Have you ever needed to quickly get the 3D structures and properties of a bunch of molecules for a research project or dataset? PubChem is a great free database, but manually looking up and downloading each molecule can be tedious. Luckily, Python makes it easy to automate this process!

In this post, I'll show you how to use the `pubchempy` library in Python to easily download 3D molecular structures in `.sdf` or `.mol` file formats along with other chemical properties from PubChem. We'll also save everything neatly into a CSV file for future use.

A Quick Intro to Molecular Structure Formats and PubChem

First, a quick overview for those new to this:

- `.sdf` or `.mol` files – Standard chemical file formats that store 3D molecular structures and properties. Visualized in chemistry software like CrysX-3D Viewer, VESTA, VMD, Avogadro, Chimera, Jmol, etc.
- PubChem – Massive public database of chemical compounds and their structures, properties, activities, and more. Maintained by the National Institutes of Health.
- pubchempy – Python library to access the PubChem API. Easily install with `pip install pubchempy`.

Got it? Okay, let's look at the code...

The Python Script to Download Structures and Info

Here is the full script I used. I'll walk through it below:

```
# Imports
import pubchempy as pcp
import os
import pandas as pd
import requests

# List of compounds to download
molecules = ['acetone', 'benzene', 'ethanol', 'methane', 'propane']

# Create output directories
sdf_dir = 'structures_sdf'

os.makedirs(sdf_dir, exist_ok=True)

# Lists to store properties
names = []
formulas = []
weights = []

# Function to download SDF
def get_sdf(cid):
    url =
f'https://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/cid/{cid}/SDF?record_type=3d'

    response = requests.get(url)
    return response.text
```

```
# Loop through compounds
for mol in molecules:

    # Get PubChem data
    compound = pcp.get_compounds(mol, 'name')[0]

    names.append(compound.iupac_name)
    formulas.append(compound.molecular_formula)
    weights.append(compound.molecular_weight)

    # Download structures
    sdf = get_sdf(compound.cid)

    with open(f'{sdf_dir}/{mol}.sdf', 'w') as f:
        f.write(sdf)

# Create dataframe
data = {'Name': names,
'Formula': formulas,
'Weight': weights}

df = pd.DataFrame(data)

# Export CSV
df.to_csv('compounds.csv', index=False)
print(df)
```

To keep it simple, I'm just downloading a few simple molecule names. But this can easily be extended to any list of compounds!

The key steps are:

1. Import `pubchempy` and `pandas`.
2. Define your list of compounds
3. Initialize some lists to store the extracted properties.
4. Loop through each compound name.
5. Use `pubchempy` to lookup the PubChem data.
6. Extract the iupac name, formula, and molecular weight.
7. Download the SDF using the custom function and the compound id obtained using `pubchempy`
8. Append the properties to our lists.
9. Print to confirm the download.
10. Create a Pandas DataFrame from the lists.
11. Export to `CSV` or Excel file!

And that's it! You've now automated downloading structures and info for any compounds you need.

Possible Use Cases

Automating this workflow has saved me tons of time on projects where I need to work with more than just a couple molecules. Some examples where this script has been useful:

- Seeding molecular dynamics simulations with diverse starting structures
- Benchmarking quantum chemistry methods on various organic molecule sets
- Collecting data for machine learning projects
- Populating an internal company database with molecules of interest

The key advantage is going from manual to automated downloading in just a few minutes of coding. No more clicking around websites or copying data by hand!

Basically, anytime you need bulk molecular data!

Wrapping Up

I hope this gives you a template to start automating your PubChem downloading for chemistry and drug discovery projects! Let me know in the comments if you have any other questions.

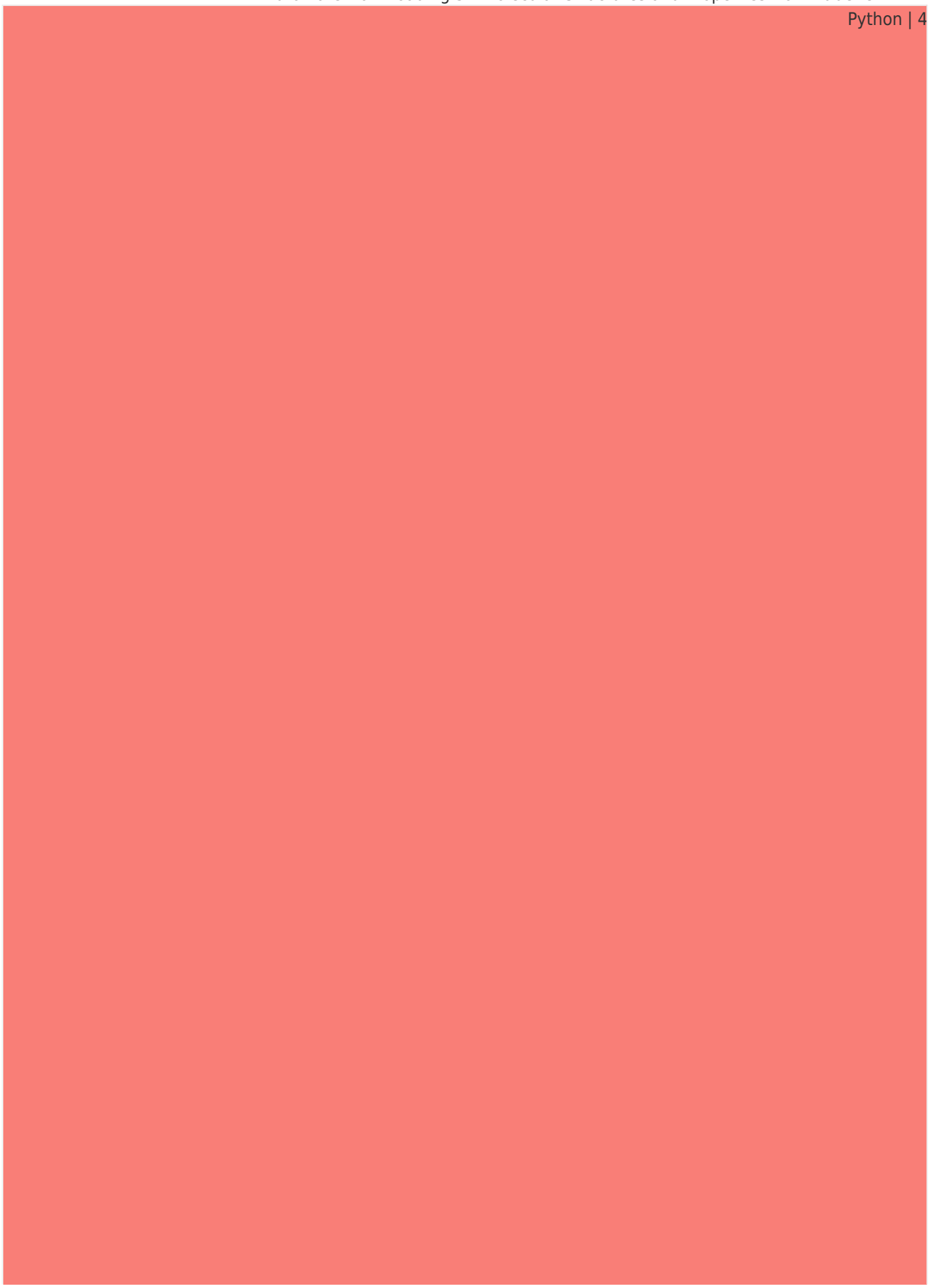
Don't forget to also check out the full `pubchempy` documentation and my other posts on related topics.

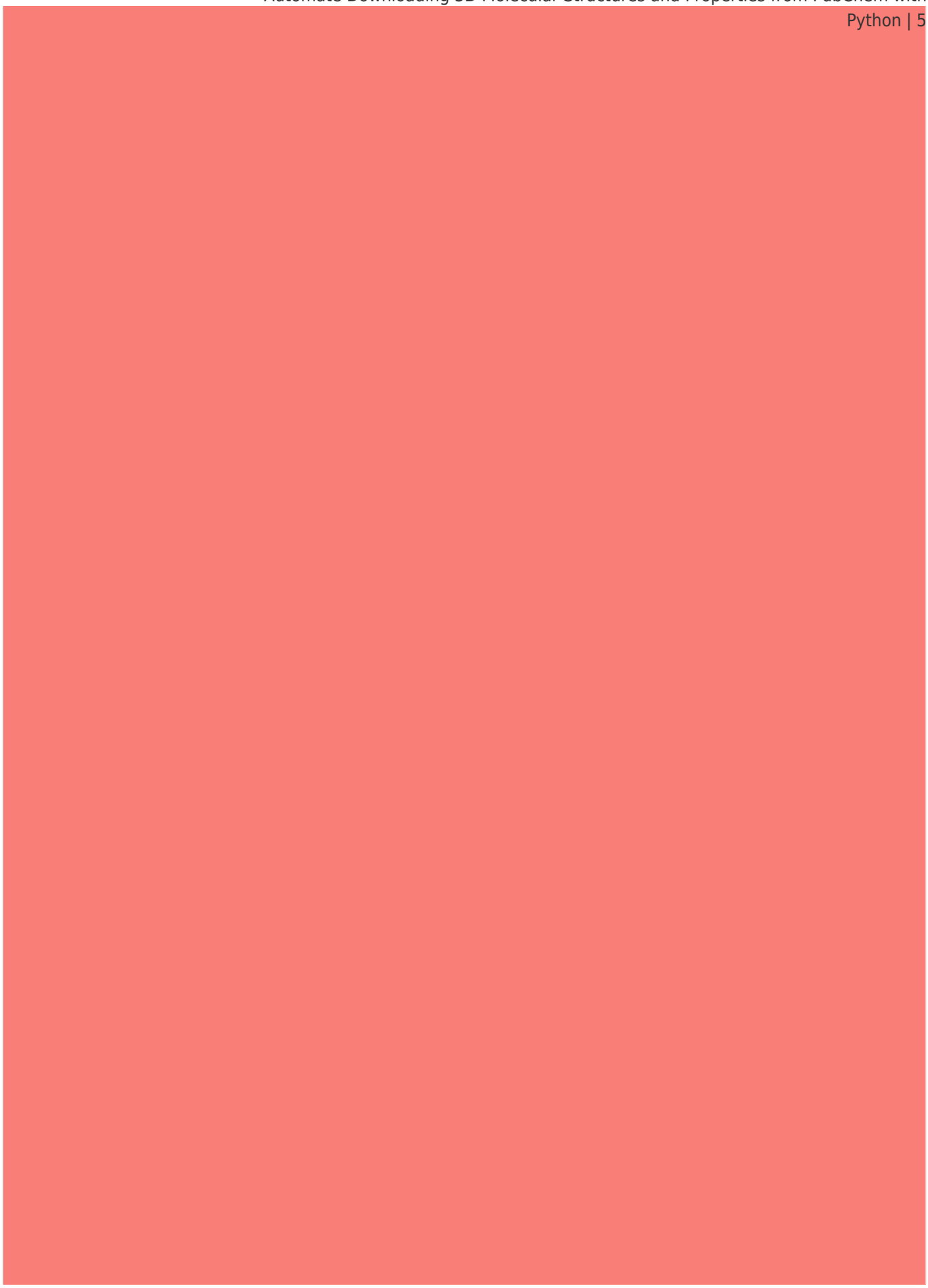
Thanks for reading! Let the compound mining begin...

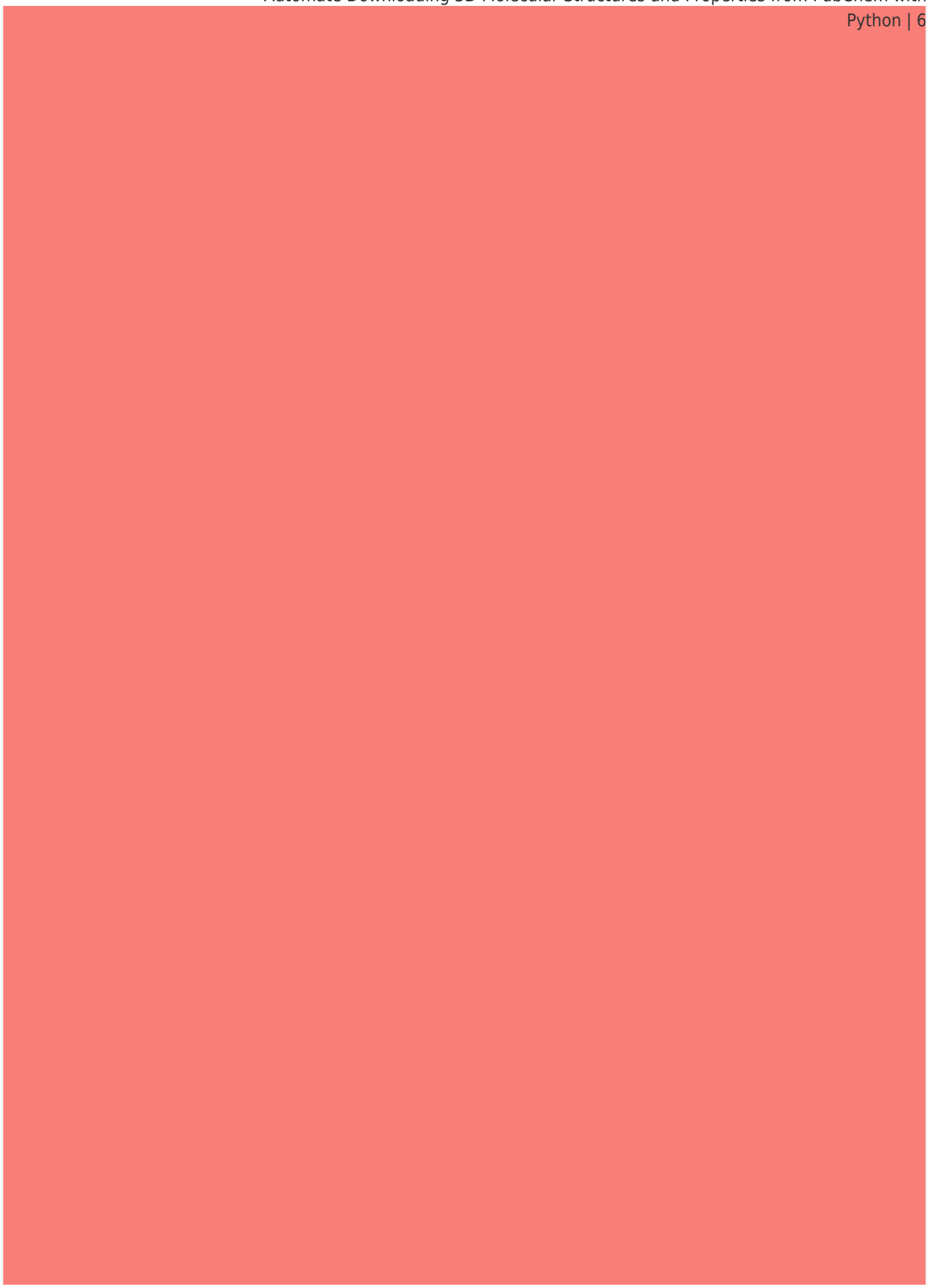
Manas Sharma

I'm a physicist specializing in computational material science with a PhD in Physics from Friedrich-Schiller University Jena, Germany. I write efficient codes for simulating light-matter interactions at atomic scales. I like to develop Physics, DFT, and Machine Learning related apps and software from time to time. Can code in most of the popular languages. I like to share my knowledge in Physics and applications using this Blog and a YouTube channel.

manas.bragitoff.com/







Share this:

Click to share on Facebook (Opens in new window)

Click to share on Twitter (Opens in new window)

Click to share on WhatsApp (Opens in new window)

Click to share on Pinterest (Opens in new window)

Click to share on Reddit (Opens in new window)

Click to share on LinkedIn (Opens in new window)

Click to email a link to a friend (Opens in new window)

Click to print (Opens in new window)

Click to share on Tumblr (Opens in new window)

Click to share on Pocket (Opens in new window)

Click to share on Telegram (Opens in new window)

[wpedon id="7041" align="center"]