

# SQL & Python for Analytics

USC Annenberg Digital Lounge  
Week 3: Intro to Python

# Today's session

This is going to be interactive, so please be ready to try out the exercises as we go!

We are going to be using a tool called Hex, which is a tool for using SQL and Python for doing data analysis and visualization.

Shout-out to Hex for giving us their Professional Plan for free!

Go to [app.hex.tech](https://app.hex.tech) and sign up with your USC email address or Microsoft login. When asked to select a workspace, choose Annenberg Digital Lounge:

**USC Annenberg Digital Lounge**

53 members | You are an Editor



# Goals

You will learn:

- What python can do for data analysis, and how to identify when to use it
- How to read python code and understand what it's doing
- How to write and edit simple python code for data analysis
- How to figure out what functions you need, when you need them

You will not learn:

- How to write complex code, apps, or websites in python
- Every piece of syntax

# What is python?

Python is a programming language

- Commonly used for scripting

**Scripting** is writing a little bit of code for a task, without coding an entire application or website. It's very powerful for data analysis.

Python is used by:

- Software engineers
- Data scientists
- Data analysts
- And more!

[python.org](https://python.org)

---

*"Python is all about automating repetitive tasks, leaving more time for your other SEO efforts."*

---

*Marnix de Munck, Sooda internetbureau*

# Python for data analysis

We'll primarily be talking about how to use python for data analysis

- In python, we interact with data in dataframes.

A **dataframe** is a table of data, very similar to a SQL table or an Excel spreadsheet

You can use python to:

- Create, import, and export dataframes
- Add to, edit, and transform dataframes
- Join and summarize dataframes
- And more!

# What are python scripts used for?

Find data that meets criteria	Anything you can write a SQL query for, you can write in python as well
Clean and format data	Python has a much wider variety of functions for cleaning and manipulating data
Run functions on data	Define your own transformation to data, and apply to to as many columns and dataframes as you want easily!
Extract data for display	Pull specific points from a dataset to display on a dashboard or final report
Prepare data for use in SQL	SQL is easier to use and more well-known, so any work you do in python is still accessible to those folks!

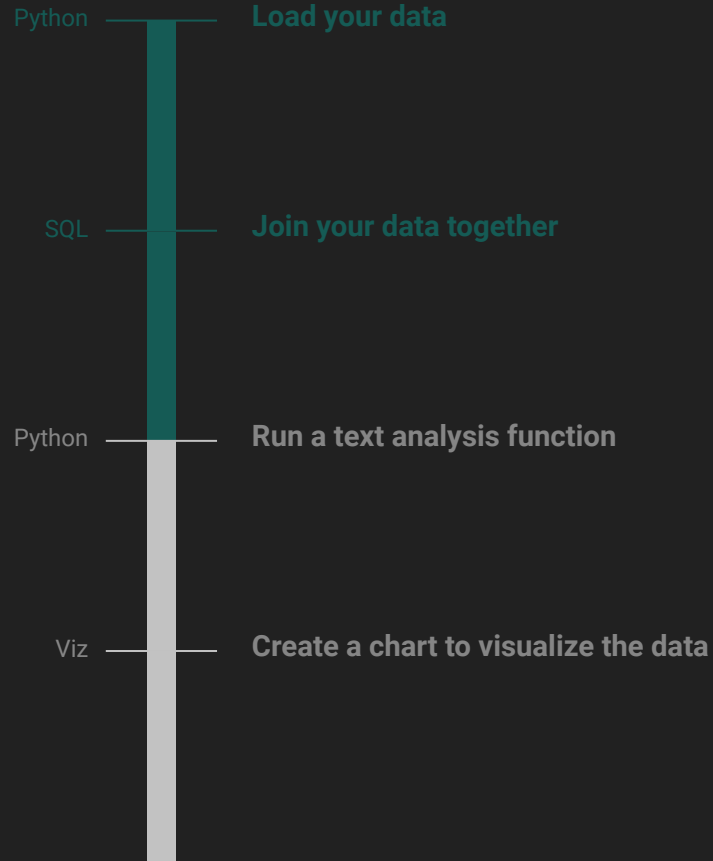
# What are python scripts used for?

*“There’s got to be a better way to do this.”*

- you, doing a repetitive task in Excel, or trying to write complicated SQL, or updating the same report repeatedly

# What is Hex?

- Hex is a SQL and Python notebook tool
- It works as a series of 'cells' that link together
- Here's an example of some cells:





# Activity

1. Go to Projects, next to Week 3 hit '...' then Duplicate
2. Load your datasets
  - a. We are going to load them with Python, we'll discuss how it works shortly
  - b. Your first cell should look like this – click into it and hit Run in the upper right, or Command-Enter

Projects All projects ▾

Week 1: Intro to SQL

Copy this template to start for Week 1!

📄 🔒 📄 Template

↗ Open in new tab

🔗 Copy link

📄 Duplicate

Code 1

```
1 import pandas as pd
2 transcripts_data = pd.read_csv('https://query.data.world/s/2svp7yxiggyn4eui6wvklptslqctle?dws=00000')
3 ratings_data = pd.read_csv("SeinfeldRatingsData.xls")
```

↳ pd transcripts\_data ratings\_data

# Starting out with python

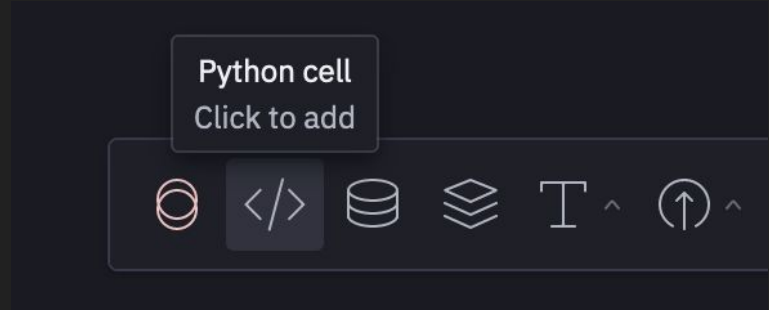
Anything you write in python can be wrapped in `print(whatever code you wrote)` to see the results. Hex will often preview results for you without that, just by running your code (Command-Enter).

A few key terms to know:

- **Dataframe**: a “structure” you can access with code by rows and columns. There are many types of “structures”; broad term for them all is **Objects**.
- **Function**: a series of steps that you define, to manipulate inputs and produce an output
- **Library**: a bunch of functions somebody else already wrote, that you can just use!

# Write your first bits of code

- Add a new Python cell to your Hex project



- Check out one of the columns of your dataset:

```
ratings_data['Title']
```

- To run your code, click Run on that cell, or hit Command-Enter

Add another new Python cell

- Check out one of the rows of your dataset:

```
ratings_data.iloc[0]
```

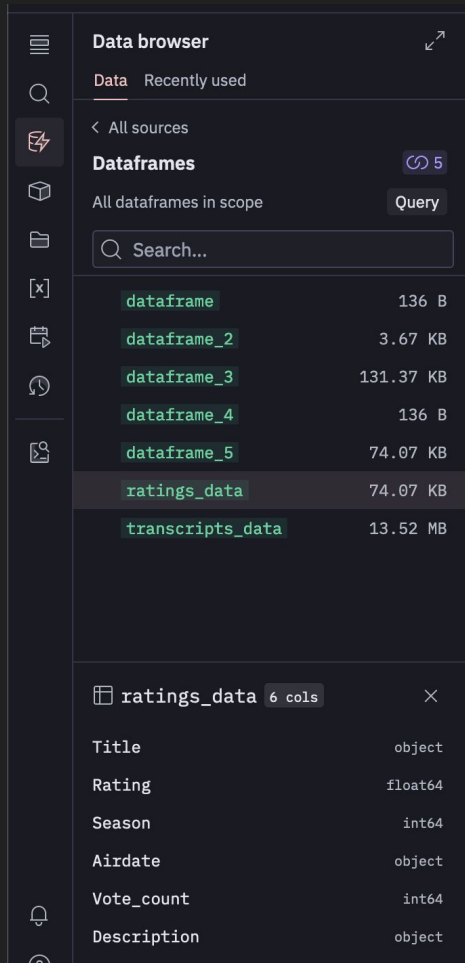
```
1 ratings_data['Title']  
2  
3 ratings_data.iloc[0]
```

# Navigating Hex

How do I know what columns are available to select from?

Go to Data Browser → choose a dataframe

See all the columns listed below



The screenshot shows the Hex Data Browser interface. At the top, there's a 'Data browser' header with a search icon. Below it, there's a 'Data' section with a 'Recently used' filter. The main area shows 'All sources' and 'Dataframes' with a refresh icon and a count of 5. A search bar is present. The list of dataframes includes:

Dataframe Name	Size
dataframe	136 B
dataframe_2	3.67 KB
dataframe_3	131.37 KB
dataframe_4	136 B
dataframe_5	74.07 KB
ratings_data	74.07 KB
transcripts_data	13.52 MB

Below the list, the 'ratings\_data' dataframe is selected, showing its schema with 6 columns:

Column Name	Column Type
Title	object
Rating	float64
Season	int64
Airdate	object
Vote_count	int64
Description	object

# Activity

Create some python cells to test out how things work! After you type it out, hit Command-Enter to run the code and see the results.

Try some of these lines of code to see how it works:

```
1 2+2
```

```
1 ['apples', 'peanut butter'] + ['chocolate chips']
```

```
1 "cat" + "dog"
```

```
1 ratings_data['Rating']*10
```

# Data types

Those were examples of different types of data:

- String
- Number
- List (can be list of strings, integers, etc.)
- Dataframe

Now sometimes we might want to save a result, to refer to later. To do that, we save it to a **variable**.

# Variables

Places to save info, to refer to it later. Think of these as post-it notes – you can have as many as you want, you can give each one a name, and you can store anything in them to refer back to later.

Example:

Code 20

```
1 favorite_snacks = ['apples', 'nutella'] + ['gummy worms']
```

Code 21

```
1 favorite_snacks
```

# Saving info to a dataframe

Similar to creating a variable, you can create a new column in any dataframe to save any results or calculations.

Example:

```
ratings_data['new_score'] = ratings_data['Rating']*10
```

Then go look in your data browser (left side) under ratings\_data!



# Activity

1. In another cell, create a new column in the `ratings_data` dataframe that multiplies the rating by 100.

# Summarizing dataframes

Some of the most powerful formulas for data analysis involve summarizing dataframes.

```
ratings_data['Rating'].mean()
```

```
ratings_data.groupby(['Season']).mean()
```

# Activity

1. Find the average number of vote counts in the ratings table
2. Find the average rating by season

# Summarizing dataframes

Other options for summarizing after `groupby()`:

- `.mean()`
- `.size()` — total number of rows
- `.count()` — total number of rows with non-null values
- `.first()` — first row of that group
- `.last()` — last row of that group
- `.get_group("name of group")` – filter by rows with that value
- `.size().sort_values(ascending=False)` – get the number of rows and sort

# Activity

1. How many medals did each country win?

# Writing code to answer a question

What are the most commonly used words in Seinfeld episode titles?

Note that if we have other similar questions, like:

What about the most commonly used words in the descriptions?

- This will be a good use case for a **function!**

# How to start writing code

Programming is fundamentally **structured thinking**. Here's how you should approach each bit of code you write:

- What is your question?
- What is your desired output?
- What is your input?
- What is the first piece of info you need to answer the question?
- Is there an existing function that can help get you to your desired output?

# EXAMPLE: How to start writing code

*What is your question?*

- What are the most commonly used words in Seinfeld episode titles?

*What is your desired output?*

- A list of words, in order of most-common to least-common

*What is your input?*

- A dataframe of episode information

*What is the first piece of info you need to answer the question?*

- Specifically the titles from my dataset

*Is there an existing function that can help get you to your desired output?*

- Great moment for a quick Google



# Code to answer a question, continued

First bit of info:

```
1 ratings_data['Title']
```

Googled around for a function:

```
1 from collections import Counter  
2 Counter(ratings_data['Title']).most_common()
```

# Text as data

```
episode_words = ratings_data['Description'].str.split().explode().reset_index()
```

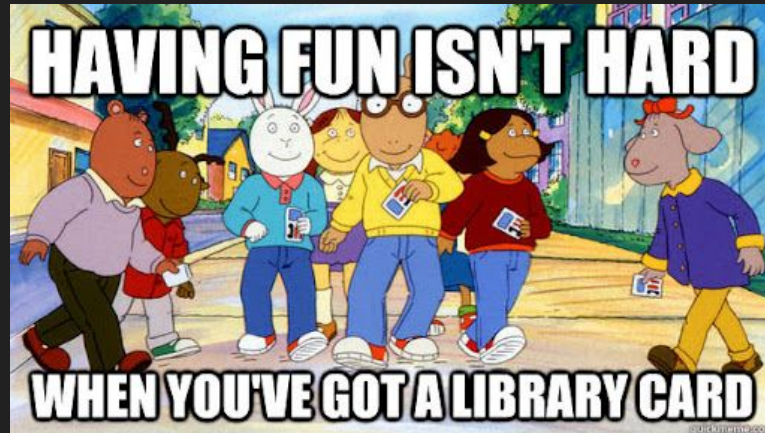
# Activity

What are the most common words in the athletes' philosophy statements?

# A word on Libraries and Objects

When you look for existing functions, those are gonna be in libraries. You need to import them in your project just once, on any line or cell before you use it:

```
1 from collections import Counter
```



# A word on Libraries and Objects

Figure out if it's a function, or a method:

```
2 Counter(ratings_data['Title']).most_common()
```

Some functions return an object, that has functions of its own. These functions are called **methods**. They work the same as functions, but instead of just their name, you call them by `Object.method_name(arguments_if_any)`.

You can check the type of anything by wrapping it in `type()` like this:

```
type(ratings_data)
type(['a', 'b'])
type(Counter(ratings_data['Title']))
```

# Code to answer a question, continued

Let's walk through this step by step:

```
1  from collections import Counter
2  Counter(ratings_data['Title']).most_common()
3
4  all_words = []
5
6  for title in ratings_data['Title']:
7      for word in title.split():
8          all_words.append(word)
9
10 print(all_words)
11
12 Counter(all_words).most_common()
13
```

# Let's make a word cloud!

```
1 from wordcloud import WordCloud
2 import matplotlib.pyplot as plt
3
4 # Generate a word cloud
5 wordcloud = WordCloud(width=800, height=800, background_color="white").generate(
6     " ".join(all_words)
7 )
8
9 # Plot the WordCloud image
10 plt.figure(figsize=(8, 8), facecolor=None)
11 plt.imshow(wordcloud)
12 plt.axis("off")
13 plt.tight_layout(pad=0)
14
15 plt.show()
```



# Activity

Let's make a word cloud of the athletes' philosophy statements!



# When to use your toolkit

You might be wondering, when should you use SQL or Python?

- In general: start with the simplest tool that does the job
- This might be Excel or Google Sheets!
- When you start working with large datasets, or manipulating a lot of data, or repeating your tasks, try it in SQL
- When you're trying to do something more complicated or customized than SQL can handle, a good time to try it in python
- And with Hex, very easy to switch between SQL and python in each step!  
We'll talk about that more next week

# Keep learning!

## Upcoming sessions:

- Python Part 2: Functions
- Data Analytics with SQL & Python

## More resources:

- [learn.hex.tech](https://learn.hex.tech)
- Lots of python tutorials online – with Hex, you can skip any setup that requires you to install anything to your computer or use Jupyter, and just type the commands they're showing you into a Hex cell directly