

Get Data

This section covers tools to get some data for your projects.

faker: Create Fake Data in One Line of Code

```
!pip install Faker
```

To quickly create fake data for testing, use faker.

```
>>> from faker import Faker
```

```
>>> fake = Faker()
```

```
>>> fake.color_name()
```

```
'CornflowerBlue'
```

```
>>> fake.name()
```

```
'Michael Scott'
```

```
>>> fake.address()
```

```
'881 Patricia Crossing\nSouth Jeremy, AR 06087'
```

```
>>> fake.date_of_birth(minimum_age=22)
```

```
datetime.date(1927, 11, 5)
```

```
>>> fake.city()
```

```
'North Donald'
```

```
>>> fake.job()
```

```
'Teacher, secondary school'
```

[Link to faker](#)

[Link to my full article on faker.](#)

Random User: Generate Random User Data in One Line of Code

Have you ever wanted to create fake user data for testing? Random User Generator is a free API that generates random user data. Below is how to download and use this data in your code.

```
import json
from urllib.request import urlopen

# Show 1 random users
data = urlopen("https://randomuser.me/api?results=1").read()
users = json.loads(data)["results"]
users
```

```
[{'gender': 'female',
  'name': {'title': 'Miss', 'first': 'Ava', 'last': 'Hansen'},
  'location': {'street': {'number': 3526, 'name': 'George
Street'},
  'city': 'Worcester',
  'state': 'Merseyside',
  'country': 'United Kingdom',
  'postcode': 'K7Z 3WB',
  'coordinates': {'latitude': '11.9627', 'longitude':
'17.6871'},
  'timezone': {'offset': '+9:00',
  'description': 'Tokyo, Seoul, Osaka, Sapporo, Yakutsk'}}},
'email': 'ava.hansen@example.com',
'login': {'uuid': '253e53f9-9553-4345-9047-fb18aec51cfe',
'username': 'heavywolf743',
'password': 'cristina',
'salt': 'xwnpqwtd',
'md5': '2b5037da7d78258f167d5a3f8dc24edb',
'sha1': 'fabbede0577b3fed686afd319d5ab794f1b35b02',
```

```
    'sha256':  
    'd42e2061f9c283c4548af6c617727215c79ecafc74b9f3a294e6cf09afc59  
06f'},  
    'dob': {'date': '1948-01-21T10:26:00.053Z', 'age': 73},  
    'registered': {'date': '2011-11-19T03:28:46.830Z', 'age':  
10},  
    'phone': '015242 07811',  
    'cell': '0700-326-155',  
    'id': {'name': 'NINO', 'value': 'HT 97 25 71 Y'},  
    'picture': {'large':  
'https://randomuser.me/api/portraits/women/60.jpg',  
    'medium':  
'https://randomuser.me/api/portraits/med/women/60.jpg',  
    'thumbnail':  
'https://randomuser.me/api/portraits/thumb/women/60.jpg'},  
    'nat': 'GB'}]
```

[Link to Random User Generator.](#)

fetch_openml: Get OpenML's Dataset in One Line of Code

OpenML has many interesting datasets. The easiest way to get OpenML's data in Python is to use the `sklearn.datasets.fetch_openml` method.

In one line of code, you get the OpenML's dataset to play with!

```
from sklearn.datasets import fetch_openml

monk = fetch_openml(name="monks-problems-2", as_frame=True)
print(monk["data"].head(10))
```

	attr1	attr2	attr3	attr4	attr5	attr6
0	1	1	1	1	2	2
1	1	1	1	1	4	1
2	1	1	1	2	1	1
3	1	1	1	2	1	2
4	1	1	1	2	2	1
5	1	1	1	2	3	1
6	1	1	1	2	4	1
7	1	1	1	3	2	1
8	1	1	1	3	4	1
9	1	1	2	1	1	1

Autoscraper

```
!pip install autoscraper
```

If you want to get the data from some websites, BeautifulSoup makes it easy for you to do so. But can scraping be automated even more? If you are looking for a faster way to scrape some complicated websites such as Stackoverflow, Github in a few lines of codes, try [autoscraper](#).

All you need is to give it some texts so it can recognize the rule, and it will take care of the rest for you!

```
from autoscraper import AutoScraper

url = "https://stackoverflow.com/questions/2081586/web-
scraping-with-python"

wanted_list = ["How to check version of python modules?"]

scraper = AutoScraper()
result = scraper.build(url, wanted_list)

for res in result:
    print(res)
```

```
How to execute a program or call a system command?
What are metaclasses in Python?
Does Python have a ternary conditional operator?
Convert bytes to a string
Does Python have a string 'contains' substring method?
How to check version of python modules?
```

[Link to autoscraper](#).

pandas-reader: Extract Data from Various Internet Sources Directly into a Pandas DataFrame

```
!pip install pandas-datareader
```

Have you wanted to extract series data from various Internet sources directly into a pandas DataFrame? That is when pandas_reader comes in handy.

Below is the snippet to extract daily data of AD indicator from 2008 to 2018.

```
import os
from datetime import datetime
import pandas_datareader.data as web

df = web.DataReader(
    "AD",
    "av-daily",
    start=datetime(2008, 1, 1),
    end=datetime(2018, 2, 28),
    api_key=os.getenv("ALPHAVANTAGE_API_KEY"),
)
```

[Link to pandas_reader.](#)

pytrends: Get the Trend of a Keyword on Google Search Over Time

```
!pip install pytrends
```

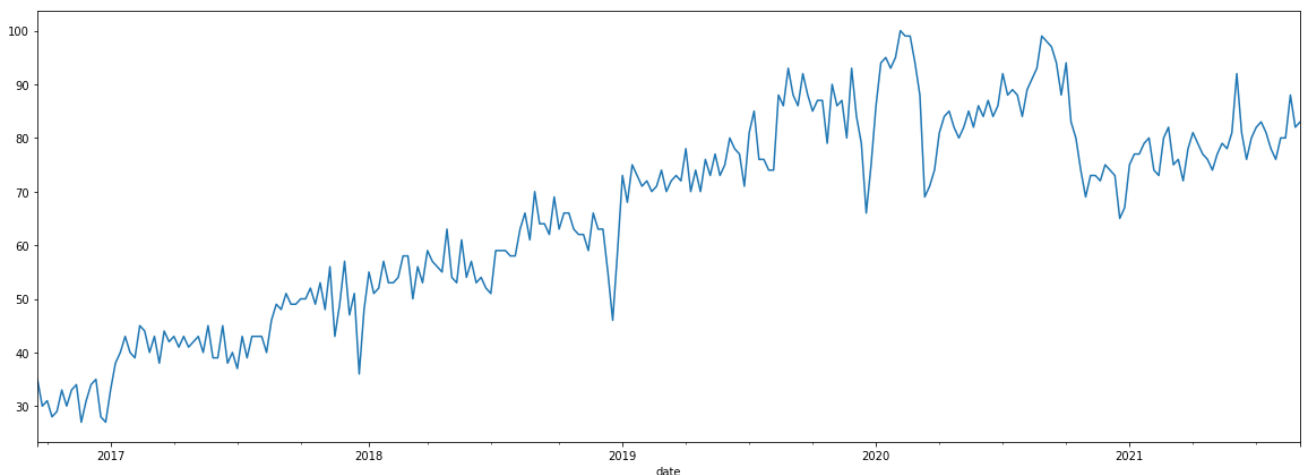
If you want to get the trend of a keyword on Google Search over time, try pytrends.

In the code below, I use pytrends to get the interest of the keyword “data science” on Google Search from 2016 to 2021.

```
from pytrends.request import TrendReq
```

```
pytrends = TrendReq(hl="en-US", tz=360)
pytrends.build_payload(kw_list=["data science"])

df = pytrends.interest_over_time()
df["data science"].plot(figsize=(20, 7))
```



[Link to pytrends](#)