How to's
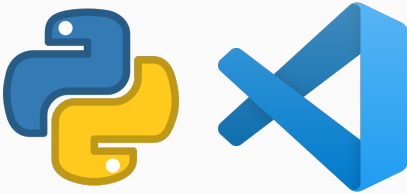
Installing Python, Visual Studio code, and Anaconda

Emilien Valat

August 29, 2022

University of Bristol, School of Economics

Before you get to write your first script, you need to install two pieces of software:

Make sure to click the icons ;)

It is the code interpreter: it will translate the code to machine language. I recommend downloading through ⟳ Anaconda.

Although Anaconda comes with Spyder, an Integrated Development Environment (IDE), I recommend installing Visual Studio Code (VsCode), as it is less clumsy than Spyder. to interface them, I recommend you go through this tutorial.

Once you
have installed your Python distribution, you can run it from a terminal:


```
Command Prompt

Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hx21262>python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> school = 'Economics'
>>> print(f'Hello Bristol School of {school}')
Hello Bristol School of Economics
>>> exit()

C:\Users\hx21262>
```
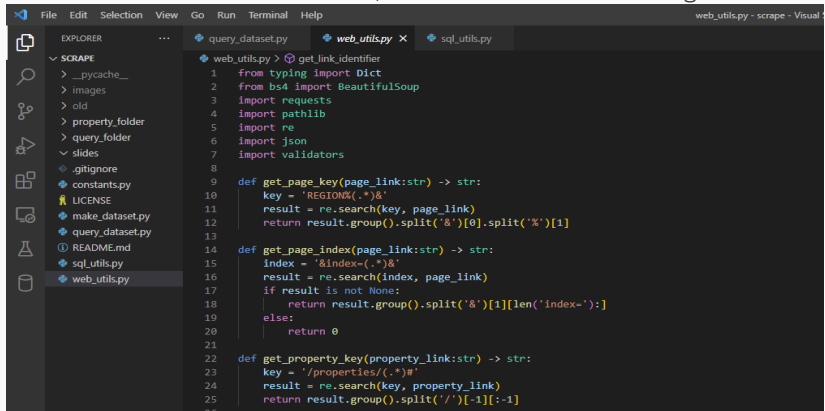
Developing without an IDE is such a 1960's thing. You need one to interact with the interpreter in a smooth fashion. I personally recommend Visual Studio Code, as it is free and well designed.

```python
from typing import Dict
from bs4 import BeautifulSoup
import requests
import pathlib
import re
import json
import validators

def get_page_key(page_link:str) -> str:
    key = 'REGION%(.*)&'
    result = re.search(key, page_link)
    return result.group().split('&')[0].split('%')[1]

def get_page_index(page_link:str) -> str:
    index = '&index=(.*)&'
    result = re.search(index, page_link)
    if result is not None:
        return result.group().split('&')[1][len('index-'):]
    else:
        return 0

def get_property_key(property_link:str) -> str:
    key = '/properties/(.*)#'
    result = re.search(key, property_link)
    return result.group().split('/')[-1][:-1]
```

4