WEB SCRAPING USING PYTHON

-KRUTI PATEL

WHAT ARE WE GOING TO LOOK TODAY……

- What is Web Scraping?
- Why Web Scraping?
- Why use Python for Web Scraping?
- Libraries in Python
- BeautifulSoup
- Python Virtual Environment
- First Web Scraper
- Advanced HTML Parsing
- Lots of Resources

WHAT'S EXACTLY WEB SCRAPING?

- **Web Scraping** is the practice of gathering data from web pages through any means other than using APIs or user doing manually.
- Web Scraping is accomplished by writing automated program that queries a web server, requests data and then parse the data to extract information.
- You could revisit your favourite website every time it updates for new information

Or,

You could write a **web scraper** to have it do it for you!

WHY DO WEB SCRAPING?

Web Scrapers are excellent at gathering & processing large amount of data quickly.

Web Scrapers can go places that traditional search engines cannot.

APIs may not always provide sufficient data for your purpose

WHERE WEB SCRAPING IN REAL LIFE?

- Extract product Information
- Extract job Postings and Internships
- Extract offers and discounts from deal-of-the-day websites
- Crawl forums and social websites
- Extract data to make search engine
- Gathering weather data
- Medical diagnostics
- Machine-language translations
- Market forecasting
- etc

WORKFLOW
ESSENTIAL PART OF WEB SCRAPING

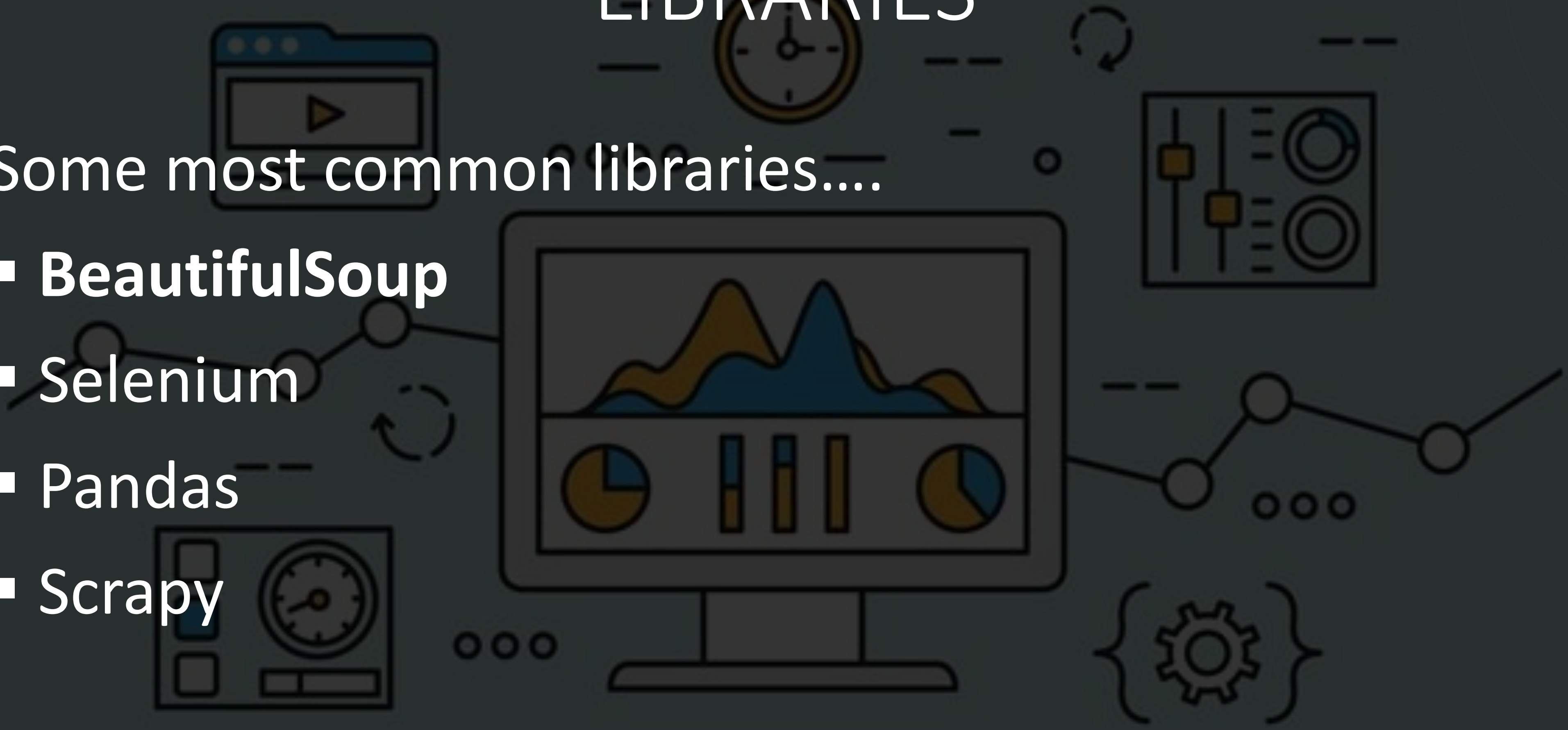
GET the website – using HTTP library

Parse the html document – using any parsing libraries

Store the results – either a db, csv, text, file, etc

WHY USE PYTHON?

- **Easy to use**:
  - ✓ Python is simple to code
  - ✓ NO Semicolons, NO curly-braces
  - ✓ Less Messy = Easy to Use
- **Large collections of libraries:**
  - ✓ Numpy, Matlplotlib, Pandas, etc
  - ✓ Suitable for Web Scraping and further manipulation
- **Dynamically typed:**
  - ✓ No defining variable, just directly use it when required
  - ✓ Saves time = Job faster
- **Easily understandable syntax**:
  - ✓ Reading Python Code is similar to reading statement in English
  - ✓ Expressive and easily readable
- **Small code, large task**
- **Community:**
  - ✓ Stuck while writing Code? Don't worry Python has biggest active community ready to help



LIBRARIES

Some most common libraries….

- **BeautifulSoup**
- Selenium
- Pandas
- Scrapy

BEAUTIFULSOUP

- Beautiful Soup is a Python library for pulling data out of HTML and XML files.
- It tries to make sense of nonsensical i.e. it helps to format & organize the messy web by fixing bad HTML & presenting us with easily traversable structure.
- It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree.
- It commonly saves programmers hours or days of work.

INSTALLATION

- Debian, Ubuntu or Linux

```
$ apt-get install python-bs4 (for Python 2)
$ apt-get install python3-bs4 (for Python 3)
```

- Macs

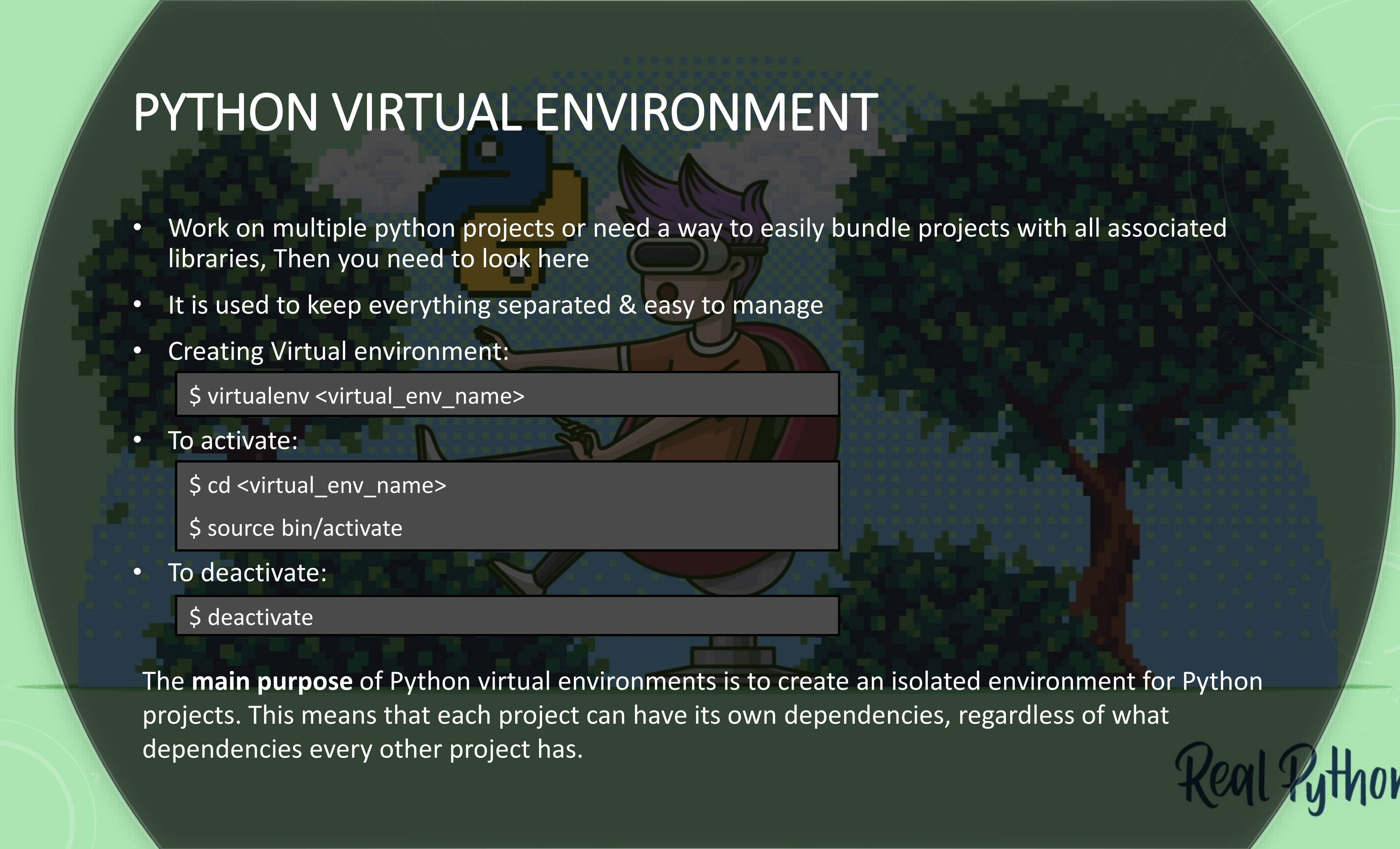
```
$ sudo easy_install beautifulsoup4
$ pip install beautifulsoup4
```

- Windows
  - Install pip:

https://github.com/BurntSushi/nfldb/wiki/Python-&-pip-Windows-installation

```
$ pip install beautifulsoup4
```

PYTHON VIRTUAL ENVIRONMENT



- Work on multiple python projects or need a way to easily bundle projects with all associated libraries, Then you need to look here
- It is used to keep everything separated & easy to manage
- Creating Virtual environment:

```
$ virtualenv <virtual_env_name>
```

- To activate:

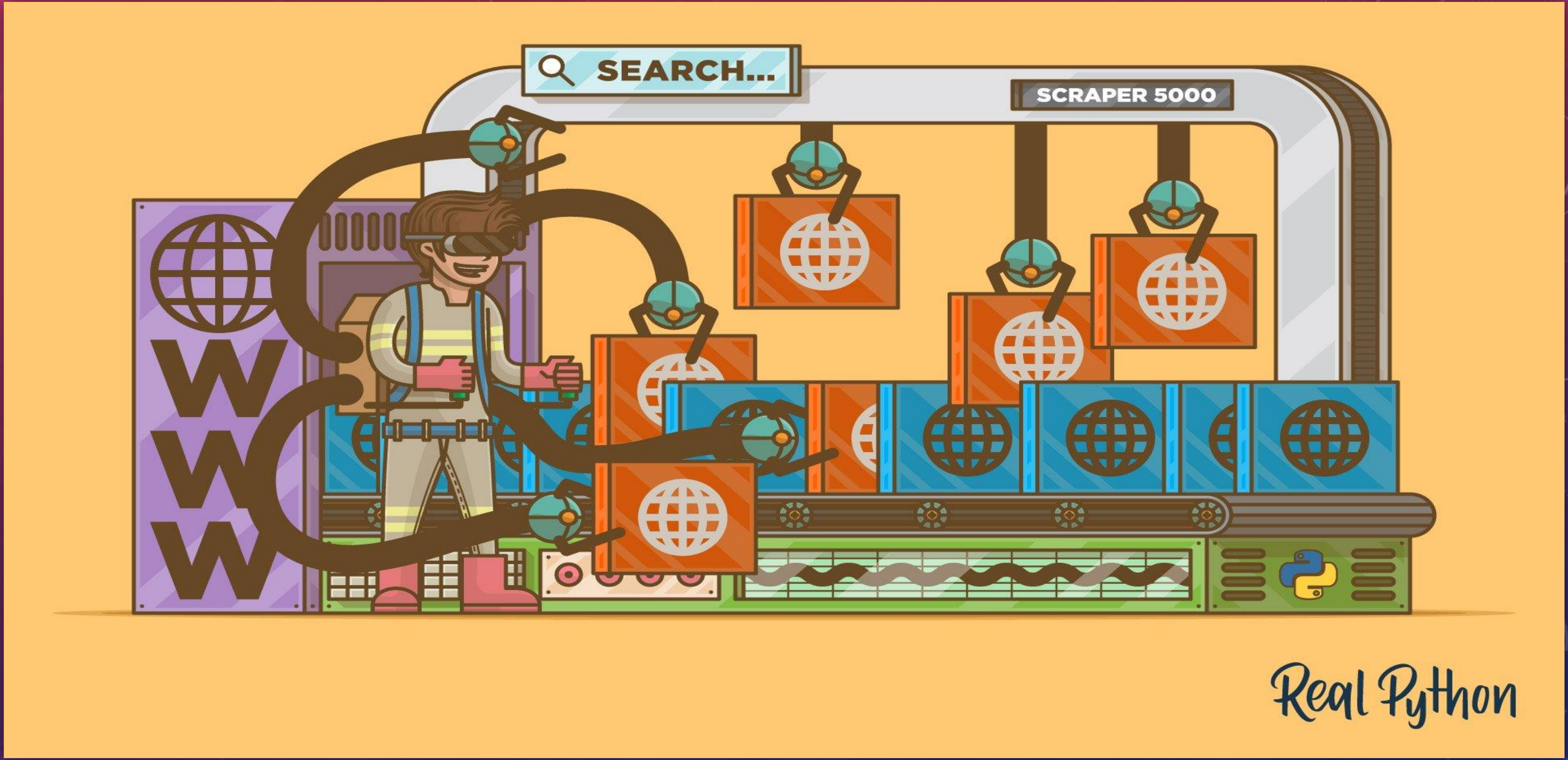
```
$ cd <virtual_env_name>
$ source bin/activate
```

- To deactivate:

```
$ deactivate
```

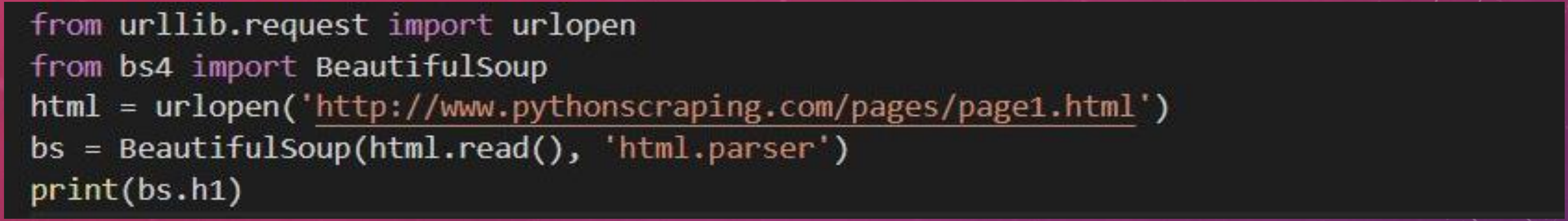
The **main purpose** of Python virtual environments is to create an isolated environment for Python projects. This means that each project can have its own dependencies, regardless of what dependencies every other project has.

Real Python



Let’s build your First Web Scraper

FIRST CODE…



```
from urllib.request import urlopen
from bs4 import BeautifulSoup
html = urlopen('http://www.pythonscraping.com/pages/page1.html')
bs = BeautifulSoup(html.read(), 'html.parser')
print(bs.h1)
```

1. **from urllib.request import urlopen**
   It looks into the python module request found within the urllib library and imports only the function urlopen

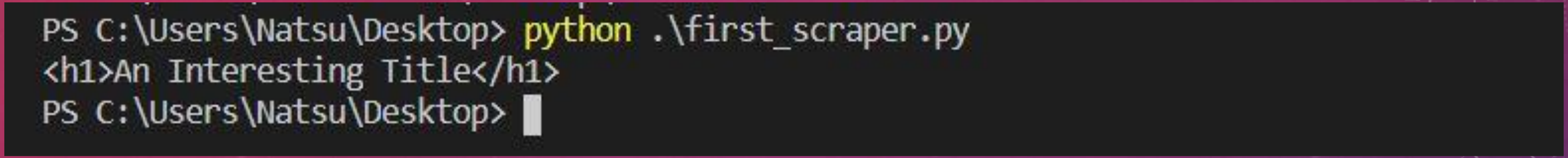
2. **from bs4 import BeautifulSoup**
   It imports BeautifulSoup function from bs4 library

3. **html = urlopen('http://www.pythonscraping.com/pages/page1.html')**
   Urlopen opens remote object across network and read it

4. **bs = BeautifulSoup(html.read(), 'html.parser')**
   html.read() – gets html content of the page and html.parser – parser you want bs4 to use to create object

5. **print(bs.h1)** – prints the h1 tag and its content from the parsed content

FIRST CODE

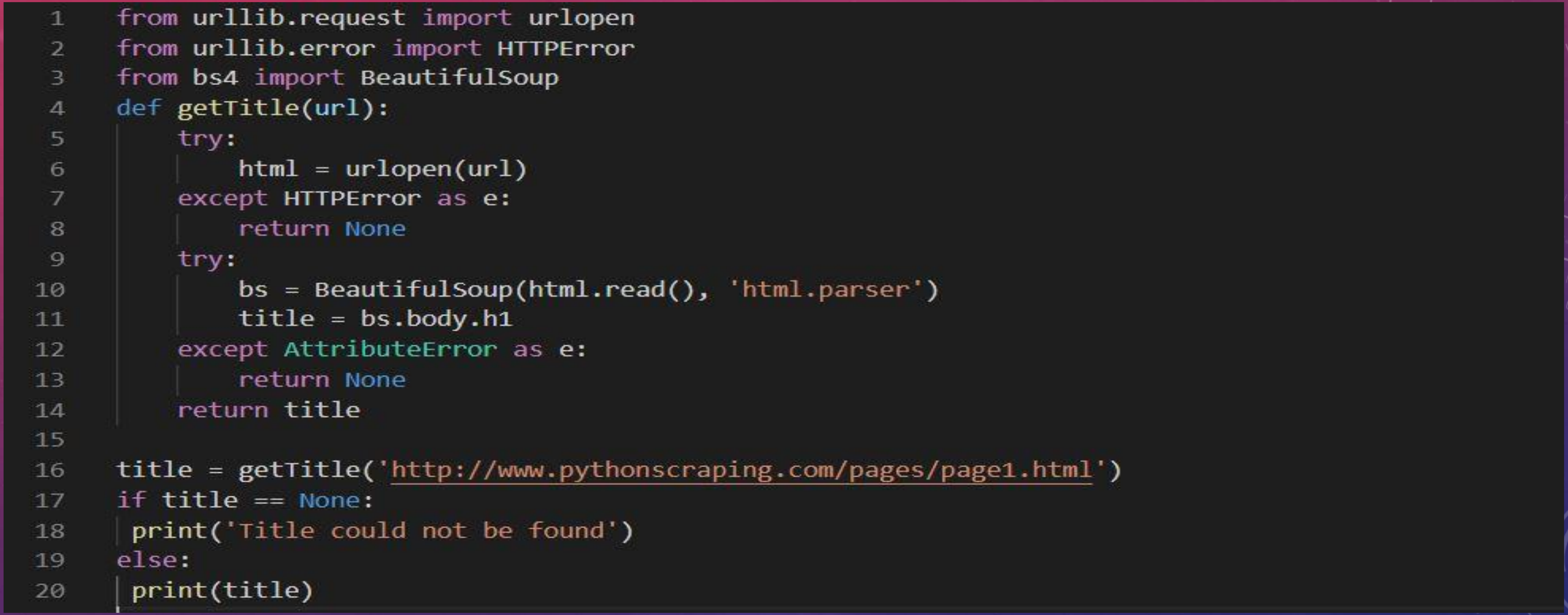


```
PS C:\Users\Natsu\Desktop> python .\first_scraper.py
<h1>An Interesting Title</h1>
PS C:\Users\Natsu\Desktop>
```

Different parsers that can be used with Beautifulsoup are as follows:

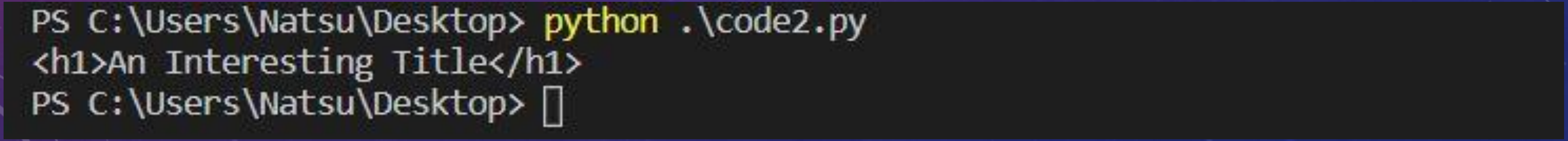
1. Html.parser() : default
2. Lxml
3. html5lib

HANDLING EXCEPTION:



```python
from urllib.request import urlopen
from urllib.error import HTTPError
from bs4 import BeautifulSoup
def getTitle(url):
    try:
        html = urlopen(url)
    except HTTPError as e:
        return None
    try:
        bs = BeautifulSoup(html.read(), 'html.parser')
        title = bs.body.h1
    except AttributeError as e:
        return None
    return title

title = getTitle('http://www.pythonscraping.com/pages/page1.html')
if title == None:
  print('Title could not be found')
else:
  print(title)
```



```
PS C:\Users\Natsu\Desktop> python .\code2.py
<h1>An Interesting Title</h1>
PS C:\Users\Natsu\Desktop> 
```

ADVANCED HTML PARSING

- For doing advanced HTML parsing we make use of mainly two functions in BeautifulSoup. They are:

  1. find (tag, attributes, recursive, text, keywords)
  2. find_all (tag, attributes, recursive, text, limit, keywords)

REGULAR EXPRESSIONS

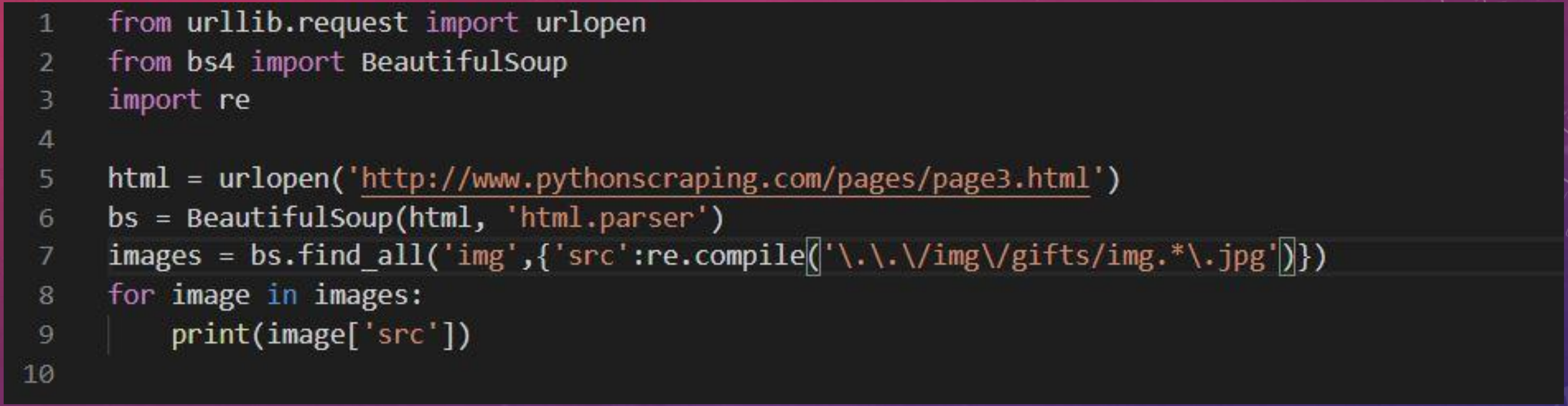
A regular expression can be inserted as any argument in the BeautifulSoup expression, allowing you to deal with a greater flexibility in finding target element.

Re is the regex library for python. It is used to extract only limited amount of text.

For re :

- You need to learn symbols
- Can become complex
- A standard library of python

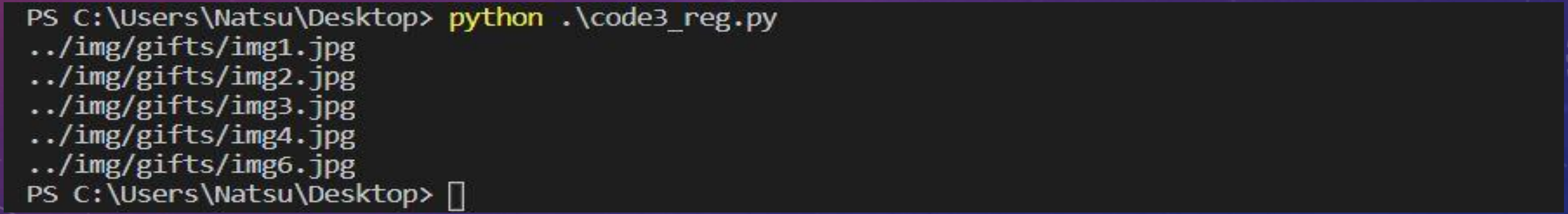
USING RE LIBRARY WITH BS4



```python
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

html = urlopen('http://www.pythonscraping.com/pages/page3.html')
bs = BeautifulSoup(html, 'html.parser')
images = bs.find_all('img',{'src':re.compile('\.\.\/img\/gifts/img.*\.jpg')})
for image in images:
    print(image['src'])

```



```
PS C:\Users\Natsu\Desktop> python .\code3_reg.py
../img/gifts/img1.jpg
../img/gifts/img2.jpg
../img/gifts/img3.jpg
../img/gifts/img4.jpg
../img/gifts/img6.jpg
PS C:\Users\Natsu\Desktop> 
```

USEFUL REFERENCES

- https://realpython.com/beautiful-soup-web-scraper-python/ - Site
- https://realpython.com/python-virtual-environments-a-primer/ - Site
- https://www.crummy.com/software/BeautifulSoup/bs4/doc/ - Site
- Web Scraping with Python – Book author- Ryan Mitchell
- https://www.edureka.co/blog/web-scraping-with-python/ - blog

THANK YOU!

https://github.com/Nickruti/