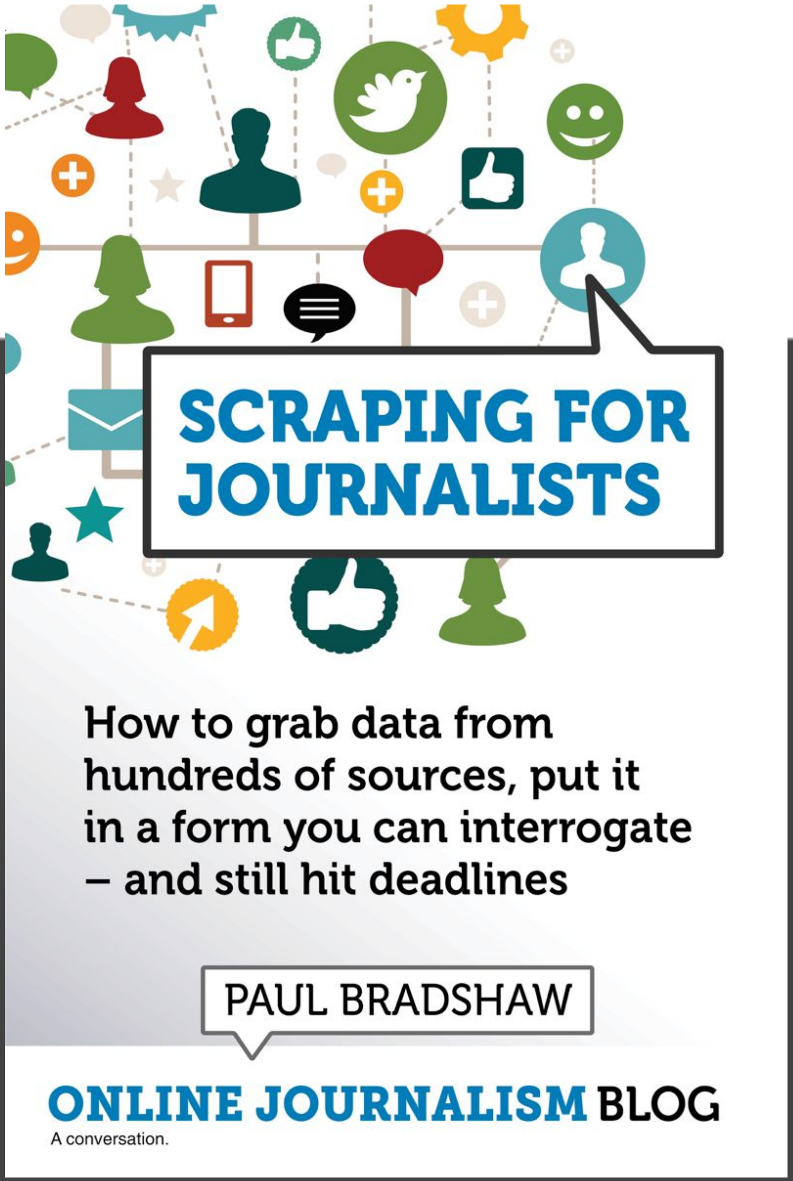
Coding in Google Colab: libraries


SCRAPING FOR JOURNALISTS

How to grab data from hundreds of sources, put it in a form you can interrogate – and still hit deadlines

PAUL BRADSHAW

ONLINE JOURNALISM BLOG

A conversation.

Paul Bradshaw

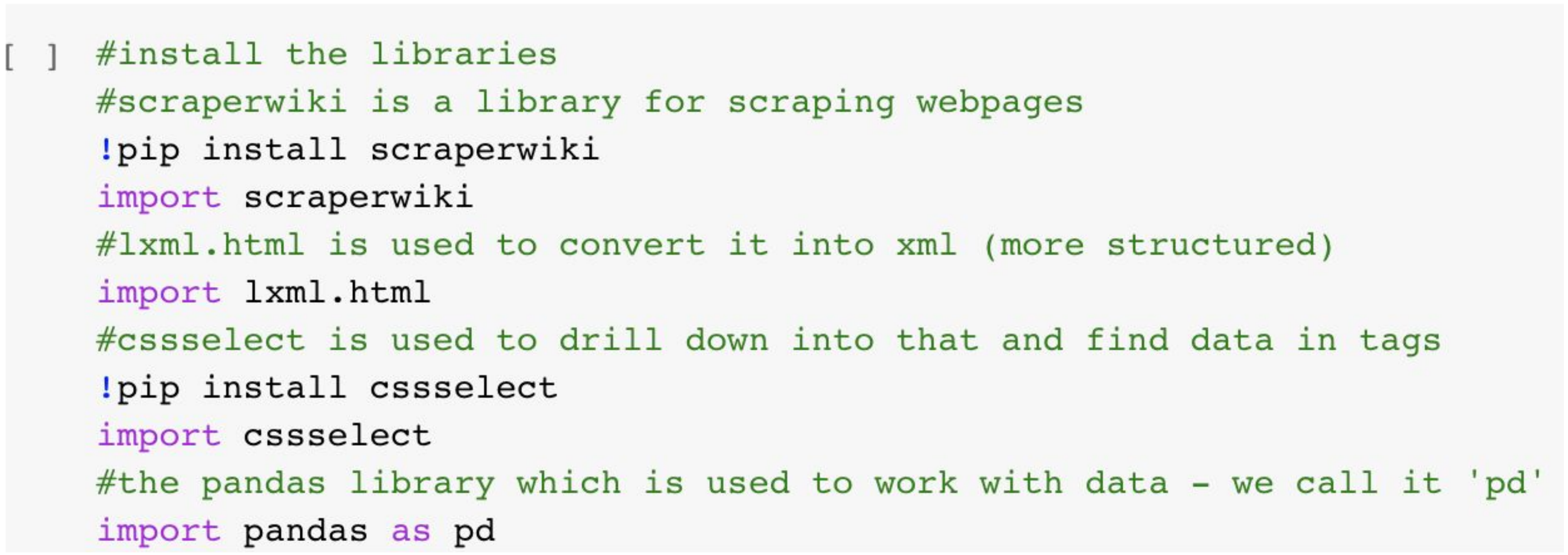
Leanpub.com/scrapingforjournalists

What we'll cover

- What are **libraries** in Python - and why you need to know
- How to **import** libraries in a Python notebook in Google Colab

Libraries

- A library is a **collection of recipes (functions)** and other stuff that someone has created for a particular type of problem
- Make it possible to ‘stand on the shoulders of giants’ & use code created by others
- E.g. the **scraperwiki** library is a collection of tools for solving scraping problems
- And **lxml.html** is a library for converting to XML
- **Pandas** is a library for data analysis
- **Matplotlib** is a library for visualisation
- **Re** for regular expressions (patterns)



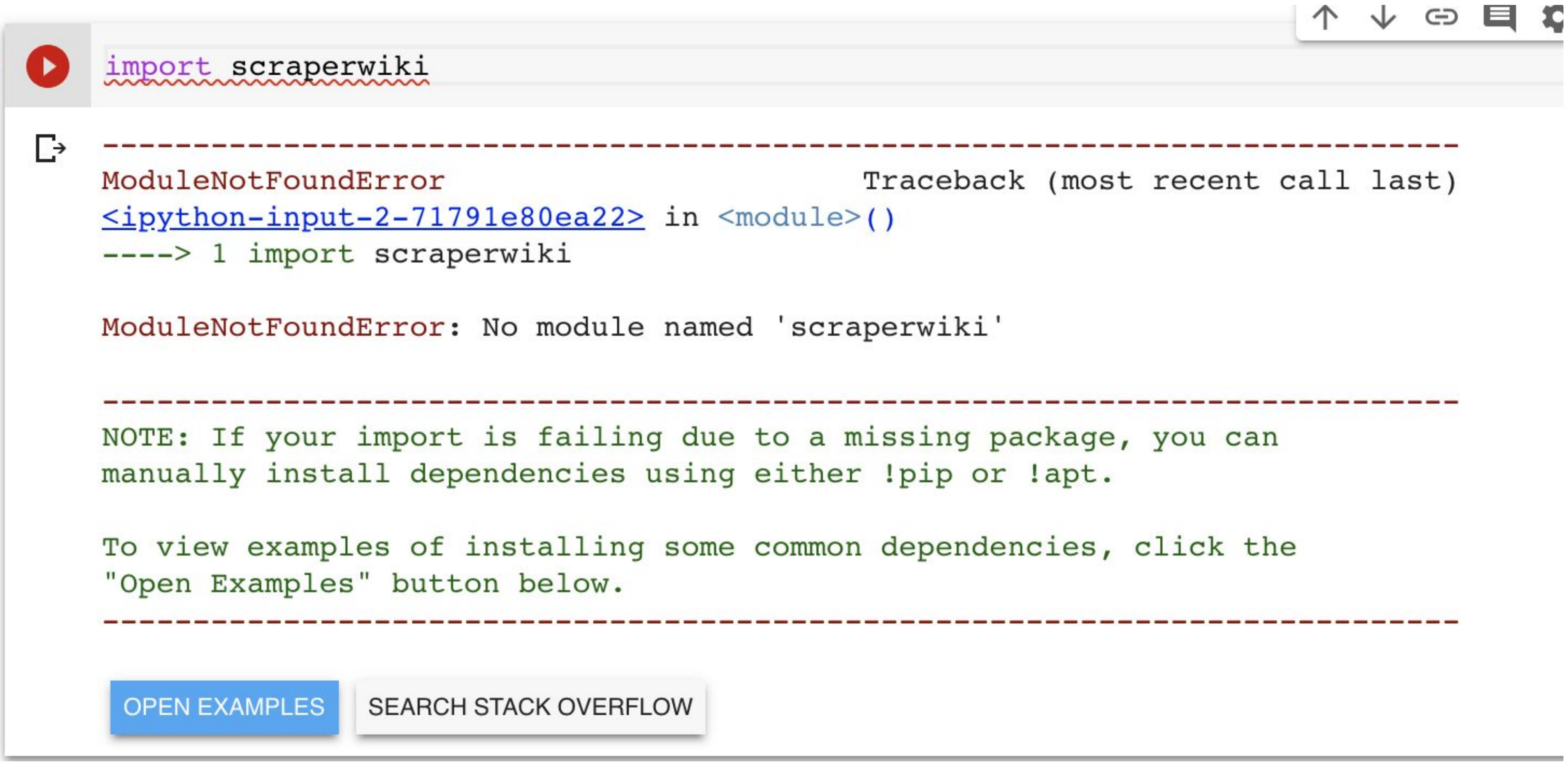
```python
#install the libraries
#scraperwiki is a library for scraping webpages
!pip install scraperwiki
import scraperwiki
#lxml.html is used to convert it into xml (more structured)
import lxml.html
#cssselect is used to drill down into that and find data in tags
!pip install cssselect
import cssselect
#the pandas library which is used to work with data - we call it 'pd'
import pandas as pd
```

Libraries… in Colab

- (Some) libraries need **installing** first
- (All) libraries need **importing**

(How do you know?)

Trial and error...



```
import scraperwiki
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-2-71791e80ea22> in <module>()
----> 1 import scraperwiki

ModuleNotFoundError: No module named 'scraperwiki'

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.
---------------------------------------------------------------------------
```

```
!pip install scraperwiki
import scraperwiki
import lxml.html
!pip install cssselect
import cssselect
```

pandas as pd?

- A library can be **renamed** at the same time as it is imported (typically with shorter names for convenience)
- ...because when you use a function from a library you need to name the library

```
import pandas as pd
```

Using a library

- When you use a **function** from a library you name the library and the function, with a period joining them:
- **scraperwiki.**scrape(fullurl)
- **lxml.html.**fromstring(html)
- **pandas.**DataFrame(columns=["title"])

  ...or if renamed when imported:
  pd. DataFrame(columns=["title"])

Library functions

- A function is always followed by parentheses to ‘pass’ any ingredients, e.g. =SUM(A1:A10)
- Library functions are attached to the library name with a period:
- scraperwiki.**scrape(fullurl)**
- lxml.html.**fromstring(html)**
- pd.**DataFrame(columns=["title"])**

Recap

- A library is (pre-)installed, and imported:

```
!pip install scraperwiki
import scraperwiki
```

- Functions (recipes) from that library are joined by a period and followed by parentheses:

```
html = scraperwiki.scrape(“http://blah.com”)
```

Try it now:

- Create a notebook and import the pandas library

Introducing pandas!

We need to store data

The pandas library has functions to create a data frame (table) and add to it

- The **`pandas.DataFrame()`** function creates a data frame with specified columns
- The **`.append()`** function adds extra rows to a data frame - those rows need to be stored in a dictionary

```
df = pandas.DataFrame(
columns=["service"] )
```

```
df = df.append(
  { "service" : servicename },
  ignore_index=True)
```

Introducing dictionaries!

The dictionary variable

- Uses **curly brackets**
- Contains a list of **pairs**, separated by a colon
- `{“name” : “Paul”, “age” : 21}`
- The first part of the pair is the **key**
- The second part is the **value**
- ...So they’re called **key-value pairs**
- The key is always a string; the value can be a string, number, True/False, or anything else
- Multiple dictionaries can be used to create rows in a table, e.g. row 2 might be:
  `{“name” : “Xian”, “age” : 31}`

Creating a dictionary

```
#create a dictionary
#with 2 key-value pairs
mydictionary = {“name” : “Paul”,
“age” : 21}
```

Expanding a dictionary

- `#create an empty dictionary`
  `mydictionary = {}`
- `#create a key and store a value`
  `mydictionary[‘name’] = “Paul”`
  `mydictionary[‘age’] = 21`
- `#print the dictionary`
  `print(mydictionary)`

We need to export the data

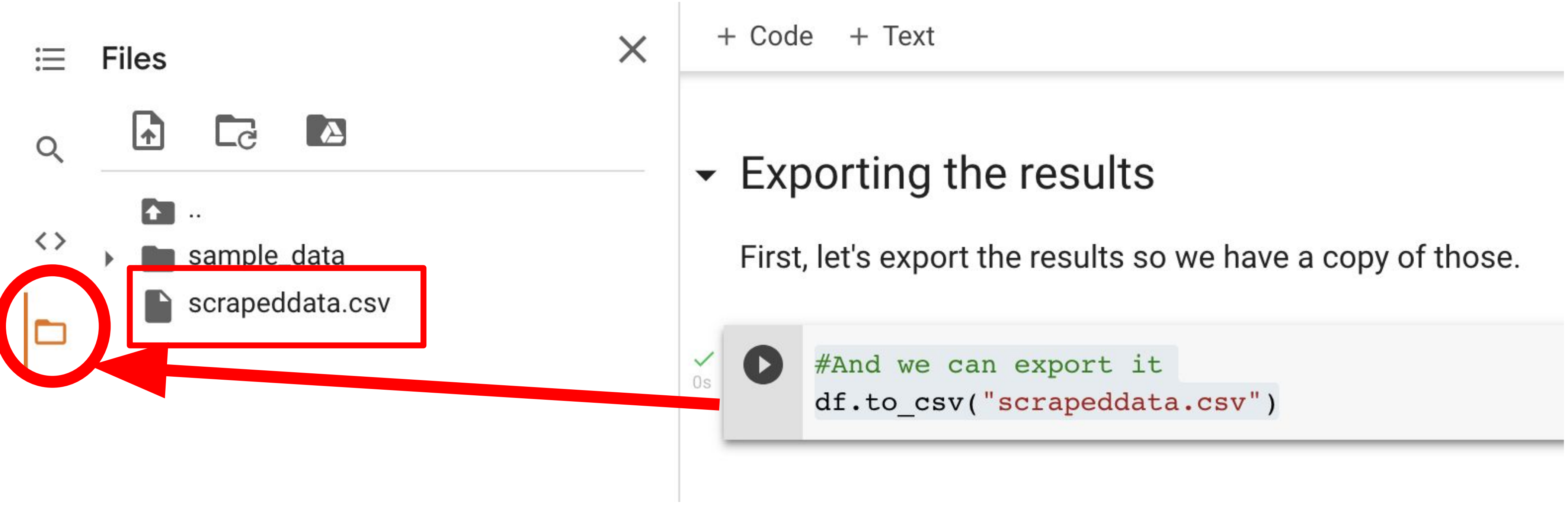
The pandas library has functions to import and export data to and from CSV

- The **`.tocsv()`** function creates a CSV with a specified name, using the data frame it’s attached to
  `mydataframe.tocsv(“mycsv.csv”)`

- The CSV file will be in the Files area in the left hand navigation in Colab

```
df.tocsv(“scrapeddata.csv”)
```

```
#export it
df.to_csv("scrapeddata.csv")
```



Exporting the results

First, let's export the results so we have a copy of those.

```
#And we can export it
df.to_csv("scrapeddata.csv")
```