## **Practical Python for News Investigations**

**Fall 2024**

[**bit.ly/24-fall-pp**](https://bit.ly/24-fall-pp)

## **Course Description**

Investigative journalists often find themselves with thousands of pages of documents that are impossible to manually analyze in any timely way. Or they have to compile their own datasets using information buried across websites. These challenges won't derail modern journalists who know how to tap intuitive programming languages like R or Python.

In this 3-credit advanced course, students learn to apply practical Python programming skills to tackle such hurdles. Students will learn core investigative programming skills, including web scraping to acquire data and natural language processing – a technique to convert words and phrases into data points to measure their frequency or to analyze their context. Imagine being able to quickly find every name and telephone number in thousands of pages of court depositions. This course will teach you how to find and tap Python libraries to tackle common investigative processes and obstacles.

And with the advent of Artificial Intelligence systems, journalists can now more easily harness code to reinforce their reporting. AI still doesn't provide the complete solutions so we'll learn techniques to best tap AI.

## **Learning Objectives**

* Leap from being a beginner to intermediate-Advanced Python journalist-programmer.
* Gather data scattered across thousands of pages.
* OCR (optical-character recognition) thousands of PDFs hold images of words and numbers.
* Extract text from PDFs (digital PDFs and scans)
* Search for patterns in text and numbers using Regular Expressions in Python.
* Elevate your Pandas programming for more complex data analysis.
* Blend Python functions and Pandas methods to simplify complex operations.
* Find and tap Python libraries to tackle common investigative processes.
* Become confident in problem solving any Python obstacles you might encounter.
* Learn to ask AI the right questions to help improve your code or problem solve.

## **Importance of Diversity & Inclusion**

It's critical that students learn to include a diverse set of voices in their stories -- something that is often glossed over when telling stories through charts and graphics. You are encouraged to look for stories about and voices from communities that are underrepresented. Diversity and inclusion here refers to race, gender, age, people with disabilities, and those at various economic and education levels living across New York city.

## **Class Meeting Dates**

| **Monday classes** |  |
| --- | --- |
| Week 1 | 09/09/2024 |
| Week 2 | 09/16/2024 |
| Week 3 | 09/23/2024 |
| Week 4 | 09/30/2024 |
| Week 5 | 10/07/2024 |
| **NO CLASS** | **10/14/2024** |
| **Week 6 Tuesday** | **10/15/2024** |
| Week 7 | 10/21/2024 |
| Week 8 | 10/28/2024 |
| Week 9 | 11/04/2024 |
| Week 10 | 11/11/2024 |
| Week 11 | 11/18/2024 |
| Week 12 | 11/25/2024 |
| Week 13 | 12/02/2024 |
| Week 14 | 12/09/2024 |
| Week 15 | 12/16/2024 |

## **Instructor**

**Sandeep Junnarkar** is the director of Data Journalism Journalism (and a founding faculty member) at the Newmark Graduate School of Journalism. He also works at Bloomberg News where he's a global newsroom trainer for data journalism and a special projects editor. He formerly worked at The New York Times as a Web editor and writer. Junnarkar has won numerous awards for his investigative reporting. His most recent project, AI Monitors, received a Knight Foundation grant to train community reporters about algorithmic accountability.

## **Office hours**

By appointment…I'm very flexible.

## **Instruction Philosophy**

1. **Instruction**: You'll learn new material each week in class.
2. **In-Class Challenges**: During class time, you'll be challenged to apply those lessons to more sophisticated, journalism-related exercises.
3. **Homework & Assignments**: You will demonstrate that you can apply those instructions material by completing exercises and major assignments outside of class.

## **Required Course Materials**

All the required material is free.

* GitHub student [developer pack](https://education.github.com/pack)
* GitHub copilot [free for students](https://docs.github.com/en/copilot/quickstart)
* VSCODE (you will receive instructions from your instructor)
* Anaconda
* Jupyter Notebooks
* Terminal

## **GitHub Course Repo**

Each week you will find updated homework, in-class demos/exercise files in [this GitHub course repo](https://github.com/sandeepmj/2024-practical-python).

## **Class Recordings**

Our class recordings can be [found in this folder](https://drive.google.com/drive/folders/1AKcXqfoonws2xPuYgGkMT05UEy84ltJ4?usp=drive_link). Please note that this is for review purposes and not a reason to miss class.

## **Zoom Link**

While this class has required in-person attendance (as mandated by CUNY), please don't come to school if you are sick. Inform me by email that you are sick and if you feel up to it, you may attend class [via this zoom link](https://journalism-cuny.zoom.us/j/88083628202?pwd=flITIy2Cy7MxEHMXvPAGlS5q5jxTjQ.1).

## **Submission form**

ALL homework, pitches and assignments must be filed [in this submission form](https://docs.google.com/forms/d/e/1FAIpQLSeQZ-7ogcPPKydPSE_Ediy61P73aad0_IKReD5fb-X98vfXtw/viewform).

## **Homework**

You will have a homework Jupyter notebook to complete each week to cement the week's lesson. There are two steps:

1. File your homework solution Jupyter notebooks every Friday by 11:00 pm in the [submission form](https://docs.google.com/forms/d/e/1FAIpQLSeQZ-7ogcPPKydPSE_Ediy61P73aad0_IKReD5fb-X98vfXtw/viewform).
2. After I post the solutions each Saturday morning, you write an assessment (maximum one or two short paragraphs) in which you compare my solution to your solution. You must file this assessment by Sunday at 11 pm also in the [submission form](https://docs.google.com/forms/d/e/1FAIpQLSeQZ-7ogcPPKydPSE_Ediy61P73aad0_IKReD5fb-X98vfXtw/viewform).

## **Assignments**

Students are encouragedto align their assignments in this course with their capstone projects or major assignments in other courses in their subject concentrations.

### **No. 1 — Web Scrape**

Scrape data from a multi-page website. It can involve downloading documents or creating a spreadsheet that contains custom columns of aggregated data. Ideally this data is required for a long term project and is not simply an exercise.

* **Milestones**
  + Plan Due: by October 20 at 11 pm ([**file here**](https://docs.google.com/forms/d/e/1FAIpQLSeQZ-7ogcPPKydPSE_Ediy61P73aad0_IKReD5fb-X98vfXtw/viewform)**)**
  + Code Due: by Nov. 10 at 10 p.m. ([**file here**](https://docs.google.com/forms/d/e/1FAIpQLSeQZ-7ogcPPKydPSE_Ediy61P73aad0_IKReD5fb-X98vfXtw/viewform)**)**

### **No. 2 — Text Analysis**

Extract text or tables from multiple PDFs or other documents to quantifyoccurrences, search for names of people, or for entities like companies, etc. You may need to scrape a site to first acquire these documents. You may do that step in Assignment 1.

* **Milestones**
  + Plan Due: by November 30 at 10 pm([**file here**](https://docs.google.com/forms/d/e/1FAIpQLSeQZ-7ogcPPKydPSE_Ediy61P73aad0_IKReD5fb-X98vfXtw/viewform)**)**
  + Results Due: by December 20 at 10 p.m. ([**file here**](https://docs.google.com/forms/d/e/1FAIpQLSeQZ-7ogcPPKydPSE_Ediy61P73aad0_IKReD5fb-X98vfXtw/viewform)**)**

### **No. 3 — Data Memo**

Please list as bullet points all the ways in which you applied applied the lessons/techniques from this course to further your work in your capstone, investigations or other course work:

* **Milestones**
  + Plan Due: by November 29 at 10 pm ([**file here**](https://docs.google.com/forms/d/e/1FAIpQLSeQZ-7ogcPPKydPSE_Ediy61P73aad0_IKReD5fb-X98vfXtw/viewform)**)**
  + Submission Due: by December 20 at 10 p.m. ([**file here**](https://docs.google.com/forms/d/e/1FAIpQLSeQZ-7ogcPPKydPSE_Ediy61P73aad0_IKReD5fb-X98vfXtw/viewform)**)**

### 

### **Rubric for assignments 1 & 2**

| **Element** | **Total possible value** |
| --- | --- |
| Does the code work? Does it result in the desired outcome without throwing errors? | 80 |
| Is the code concise and efficient? | 5 |
| Is the code commented in a manner that a peer could easily understand your thought process? | 3 |
| Is there an element in your code that shows you built up from the class's foundational material and did something I find impressive? | 5 |
| Was the assignment (and its associated pitch) filed on time? | 7 |
| **SUBTOTAL** | **100** |

### **Assignment Scale**

Grading for individual assignments is based on the quality of the finished work:

| **Grade** | **Quality** |
| --- | --- |
| A | Clean and efficient code that works, and others can read and understand |
| B | It works, but needs a good scrubbing to be readable and efficient |
| C | Kind of works, kind of doesn't work. Reads like painful credit card fine print! |
| D | Does not work and barely an attempt to make it work. |
| F | Atrocious, non-existent effort |

### 

### **Assignment Percentage Value**

| **Value (%)** | **Description** |
| --- | --- |
| 10 | Attendance, preparedness and participation |
| 25 | Homeworks |
| 25 | Web Scrape |
| 25 | Text Analysis |
| 15 | Data Memo |
| **100** | **Total** |

## 

## **Week-by-Week Outline**

### **Week 1 – Overview and Setup, Python Warm up**

#### **In-class**

* Introductions
* Course overview
* Foundational Python (data structures)

### **Week 2 – Flow, Loops and Control Structures**

#### **In-class**

* Conditional statements (if, elif, else);
* Handling exceptions with try except

### **Week 3 – Web** **Scraping for Data 1 (single page scrapes)**

#### **In-class**

* Understanding how HTML holds content on a webpage and CSS creates categories of info using classes and IDs.
* Introduction to the Python BeautifulSoup library by scraping a single webpage.
* We target specific elements on the page and export to a CSV file.
* Scraping pitfalls - certifying the integrity of the acquired data.

### **Week 4 – Web** **Scraping for Data 2 (multiple page scrapes)**

#### **In-class**

* We tap BeautifulSoup for a multi-page scrape
* Add random delays to minimize impact on the host’s servers.
* Schedule scrapes to run on a schedule

### **Week 5 –** **Web Scraping for Data 3 (downloading documents)**

#### **In-class**

* We review scraping 1 & 2
* We scrape to download documents (like PDFs or text files) from web pages.

### **Week 6 – Methods & Functions**

#### **In-class**

* Write functions to produce efficient code;
* Write functions that trigger other functions;
* Return values that can be used for more calculations;
* Incorporate flow and control in functions.

### **Week 7 –** **Working with Text files**

#### **In-class**

* Text files can contain reams of unstructured data.
* We learn to scrape data from a large volume of text files

### **Week 8 - Working with PDF tables**

#### **In-class**

* We learn to extract tabular data from PDFs.

### **Week 9** **–** **Obnoxious PDFs**

#### **In-class**

* Scrape data from troublesome PDFs and scanned documents.

### **Week 10** **–** **Natural Language Processing**

#### **In-class**

* Use Spacy to analyze text
* Preparing text for analysis (tokenize, lowercase, remove stop words, etc)
* Frequency analysis
* Topic segmentation
* Entity recognition (people, places, companies, monetary quantities)
* Pitfalls in text analysis - Do these findings even matter?

### **Week 11 –** **Search and Capture Using** **Regular Expressions I**

#### **In-class**

* Introduction to Regular Expressions
* Using Regex to capture patterns that involve text, numbers, spaces, etc.
* Return capture groups.
* We incorporate Python and Regex using the re library to find patterns.

### **Week 12 –** **Search and Capture Using** **Regular Expressions II**

#### **In-class**

* Introduction to Regular Expressions
* Using Regex to capture patterns that involve text, numbers, spaces, etc.
* Return capture groups.
* We incorporate Python and Regex using the re library to find patterns.

### **Week 13–** **TBD based on what we have accomplished**

### **Week 14 – TBD based on what we have accomplished**

#### 

### **Week 15 – TBD based on what we have accomplished**