Essential Python for Data Analyst Live Class

Tags	Live Class Python
Class	
Finished Yet?	☑
Knowledge	Live Classes

เนื้อหาสำคัญที่จะเรียนใน Class นี้:

- Loop over list/dictionary
- Try except block
- csv, json
- Context Manager
- Requests, API
- Basic web scraping
- Intro to numpy, pandas, sklearn

-ใช้ Google Colab เหมือนคลาสก่อนหน้า

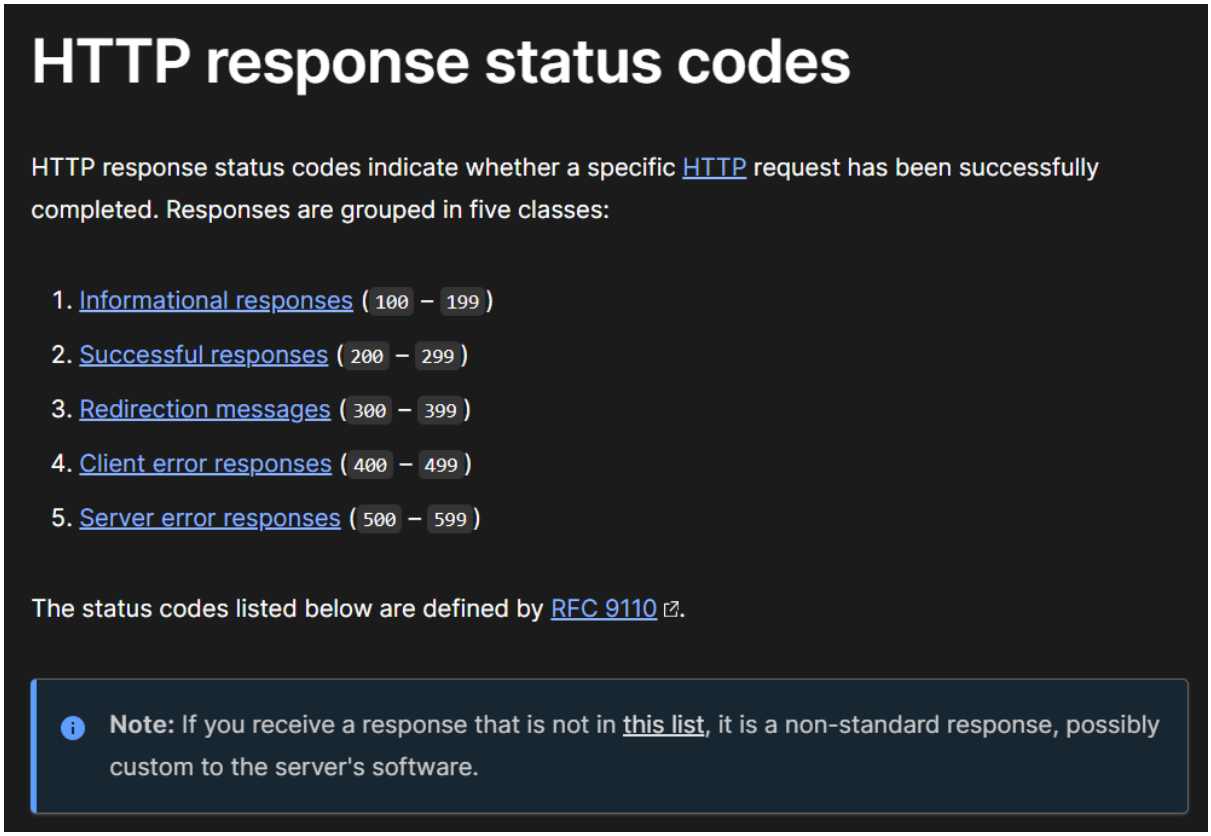
My Google Colab Link: https://colab.research.google.com/drive/1Ic6CH4mZk1KkdZKnlIH8JCb9VOJoejCs?usp=sharing

อ่านเรื่อง API เพิ่มเติมได้ที่:

Intro to API (Jitta)

*Get request สำคัญ ต้องทำให้เป็น

HTTP docs: https://developer.mozilla.org/en-US/docs/Web/HTTP



-ใน Excel จะมี Flash Fill (CTRL + E) ซึ่งมีประโยชน์มาก ทำให้ประหยัดเวลาได้เยอะ (Google Sheets ทำไม่ได้)

imdb_anime:

> https://s3-us-west-2.amazonaws.com/secure.notion-static.com/264f330c-022e-4bf6-a045-13c5a6c9b093/imdb_anime.csv

อ่านเรื่อง Pandas เพิ่มเติมได้ที่:

1. 🐼 Intro to Pandas
2. 🐼 Intermediate Pandas 1
3. 🐼 Intermediate Pandas 2

Homework

```
# HW 01: Find public API and create a dataframe
import pandas as pd
import requests as re

url = "https://animechan.vercel.app/api/quotes/anime?title=naruto"

headers = {"Accept-Language": "en-US"}

response = re.get(url, headers = headers)

data_json = response.json()
df = pd.DataFrame(data_json, columns = ['anime', 'character', 'quote'])
```

```
df.to_csv("Naruto_Quotes.csv")
```

```
# HW 02: Find 3 models in sklearn and test them with small dataset
import pandas as pd
import numpy as np
from xgboost.sklearn import XGBRegressor
from sklearn.kernel_ridge import KernelRidge
from sklearn.linear_model import BayesianRidge
from sklearn.model_selection import train_test_split

mtcars = pd.read_csv("https://gist.githubusercontent.com/seankross/a412dfbd88b3db70b74b/raw/5f23f993cd87c283ce766e7ac6b329ee7cc2e1d1/m

mtcars.head()

# Prepare data
# mpg = f(hp, wt, qsec)
X = mtcars[ ["hp", "wt", "qsec"] ]
y = mtcars["mpg"]

# Split data
# Train = 80%, Test = 20%
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)

# Model: XGBoost
model = XGBRegressor().fit(X_train, y_train)

# Test model [scoring]
pred = model.predict(X_test)

# Evaluate model
# MAE (Mean Absolute Error)
mae_xgb = np.mean(np.absolute((y_test - pred)))
# MSE (Mean Squared Error)
mse_xgb = np.mean((y_test - pred)**2)

# Model: Kernel Ridge
model = KernelRidge().fit(X_train, y_train)

# Test model [scoring]
pred = model.predict(X_test)

# Evaluate model
# MAE (Mean Absolute Error)
mae_kr = np.mean(np.absolute((y_test - pred)))
# MSE (Mean Squared Error)
mse_kr = np.mean((y_test - pred)**2)

# Model: Bayesian Ridge
model = BayesianRidge().fit(X_train, y_train)

# Test model [scoring]
pred = model.predict(X_test)

# Evaluate model
# MAE (Mean Absolute Error)
mae_br = np.mean(np.absolute((y_test - pred)))
# MSE (Mean Squared Error)
mse_br = np.mean((y_test - pred)**2)

model = ["XGBoost", "Kernel Ridge", "Bayesian Ridge"]
mae = [mae_xgb, mae_kr, mae_br]
mse = [mse_xgb, mse_kr, mse_br]

df_mae_mse = pd.DataFrame({
    "Model": model,
    "MAE": mae,
    "MSE": mse
})

df_mae_mse.to_csv("mtcars_mpg_prediction_3_models.csv")
```