Bundesverwaltungsgericht ist in Sachen Transparenz ein Musterschüler. Mit Ausnahme des Bundesgerichts stellt kein Schweizer Gericht so viele Urteile in voller Länge im Internet zur Verfügung

```
      ArticleContainer_content-width__FRl7F">…</div>
    <figure class="ArticleImage_root__ljxWa
      ArticleContainer_full-width__Pv_7U article-image">
      <div class="ArticleImage_image-wrapper__CgipZ">…
      </div>
      <figcaption class="ContentCaption_root__QGjxc
        link_regular__O0hk0 link_externalicon__qcwXs
        content-caption"> grid
        <div class="ContentCaption_contenttext__NCrCN
          ContentCaption_width__xf6My">
          <span class="HtmlText_root__A1OSq">…</span>
        </div>
      </figcaption>
```

```
element {                                inline
    overflow-y: auto;
    position: relative;
}
.tagesanzeiger {        5d0f086af257257f.css:3
    --tenant-color: ● #0a79bd;
}
body {                  5d0f086af257257f.css:3
    background: var(--gray-background);
}
body, body.tp-          5d0f086af257257f.css:3
```

```
box-sizing
  border-box
font-size
  10px
height
  353px
line-height
```

Looking for Information in the HTML

```
import requests
from bs4 import BeautifulSoup

r = requests.get('https://www.tagesanzeiger.ch/' + \
    'die-gutheissungsquoten-sinken-bei-allen-' + \
    'richtern-404322581374')
# print(r.text)
soup = BeautifulSoup(r.text)
print(soup.find('figcaption').find('span'))
```

Gives us:

```
<span class="HtmlText_root__A1OSq">«Grundsätzlich ist
der ‹Tages-Anzeiger› nicht von einer Falschannahme
ausgegangen»: Jean-Luc Baechler (SVP), Präsident des
Bundesverwaltungsgerichts. Foto: Alessandro della
Valle (Keystone)</span>
```

Exploit the structure in the HTML

We are not limited to the tags! We can also search by CSS class:

```
print(soup.find('figcaption').find('span', \
    class_='HtmlText_root__A1OSq'))
```

Nice!

As you can guess, `__A1OSq` is some random value, that is only there to make the life harder for the honest scraper.

You might want to get rid of it ...

Exploit the structure in the HTML

Luckily, BeautifulSoup has an answer for it.

We can use a RegEx!

```
print(soup.find('figcaption').find('span', \
    class_=re.compile('^HtmlText_root.*')))
```

Or even a function!

```
print(soup.find('figcaption').find('span', \
    class_=lambda e: \
        e.startswith('HtmlText_root') \
        if e else False))
```

Getting *complex* pages

With `requests` and `beautifulsoup4` together we can get far. But, sometimes, we need to run Javascript code to get the full content.

For such cases, you can use `requests-html`, which uses Chromium to build the page as if you were browsing the web:

```
$ pip install requests-html
```

```
from requests_html import HTMLSession

session = HTMLSession()
url = 'https://www.nzz.ch/schweiz'
r = session.get(url)
r.html.render(sleep=1, scrolldown=5)
articles = r.html.find('article')
```

Requests to an API

```
import requests

url = 'https://gitlab.com/api/v4/projects/' + \
    'a.l.e%2Fadvent-of-code/repository/commits'
response = requests.get(url)

if response.ok:
    print(response.json()[0]['title'])
else:
    print(response.status_code)
```

- Often, you need to generate (and use) an authetncation token to access (REST) APIs
- Some websites make it harder to access *non official* APIs. As an example, the DuckDuckGo request we saw before *hides* the json data Javascript code.

Open Data

"Open data is data that is openly accessible, exploitable, editable and shared by anyone for any purpose. Open data is licensed under an open license." (Wikipedia ↗)

- Zürcher Open Data Katalog: https://data.stadt-zuerich.ch/
- Swiss Open Government Data: https://opendata.swiss

Caching the request

There is a requests-cache for a serious approach to caching your requests to an API.

But that's a bit overkill, if you're just playing around with your ideas and don't want to hit the real server each time you test a different BeautifulSoup query on a specific page.
Then, you can *simply* put the result of the request in a local text file and use that instead.

This is more important if you retrieving data from an API that enforces a tight access rate.

Caching the request

```
import requests
from pathlib import Path

requests_cache = Path('./requests-cache.html')
if not requests_cache.is_file():
    r = requests.get('https://www.tagesanzeiger.ch/' + \
        'die-gutheissungsquoten-sinken-bei-allen-' + \
        'richtern-404322581374')
    with open(requests_cache, 'w') as cache_file:
        cache_file.write(r.text)
with open(requests_cache, 'r') as cache_file:
    html = cache_file.read()

print(html)
```

And don't forget to disable the cache when the script goes into production!

For more information

- Beautiful Soup Documentation ↗