# Python for DA

- OOP (Object Oriented Programming
- Read and Write File
- Context Manager
- Handle Error
- Pandas & Numpy (on demand)
- Request API

## OOP (Object Oriented Programming)

การสอนให้ Computer รู้จัก Object

Concept เหมือน Cookie Cutter

```
class Dog: ##ตัวแรก Capital
    def __init__(self, name, species, age) -> None: ##double un
        self.name = name
        self.species = species
        self.age = age
    def bark(self):
        print(f"{self.name} is barking")
    def birthday(self):
        self.age += 1
        print(f"{self.name} is {self.age}")
    pass
```

```
dog1 = Dog("Milo", "Pom", 3)
dog2 = Dog("Milk", "Husky", 4)
```

```
print(f"{dog1.name} is {dog1.species}")
print(f"{dog2.name} is {dog2.species}")
```

```
Milo is Pom
Milk is Husky
```

```
1 dog1.bark()
```

```
Milo is barking
```

```
1 dog1.birthday()
2 dog2.birthday()
```

```
Milo is 4
Milk is 5
```

# องค์ประกอบ OOP

- attributes: name, species, age
- methods: function bark() birthday()

```
1 class ATM:
2     def __init__(self, username, bank, balance):
3         self.username = username
4         self.bank = bank
5         self.balance = balance
6 
7     def __str__(self): ##ใช้ return function print()
8         return f"Hi {self.username}! I'm {self.bank}'s ATM Mach
9 
10    def check_balance(self):
11        print(f"Your balance is {self.balance} THB.")
12    def deposit(self, amount):
13 
14        self.balance += amount
15        print(f"Deposit {amount} THB success.")
```

```
1 tor = ATM("Tor", "K-bank", 100)
```

```
1 print(f"Welcome to {tor.username}'s {tor.bank} account.")
```

```
Welcome to Tor's K-bank account.
```

```
1 tor.check_balance()
```

```
Your balance is 100 THB.
```

```
1 tor.deposit(3000)
```

```
Deposit 3000 THB success.
```

```
1 print(tor)
```

```
Hi! I'm K-bank's ATM Machine.
```

# Super Class

สาย Software ได้ใช้ในการขาย Base Class

สาย DA ไม่ค่อยได้ใช้

```
1 class Person:
2     def __init__(self, name, age):
3         self.name = name
4         self.age = age
5 
6 ## extension / build on top
7 class Employee(Person):
8     def __init__(self, name, age, company):
9         super().__init__(name, age) ##super refer to Person
10        self.company = company
11 
12    def greeting(self):
13        print("Hi!")
```

```
1 person1 = Person("tor", 27)
```

```
1 person2 = Employee("May", 28, "SCB")
```

```
1 person2.greeting()
```

```
Hi!
```

# Read File

## Magic command to run Terminal

```
!pwd #magic command to run Terminal
```

```
/content
```

```
!ls
```

```
hotel.csv  sample_data
```

## Try Except Block

ใช้แก้ปัญหา error เพราะ Python 1 error ทำให้ code ใช้ไม่ได้เลย

```
try:
    result = 1/0
except:
    print("error here!")
```

```
error here!
```

## Read CSV File

```
!cat hotel.csv
```

```
hotel_id,name,city,star_rating,price_per_night
101,Grand Plaza,New York,5,350
102,Ocean View,Miami,4,220
103,Mountain Retreat,Denver,4,185
104,The Urban Stay,Chicago,3,140
105,Sunset Resort,Los Angeles,5,410
106,Desert Inn,Phoenix,3,95
107,Heritage House,Boston,4,210
108,Lakeside Lodge,Seattle,3,130
109,Golden Gate Suites,San Francisco,5,380
110,Central Boutique,Austin,4,195
```

```
## read file
import csv ##ควร import บนสุด / csv เป็น standard python
```

```
try:
    file = open("hotel.csv", "r") ## r=read/ w=write/ a=append

    ## module.function()
    data = csv.reader(file)

    for row in data:
        print(row)

    file.close()
except:
    print("file not found")
finally:
    print("read file success")
```

```
['hotel_id', 'name', 'city', 'star_rating', 'price_per_night']
['101', 'Grand Plaza', 'New York', '5', '350']
['102', 'Ocean View', 'Miami', '4', '220']
['103', 'Mountain Retreat', 'Denver', '4', '185']
['104', 'The Urban Stay', 'Chicago', '3', '140']
['105', 'Sunset Resort', 'Los Angeles', '5', '410']
['106', 'Desert Inn', 'Phoenix', '3', '95']
['107', 'Heritage House', 'Boston', '4', '210']
['108', 'Lakeside Lodge', 'Seattle', '3', '130']
['109', 'Golden Gate Suites', 'San Francisco', '5', '380']
['110', 'Central Boutique', 'Austin', '4', '195']
read file success
```

# Context Manager

ช่วยในการลดขั้นตอน Read/Close File

# Standard python

ใช้ csv. Reader

```
read_data = []

with open("hotel.csv", "r") as file:
    data = csv.reader(file)
    for row in data:
        read_data.append(row)
```

```
1 read_data
```

```
[['hotel_id', 'name', 'city', 'star_rating', 'price_per_night'],
 ['101', 'Grand Plaza', 'New York', '5', '350'],
 ['102', 'Ocean View', 'Miami', '4', '220'],
 ['103', 'Mountain Retreat', 'Denver', '4', '185'],
 ['104', 'The Urban Stay', 'Chicago', '3', '140'],
 ['105', 'Sunset Resort', 'Los Angeles', '5', '410'],
 ['106', 'Desert Inn', 'Phoenix', '3', '95'],
 ['107', 'Heritage House', 'Boston', '4', '210'],
 ['108', 'Lakeside Lodge', 'Seattle', '3', '130'],
 ['109', 'Golden Gate Suites', 'San Francisco', '5', '380'],
 ['110', 'Central Boutique', 'Austin', '4', '195']]
```

```
1 read_data[0]
```

```
['hotel_id', 'name', 'city', 'star_rating', 'price_per_night']
```

```
1 read_data[1:]
```

```
[['101', 'Grand Plaza', 'New York', '5', '350'],
 ['102', 'Ocean View', 'Miami', '4', '220'],
 ['103', 'Mountain Retreat', 'Denver', '4', '185'],
 ['104', 'The Urban Stay', 'Chicago', '3', '140'],
 ['105', 'Sunset Resort', 'Los Angeles', '5', '410'],
 ['106', 'Desert Inn', 'Phoenix', '3', '95'],
 ['107', 'Heritage House', 'Boston', '4', '210'],
 ['108', 'Lakeside Lodge', 'Seattle', '3', '130'],
 ['109', 'Golden Gate Suites', 'San Francisco', '5', '380'],
 ['110', 'Central Boutique', 'Austin', '4', '195']]
```

```
1 ## for loop to calculate avg.price_per_night
2 prices = []
3
4 for row in read_data[1:]:
5     prices.append(int(row[4]))
6
7 print(f"Avg. prices: {sum(prices)/len(prices)}")
```

```
Avg. prices: 231.5
```

# Modern python

ใช้ pandas module ช่วยในการทำงาน df

```
## modern python
import pandas as pd
df = pd.read_csv("hotel.csv")
df.head()
```

| | hotel_id | name | city | star_rating | price_per_night |
|---|---|---|---|---|---|
| 0 | 101 | Grand Plaza | New York | 5 | 350 |
| 1 | 102 | Ocean View | Miami | 4 | 220 |
| 2 | 103 | Mountain Retreat | Denver | 4 | 185 |
| 3 | 104 | The Urban Stay | Chicago | 3 | 140 |

Next steps: Generate code with df | New interactive sheet

```
df.price_per_night.mean()
```

```
np.float64(231.5)
```

# Write File

# Standard python

```
import csv
```

```
# write file + context Manager
head = ["id", "name", "city"]
body = [
    [1, "CU", "Bangkok"],
    [2, "LSE", "London"],
    [3, "Reading", "Reading"]
]

with open("school.csv","w") as file:
    writer = csv.writer(file)
    writer.writerow(head)
    writer.writerows(body)
```

```
!ls
```

```
hotel.csv  sample_data  school2.csv  school.csv
```

```
!cat school.csv
```

```
id,name,city
1,CU,Bangkok
2,LSE,London
3,Reading,Reading
```

## Modern python

```
import pandas as pd
df = pd.read_csv("school.csv")
df.head()
```

| | id | name | city |
|---|---|---|---|
| 0 | 1 | CU | Bangkok |
| 1 | 2 | LSE | London |
| 2 | 3 | Reading | Reading |

Next steps: Generate code with df | New interactive sheet

```
df.to_csv("school2.csv")
```

```
!ls
```

```
hotel.csv  sample_data  school2.csv  school.csv
```

```
!cat school2.csv
```

```
,id,name,city
0,1,CU,Bangkok
1,2,LSE,London
2,3,Reading,Reading
```

```
## read data from internet
url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/r
```

```
1 import pandas as pd
2 df_penguin = pd.read_csv(url)
```

```
1 df_penguin.head()
```

| | species | island | bill_length_mm | bill_depth_mm | flipper_length_mm |
|---|---|---|---|---|---|
| 0 | Adelie | Torgersen | 39.1 | 18.7 | 181.0 |
| 1 | Adelie | Torgersen | 39.5 | 17.4 | 186.0 |
| 2 | Adelie | Torgersen | 40.3 | 18.0 | 195.0 |
| 3 | Adelie | Torgersen | NaN | NaN | NaN |
| 4 | Adelie | Torgersen | 36.7 | 19.3 | 193.0 |

Next steps: Generate code with df_penguin | New interactive sheet

```
1 ## select column
2 df_penguin[["species", "island", "body_mass_g"]].head(10)
```

| | species | island | body_mass_g |
|---|---|---|---|
| 0 | Adelie | Torgersen | 3750.0 |
| 1 | Adelie | Torgersen | 3800.0 |
| 2 | Adelie | Torgersen | 3250.0 |
| 3 | Adelie | Torgersen | NaN |
| 4 | Adelie | Torgersen | 3450.0 |
| 5 | Adelie | Torgersen | 3650.0 |
| 6 | Adelie | Torgersen | 3625.0 |
| 7 | Adelie | Torgersen | 4675.0 |
| 8 | Adelie | Torgersen | 3475.0 |
| 9 | Adelie | Torgersen | 4250.0 |

```
## filter rows
## จัดบรรทัด python ด้วย \
df_penguin[["species", "island", "body_mass_g"]] \
    .query("species == 'Adelie' and island == 'Torgersen'") \
    .head(10)
```

| | species | island | body_mass_g |
|---|---|---|---|
| 0 | Adelie | Torgersen | 3750.0 |
| 1 | Adelie | Torgersen | 3800.0 |
| 2 | Adelie | Torgersen | 3250.0 |
| 3 | Adelie | Torgersen | NaN |
| 4 | Adelie | Torgersen | 3450.0 |
| 5 | Adelie | Torgersen | 3650.0 |
| 6 | Adelie | Torgersen | 3625.0 |
| 7 | Adelie | Torgersen | 4675.0 |
| 8 | Adelie | Torgersen | 3475.0 |
| 9 | Adelie | Torgersen | 4250.0 |

```
df_penguin[df_penguin['body_mass_g'] >= 4500][["species", "islan
```

| | species | island | body_mass_g |
|---|---|---|---|
| 7 | Adelie | Torgersen | 4675.0 |
| 17 | Adelie | Torgersen | 4500.0 |
| 39 | Adelie | Dream | 4650.0 |
| 45 | Adelie | Dream | 4600.0 |
| 81 | Adelie | Torgersen | 4700.0 |
| 101 | Adelie | Biscoe | 4725.0 |
| 109 | Adelie | Biscoe | 4775.0 |
| 111 | Adelie | Biscoe | 4600.0 |
| 181 | Chinstrap | Dream | 4550.0 |
| 189 | Chinstrap | Dream | 4800.0 |

# Requests API

How to get data via API the easy way API คือ standard Protocol ที่ใช้รับ-ส่งข้อมูลผ่าน Computer ด้วยการวิ่งไปที่ API Endpoint

Local -> Server (Get Request)

- Complete = 200 ## status code
- Site not found = 404
- Server error = 503

Server -> Local (Response)

```
1 import requests
2 url = "https://swapi.info/api/people/1"
```

```
1 response = requests.get(url)
```

```
1 response.json()["name"]
```

```
'Luke Skywalker'
```

```python
## for loop
import requests
import time


character = []
for i in range(5):
    url = f"https://swapi.info/api/people/{i+1}"
    response = requests.get(url)
    response_json = response.json()
    data = [response_json["name"],
            response_json["height"],
            response_json["mass"]
            ]
    character.append(data)
    print(f"Success: {i+1}")
    time.sleep(1)

print(character)
```

```
Success: 1
Success: 2
Success: 3
Success: 4
Success: 5
[['Luke Skywalker', '172', '77'], ['C-3PO', '167', '75'], ['R2-D2',
```

```python
import pandas as pd
starwars = pd.DataFrame(character, columns=["name", "height", "
```

```
Start coding or generate with AI.
```