

Class Objectives

By the end of today's class you will be able to:



Load JSON data into pandas DataFrames



Use 'try' and 'except' to resolve missing key values without terminating the code



Use linear regression to predict temperature at certain latitudes



Activity: JSON Traversal

In this activity, you will be traversing a JSON file using your knowledge of Python.



JSON Traversal Instructions

- Load the provided JSON.
- Retrieve the video's title.
- Retrieve the video's rating.
- Retrieve the link to the video's thumbnail.
- Retrieve the number of views this video has.

```
{"apiVersion": "2.0",
 "data":{
    "updated": "2010-01-07T19:58:42.949Z",
    "totalItems":800.
    "startIndex":1,
    "itemsPerPage":1,
    "items":
        {"id": "hYB0mn5zh2c",
         "uploaded": "2007-06-05T22:07:03.000Z",
         "updated": "2010-01-07T13:26:50.000Z",
         "uploader": "GoogleDeveloperDay",
         "category": "News",
         "title": "Google Developers Day US - Maps API Introduction",
         "description": "Google Maps API Introduction ...",
         "tags":[
            "GDD07", "GDD07US", "Maps"
         "thumbnail":{
            "default": "http://i.ytimg.com/vi/hYB0mn5zh2c/default.jpg",
            "hqDefault": "http://i.ytimg.com/vi/hYB0mn5zh2c/hqdefault.jpg
         "player":{
            "default": "http://www.youtube.com/watch?vu003dhYB0mn5zh2c"
```





Activity: Requests Review

In this activity, you will be making an API call to a NYT scraper and printing out data from the response.



Request Review Instructions

- Make a request to the following endpoint
 (http://nyt-mongo-scraper.herokuapp.com/api/headlines), and store the response.
- Parse the response JSON
- Print the JSON representations of the first and last posts.
- Print number of posts received.





Instructor Demonstration OpenWeatherMap API





Activity: Weather in Burundi

In this activity, you will work with the OpenWeather API and create an application which provides the user with the current temperature in the largest city of Burundi.



Weather in Burundi Instructions

- Gather all of your "config" information -- your API key and the base URL.—before starting.
- Build your query URL. Check the documentation to figure out how to request temperatures in Celsius.
- Make your request, and save the API response.
- Retrieve the current temperature in Bujumbura from the JSON response.
- Print the temperature to the console.
- **Bonus:** Augment your code to GET the temperature in both Fahrenheit _and_ Celsius.
- Note: Don't forget to change the API key in config.py!





Instructor Demonstration OpenWeatherMap DataFrame

OpenWeatherMap DataFrame

- Using our previous OpenWeatherMap API requests
- The API response contains fields such as temperature and latitude.
- A for loop is used to loop through the cities list, make a request and append to a list.
- What would be an easy way to analyze the different metrics?

```
cities = ["Paris","London","Oslo","Beijing"]

# set up lists to hold reponse info
lat = []
temp = []

# Loop through the list of cities and perform a request for data on each
for city in cities:
    response = requests.get(query_url + city).json()
    lat.append(response['coord']['lat'])
    temp.append(response['main']['temp'])

print(f"The latitude information received is: {lat}")
print(f"The temperature information received is: {temp}")

The latitude information received is: [48.86, 51.51, 59.91, 39.91]
The temperature information received is: [8.59, 6, 0, 1]
```

OpenWeatherMap DataFrame cont.

- Once all the data is collected the list can be stored into a dictionary then into a DataFrame.
- With the data now in a DataFrame it can be plotted with MatplotLib

```
# create a data frame from cities, lat, and temp
weather_dict = {
    "city": cities,
    "lat": lat,
    "temp": temp
}
weather_data = pd.DataFrame(weather_dict)
weather_data.head()
```

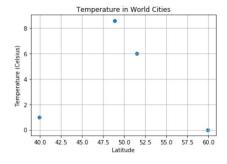
	city	lat	temp
0	Paris	48.86	8.59
1	London	51.51	6.00
2	Oslo	59.91	0.00
3	Beijing	39.91	1.00

```
# Build a scatter plot for each data type
plt.scatter(weather_data["lat"], weather_data["temp"], marker="o")

# Incorporate the other graph properties
plt.title("Temperature in World Cities")
plt.ylabel("Temperature (Celsius)")
plt.xlabel("Latitude")
plt.grid(True)

# Save the figure
plt.savefig("TemperatureInWorldCities.png")

# Show plot
plt.show()
```







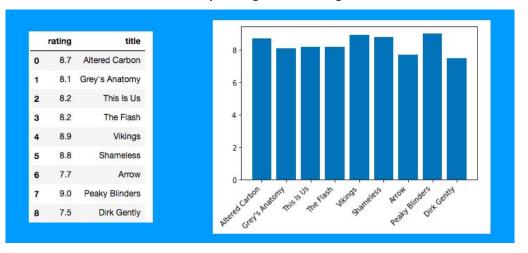
Activity: TV Ratings

In this activity, you will take some time to create an application that reads in a list of TV shows, makes multiple requests from an API to retrieve rating information, creates a pandas dataframe, and visually displays the data.

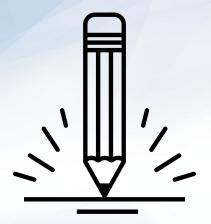


TV Ratings Instructions

- You may use the list of TV shows provided in the starter file or create your own.
- Request information on each TV show from https://www.tvmaze.com/api#show-search
- Store the name and rating information into lists.
- Store this data in a dictionary and use it to create a Pandas DataFrame.
- Use matplotlib to create a bar chart comparing the ratings of each show.







Activity: Weather Statistics

In this activity, you will generate a regression model on a dataset from the Open Weather API to predict the temperature of a city.



Weather Statistics Instructions

- Using the starter file as a guide complete the following:
 - Create a scatter plot of Temperature vs. Latitude.
 - Perform linear regression.
 - Create a line equation for the regression.
 - Create a scatter plot with the linear regression line.
 - Predict the temperature of Florence at latitude 34.8
 - Use the API to determine the actual temperature of Florence.
- If you finish early feel free to try and predict the temperature at other cities.
- **HINT:** if you need help be sure to revisit your stats material from 5.3.



Take a Break!





Instructor Demonstration Exception Handling



What would happen if an application tried to look up a key within a dictionary and that key was missing?

Errors

```
students = {
    # Name : Age
    "James": 27,
    "Sarah": 19,
    "Jocelyn": 28
print(students["Jezebel"])
print("This line will never print.")
KeyError
                                           Traceback (most recent call last)
<ipython-input-1-4692324b5d88> in <module>()
      6 }
---> 8 print(students["Jezebel"])
     10 print("This line will never print.")
KeyError: 'Jezebel'
```

- So far our requests to our APIs have had the values we are looking for.
- When a value is not found Python returns an error.
- As we can see in the notebook when `students` dictionary does not have a key for "Jezebel".

Try/Except

- The try/except code will let an application recover from errors like the one previously.
- Try and except are keywords, like for and if.
- Python will "try" to run the code.
- If the code throws an error or exception,
 the code in the except block in executed

```
students = {
    # Name : Age
    "James": 27,
    "Sarah": 19,
    "Jocelyn": 28
}

# Try to access key that doesn't exist
try:
    students["Jezebel"]
except KeyError:
    print("Oops, that key doesn't exist.")

# "Catching" the error lets the rest of our code execute
print("...But the program doesn't die early!")
```

Oops, that key doesn't exist.
...But the program doesn't die early!





Activity: Making Exceptions

In this activity, you will create an application that, through 'try' and 'except', resolves a number of errors.



Making Exceptions Instructions

Without removing any of the lines from the starter code provided, create 'try' and 'except' blocks
that will allow the application to run without terminating.





Activity: API Call Exceptions

In this activity, you will implement **try/except** as you make API calls to narrow down a list of fictional characters to include only characters from Star Wars.



Making Exceptions Instructions

- Loop through the characters in the list and send a request to the Star Wars API.
- Create a try clause to append the height and mass for characters available in the Star Wars API and an except clause to append None for those that do not.
- Create a DataFrame from the results.
- Drop any rows with null values to remove characters not in the Star Wars universe.

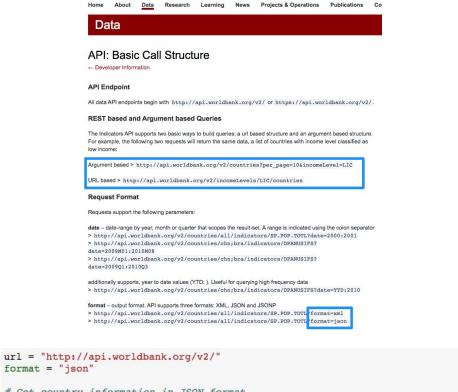




Instructor Demonstration World Bank API

World Bank API

- Up to now we have been working with fairly straightforward API queries
- There are more complicated API frameworks that exist
- For the remainder of class we will practice working with more complicated APIs

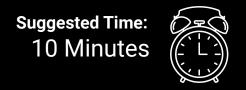






Activity: Two Calls

In this activity, you will be utilizing the World Bank API to make two API calls in a sequence. The second API call depends on the response of the first.



Two Calls Instructions

- Retrieve a list of the lending types the world bank keeps track of, and extract the ID key from each
 of them.
- Next, determine how many countries are categorized under each lending type. Use a dict to store this information.
 - This data is stored as the first element of the response array.
- Finally, print the number of countries of each lending type.

```
The number of countries with lending type IBD is 69. The number of countries with lending type IDB is 16. The number of countries with lending type IDX is 59. The number of countries with lending type LNX is 74.
```





One more thing!

Please have a credit card handy on Saturday!

We will be signing up for the Google Places API. While there is **NO COST** it does require a credit card.