# MSc Web and Social Media Analytics

## Tutorial Pack 9
## (90 minutes)

(To be completed during LW9 tutorial)

**LEARNING OBJECTIVES**

- **To revise core Python concepts and practice reading Python statements.**
- **To develop a Python program that communicates with the Reddit API.**

**LEARNING OUTCOMES**

- **By the end of this tutorial students will have;**
  - **Revised key Python programming concepts**
  - **Familiarised themselves with object-orientated programming principles**
  - **Setup a Reddit developer account and obtained access keys.**
  - **Practiced collecting data using the Reddit API**
  - **Practiced using external Python packages (Pandas)**

**RESOUCRES AND TOOLS REQUIRED**

- **Python 3 via google collab or repl.it**

**IMPORTANT:**

This pack is designed for you to go at your own speed. At the end of each section there is a series of practice questions and exercises. You should attempt to answer **all questions**.

Any questions you do not complete today should be completed before your next tutorial.

You may find this tutorial pack and the exercises useful preparation for the coursework.

## 1. Introduction

In LW8 we covered the basics of Python programming. We will begin with a revision of last week's material and a review of some important Python concepts.

## 2. Revision of Python concepts (30mins)

Using your understanding of how Python programs work answer the following questions. You should consult your notes from last week's tutorial if you need help.

| QNum | Question |
|---|---|
| 1 | What will be displayed when the following Python code is executed?<br>**print("windy and overcast weather".title())** |
| | Answer |
| | |
| | |
| QNum | Question |
| 2 | What will be displayed when the following Python code is executed?<br>**x = 2**<br>**y = 3**<br>**print(x**y)** |
| | Answer |
| | |
| | |
| QNum | Question |
| 3 | What will be displayed when the following Python code is executed?<br>**i = 10**<br>**if i > 5:**<br>**  print(I*2)**<br>**else:**<br>**  print(i*3)** |
| | Answer |
| | |

| | |
|---|---|
| QNum | Question |
| 4 | What will be displayed when the following Python code is executed?<br>**k = [1,2,3,4,5]**<br>**for i in k:**<br>**  print(i + i)** |
| | Answer |
| | |

| | |
|---|---|
| QNum | Question |
| 5 | What will be displayed when the following Python code is executed?<br>**tweet = "Today is a new day!"**<br>**print(tweet[-4])** |
| | Answer |
| | |

| | |
|---|---|
| QNum | Question |
| 6 | What will be displayed when the following Python code is executed?<br>**tweet = "Today is a new day!"**<br>**print(tweet.split()[3])** |
| | Answer |
| | |

| | |
|---|---|
| QNum | Question |
| 7 | Write a python statement to print the number of characters in the following tweet<br>**tweet = "I've just watched the film Gravity…What's up with that?**<br>**At one point, Scarlett Johansson was in the running to star in this film, as was Naomi Watts"** |
| | Answer |
| | |

| QNum | Question |
|---|---|
| 8 | Explain the purpose of the following program. What will be the output when it is run? |
| | **def stopwords(text):**<br>**found = []**<br>**for word in text.split():**<br>**if word in [“the”, “from”,”to”,”at”,”they”, “them”]:**<br>**if word not in found:**<br>**found.append(word)**<br>**return found**<br><br>**print(stopwords(“Make a donation, help save lives around the world”))** |
| | Answer |
| | |

| QNum | Question |
|---|---|
| 9 | Why does the following program fail to run? What changes should you make to fix it? |
| | **class Tweet:**<br>**def __init__(self, author, text, post_date):**<br>**self._author = author**<br>**self._text = text**<br>**self._post_date = post_date**<br><br>**user_tweet = Tweet()** |
| | Answer |
| | |

| QNum | Question |
|---|---|
| 10 | Consider the following Python statements. Using the “all_users” variable **ONLY**, print out the surname of the 2nd user. |
| | **user_info = [{“name”: “macy”, “surname”: “gray”},{“name”: “john”, “surname”: “smith”}]**<br>**all_users = {“users”: user_info, “posts”: []}** |
| | Answer |
| | |

# 3. The Reddit API

## Setting up a developer account

Reddit is a social news and online discussion platform. Reddit consists of thousands of different subreddits. Each subreddit corresponds to a community based around a particular topic or theme. Social media data posted to Reddit can be accessed through the Reddit API.

*Access to the Reddit API is achieved in **THREE** steps.*

1. Sign up for a normal Reddit account and login.
2. Read the API terms and register your account (https://www.reddit.com/wiki/api/)
3. Create an application by completing the form (https://www.reddit.com/prefs/apps) NB. this will generate your secret key and APP_ID. Keep these for later.


*Figure 1 The create application form*

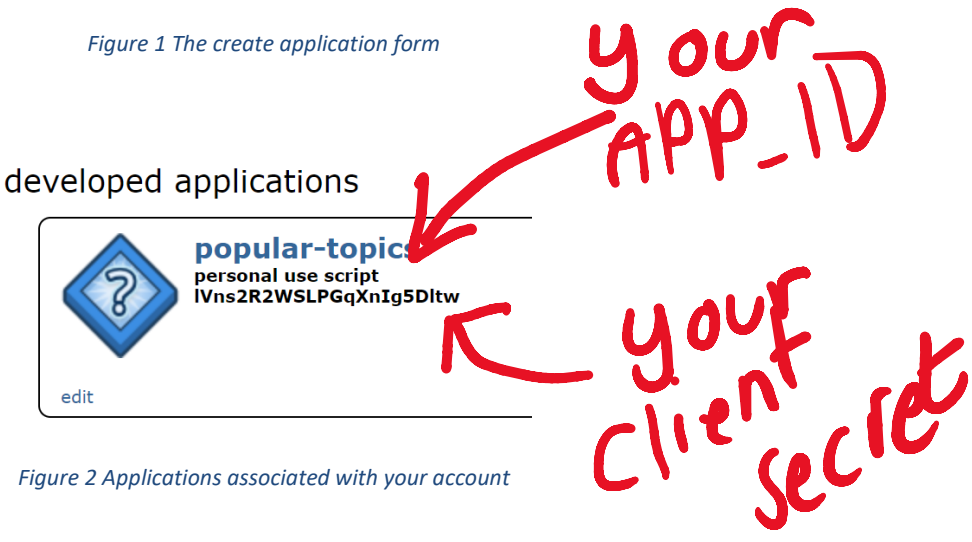



*Figure 2 Applications associated with your account*

## The Python Reddit API Wrapper (PRAW)

PRAW is an external Python package that simplifies working with the Reddit API in Python applications.

To use external packages in Python they should first be installed. In google collab they can be added to your environment by using the pip command.

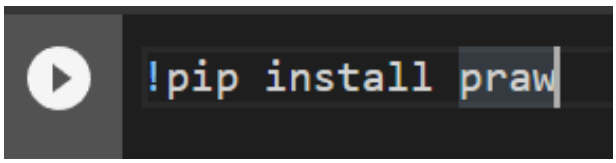



*Figure 3 Installing the PRAW package in Google Collab*

To use PRAW we first initialise a new praw.Reddit object. This object needs to be initialised with your own APP_ID and secret phrase.

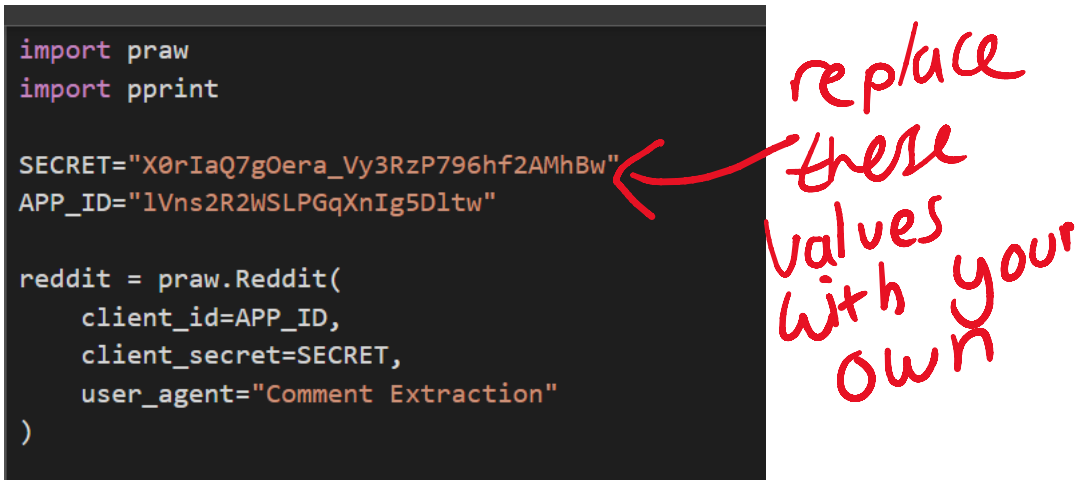



*Figure 4 Example setting up the praw.Reddit object*

Next, we create a new submission object to represent the post we would like to extract comments from. The submission object requires the full URI of a post on Reddit.

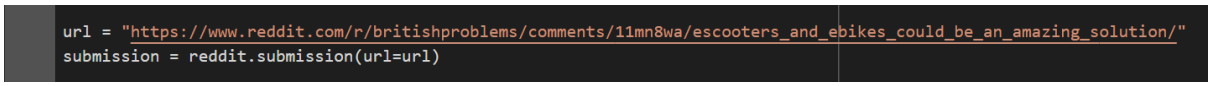



*Figure 5 Setting up the submission object.*

To access the top-level comments associated with a submission, we access its comments property.

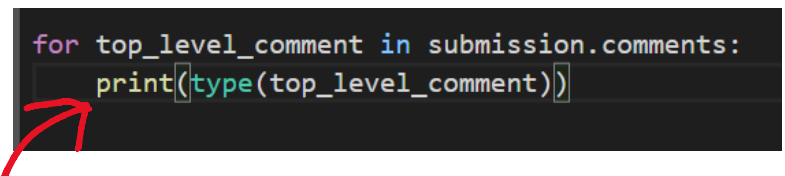



*Figure 6 Using a loop to access top-level comments.*

here we are only printing the objects type

When running the above program you should observe that the object type of each comment is itself of type `<class 'praw.models.reddit.comment.Comment'>.`

By reviewing the documentation for the PRAW package we can identify all the properties associated with an individual Comment object. For example, comment.body would give us the text associated with a comment (in Markdown format)

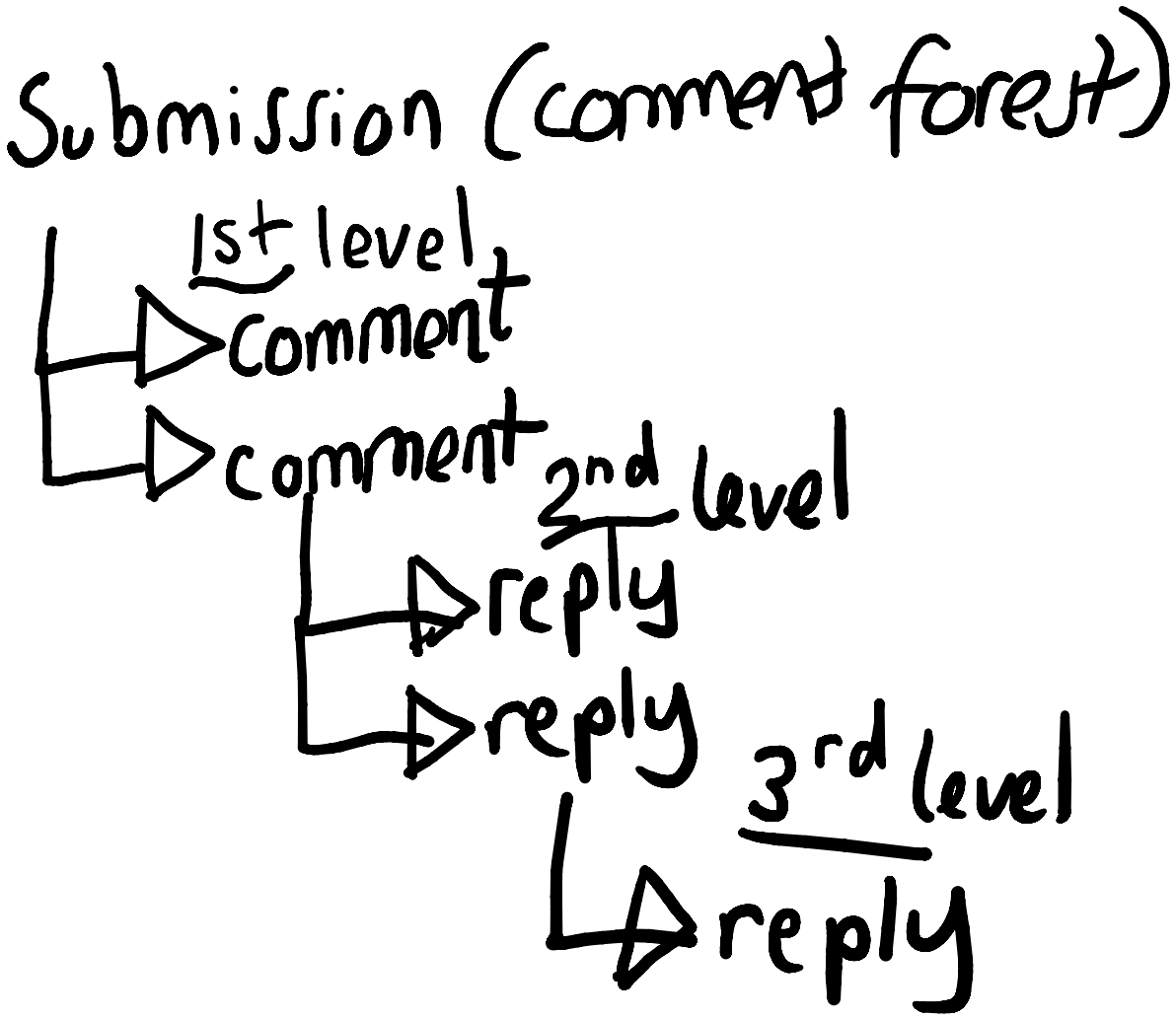
| Attribute | Description |
|---|---|
| `author` | Provides an instance of `Redditor`. |
| `body` | The body of the comment, as Markdown. |
| `body_html` | The body of the comment, as HTML. |
| `created_utc` | Time the comment was created, represented in Unix Time. |
| `distinguished` | Whether or not the comment is distinguished. |
| `edited` | Whether or not the comment has been edited. |
| `id` | The ID of the comment. |
| `is_submitter` | Whether or not the comment author is also the author of the submission. |
| `link_id` | The submission ID that the comment belongs to. |
| `parent_id` | The ID of the parent comment (prefixed with `t1_`). If it is a top-level comment, this returns the submission ID instead (prefixed with `t3_`). |
| `permalink` | A permalink for the comment. `Comment` objects from the inbox have a `context` attribute instead. |
| `replies` | Provides an instance of `CommentForest`. |
| `saved` | Whether or not the comment is saved. |
| `score` | The number of upvotes for the comment. |
| `stickied` | Whether or not the comment is stickied. |
| `submission` | Provides an instance of `Submission`. The submission that the comment belongs to. |
| `subreddit` | Provides an instance of `Subreddit`. The subreddit that the comment belongs to. |
| `subreddit_id` | The subreddit ID that the comment belongs to. |

*Figure 7 Attributes associated with each Comment instance.*

By default, accessing the comments property will provide us with access to all the 1st level comments associated with a submission.

In practice there may be more than one level of comments, given the ability of users to response to each other. 2nd level comments are those that are posted in response to 1st level comments. 3rd level comments are those that were posted in response to 2nd level comments and so forth.

Together these comments resemble a tree-like structure (known as a CommentForest in PRAW)




To access comments further down the tree, we need to loop over the replies of comments in proceeding levels. This is illustrated in the following code, which walks down the tree to the 3rd level.

```
for top_level_comment in submission.comments:
  for second_level_comment in top_level_comment.replies:
    for third_level_comment in second_level_comment.replies:
      print(type(third_level_comment))
```

*Figure 8 Accessing third level comments in the CommentForest*

this is an attribute of a comment

Alternatively, we can access the list function of the comments attribute associated with each submission to loop over all comments attached to a submission (regardless of their level in the comment forest).

Keep in mind that since this approach makes many calls to the Reddit API you should be careful not abuse this feature or risk your account being blocked.

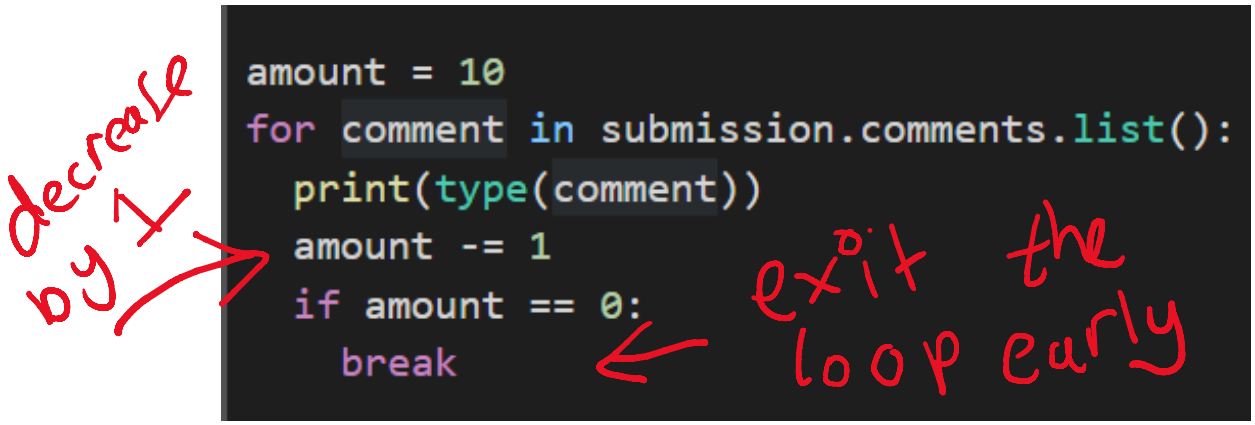



*Figure 9 Collapsing the Comment Forest*

Attributes of a comment may themselves be other objects. For example, the author associated with each comment is returned as a `<class 'praw.models.reddit.redditor.Redditor'>`. This can be verified using the code below.

```
for comment in submission.comments.list():
  print(type(comment.author))
  break
```

*Figure 10 Accessing the author property.*

The author's name is a property of the Redditor object. The Redditor object is itself attached as a property to each comment.

The following code demonstrates how to print out the author names of 10 comments attached to a submission. The **enumerate** function is used to support the loop by returning a counter and the next item from the list each time the loop is run.

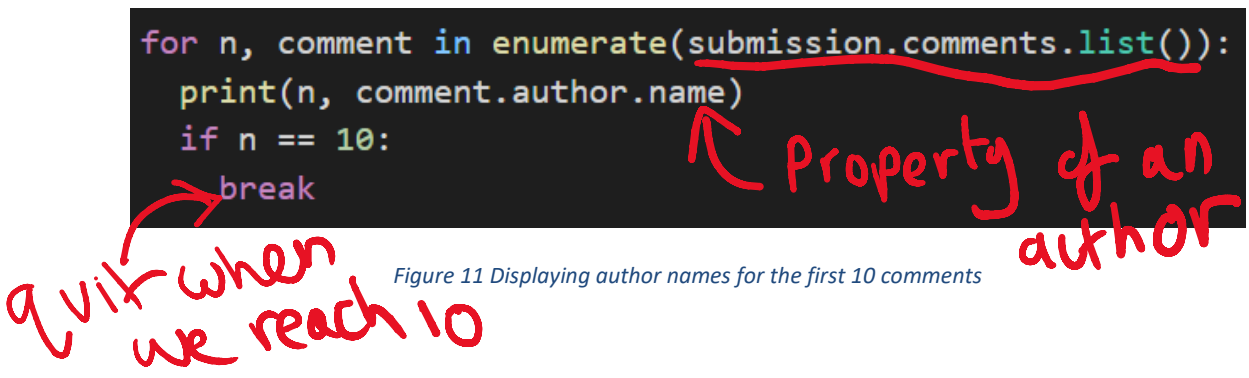



*Figure 11 Displaying author names for the first 10 comments*

| Attribute | Description |
| --- | --- |
| `comment_karma` | The comment karma for the `Redditor`. |
| `comments` | Provide an instance of `SubListing` for comment access. |
| `submissions` | Provide an instance of `SubListing` for submission access. |
| `created_utc` | Time the account was created, represented in Unix Time. |
| `has_verified_email` | Whether or not the `Redditor` has verified their email. |
| `icon_img` | The url of the Redditors' avatar. |
| `id` | The ID of the `Redditor`. |
| `is_employee` | Whether or not the `Redditor` is a Reddit employee. |
| `is_friend` | Whether or not the `Redditor` is friends with the authenticated user. |
| `is_mod` | Whether or not the `Redditor` mods any subreddits. |
| `is_gold` | Whether or not the `Redditor` has active Reddit Premium status. |
| `is_suspended` | Whether or not the `Redditor` is currently suspended. |
| `link_karma` | The link karma for the `Redditor`. |
| `name` | The Redditor's username. |
| `subreddit` | If the `Redditor` has created a user-subreddit, provides a dictionary of additional attributes. See below. |
| `subreddit["banner_img"]` | The URL of the user-subreddit banner. |
| `subreddit["name"]` | The fullname of the user-subreddit. |
| `subreddit["over_18"]` | Whether or not the user-subreddit is NSFW. |
| `subreddit["public_description"]` | The public description of the user-subreddit. |
| `subreddit["subscribers"]` | The number of users subscribed to the user-subreddit. |
| `subreddit["title"]` | The title of the user-subreddit. |

*Figure 12 Properties of the Redditor object type*

## 4. Pandas

Pandas is an extremely powerful data manipulation and analysis framework for Python. A core object type in Pandas is the pandas.DataFrame.

A **data frame** is roughly equivalent to a Microsoft Excel worksheet, in the sense that it stores a series of data instances (rows) across a set of fields (columns).

Each **row** in a data frame is associated with a unique identifier, this could be the row number or a unique field present in the dataset. In pandas this unique identifier is known as the **index**.

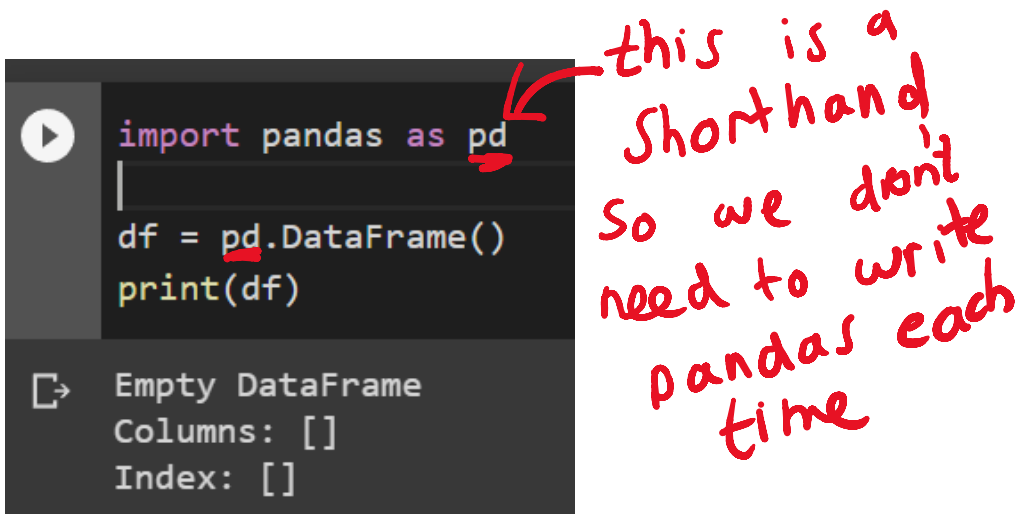



*Figure 13 Creating an empty data frame in pandas*

Data frames can be created in various ways. One way is to populate a data frame from a list of dictionaries.

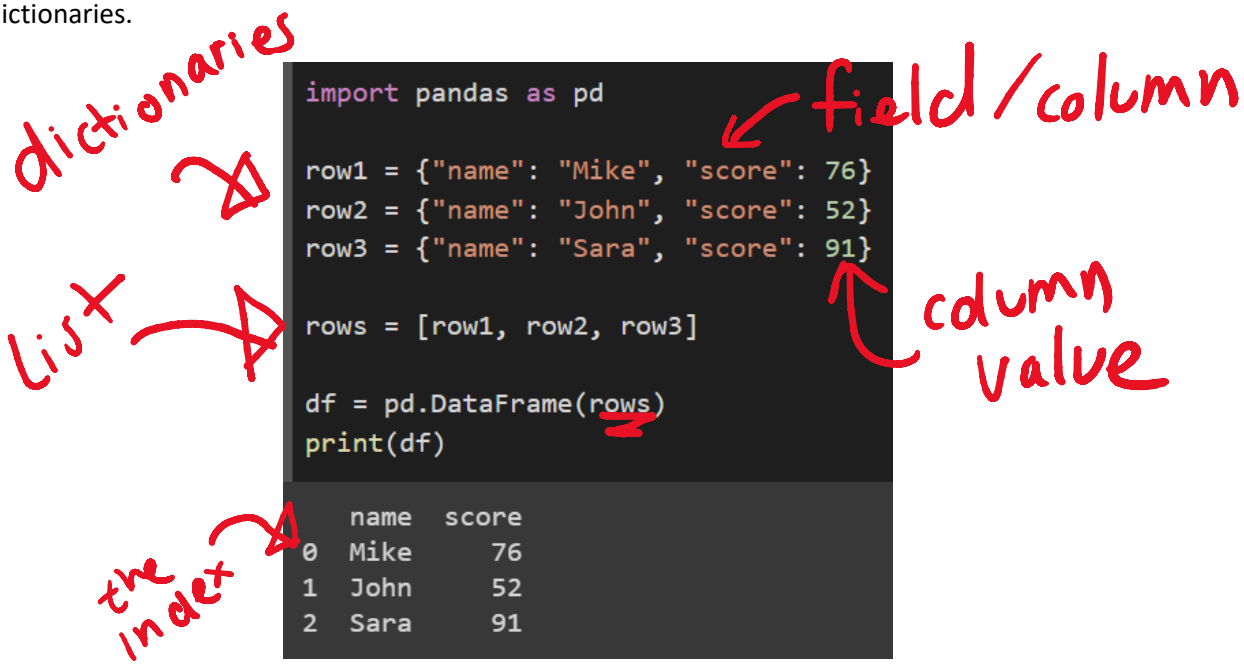



*Figure 14 Creating a data frame from a list of dictionaries.*

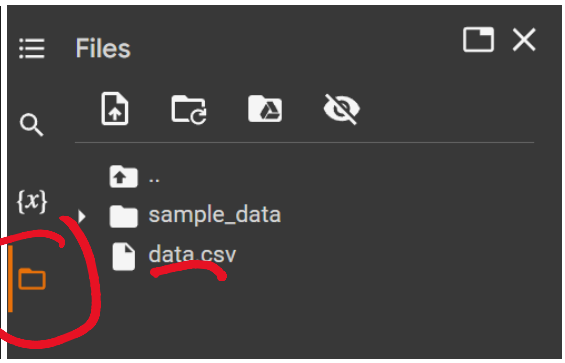
Data frames can be saved to CSV (comma separated values) files by calling the to_csv method. Similarly, they can also be created by using the pandas.read_csv(*filename*) function.

```
import pandas as pd

row1 = {"name": "Mike", "score": 76}
row2 = {"name": "John", "score": 52}
row3 = {"name": "Sara", "score": 91}

rows = [row1, row2, row3]

df = pd.DataFrame(rows)
df.to_csv("data.csv")
```




Individual columns or sets of columns can be accessed using array notation and specifying the column name(s).

```
df = pd.DataFrame(rows)
df["name"]
```

```
0    Mike
1    John
2    Sara
Name: name, dtype: object
```

*Figure 15 Selecting a single column from a data frame.*

Various statistics can be computed on a column containing numeric data, e.g. min(), max(), mean(). std(), var(). The describe method returns a set of summary statistics.

```
df["score"].describe()
```

```
count     3.000000
mean     73.000000
std      19.672316
min      52.000000
25%      64.000000
50%      76.000000
75%      83.500000
max      91.000000
Name: score, dtype: float64
```

Adding a new column to a data frame can be done by using array notation and specifying both the name of the new column and its associated value.

```
df = pd.DataFrame(rows)
df["year_of_study"] = 2013
df
```

| | name | score | year_of_study |
|---|---|---|---|
| 0 | Mike | 76 | 2013 |
| 1 | John | 52 | 2013 |
| 2 | Sara | 91 | 2013 |

*Figure 16 Adding a new column.*

If we want to set the value of a new column according to the values of other fields in the row we can use the apply method. The apply method requires you to specify a function that returns the value of a column given a specific row.

```
df = pd.DataFrame(rows)
df["year_of_study"] = 2023

def score_to_grade(row):
  if row.score >= 70:
    return "distinction"
  if row.score >= 60:
    return "merit"
  if row.score >= 50:
    return "pass"
  return "fail"

df["grade"] = df.apply(score_to_grade, axis=1)
df
```

| | name | score | year_of_study | grade |
|---|---|---|---|---|
| 0 | Mike | 76 | 2023 | distinction |
| 1 | John | 52 | 2023 | pass |
| 2 | Sara | 91 | 2023 | distinction |

the new field name

the function that will determine the value of a field for each row

*Figure 17 Calculating a new column value based on the value of other columns in the row*

# *Exercises*

Using your understanding of the Reddit API and Pandas data frames, attempt the following exercises. It is important that you add comments to your code as you go along.

Q1. **Create a Reddit developer account if you haven't already done so**.

Select a reddit submission in the reddit.com/r/news subreddit with at least 50 comments. Note the full URI to this submission. Develop a Python program that prints out the top-level comments for this submission to the console.

Q2. Modify your program so that your program collects 50 comments across *any* level of the Comment Forest. For each comment you should print the body of the comment, the author's name, and the length of the comment in words.

Q3. Modify the program you developed in Q2 so that the comments collected are stored in a data frame. Save this data frame to a CSV file called "comments-dataset.csv". Add a new column to your data frame that is based on the length (in characters) of each comment. Use this new column to determine the min, max and average length of a comment posted.