

เอกสารสอบ Midterm

ชื่อไฟล์ Python101

Print คือ เรียกให้ข้อมูลนั้นออกมา การแสดง Output ออกมาทางหน้าจอ

syntax print >>print(object(s),sep=separator,end=end,file=file,flush=flush

1 sep แสดงตัวคั่นระหว่างคำ (พื้นฐานเป็นช่องว่างหนึ่งช่อง)

2 end ให้แสดงว่าตัวจบเป็นอะไร(พื้นฐาน n\ หรือจบบรรทัด)

print() -> separator ตัวคั่นที่แสดงในวงเล็บ

print() -> end ไม่อยากให้ข้อความที่ปรากฏอยู่บรรทัดเดียวกันให้ใช้คำสั่งนี้

ชนิดของตัวแปร (variables)

- Numeric เราสามารถตรวจสอบชนิดของตัวแปรด้วยคำสั่ง **type**
- modulo (%) การหารเอาเศษ ->> มันจะแสดงผลแค่เศษเท่านั้น ใช้ในการตรวจสอบว่าเลขนั้นเป็นเลขคู่ เลขคี่
- ยกกำลัง (**)

string ("&"")

string +string คือการต่อ string

string -string ทำไม่ได้

Len(ตัวแปร) คือ นับว่าตัวแปรนั้นมีกี่ตัว

Advanced Printing (print code ร่วมกับข้อความ) คือคำสั่งที่ print คำสั่งให้สวยงาม

เราสามารถ print code ร่วมกับข้อความได้ **print(f'ข้อความ{code}')** #ข้างในปีกกาใส่ข้อความอะไรก็ได้

print(f'หนึ่ง บวก สอง (1+2) เท่ากับ สาม ({1+2})')

decimal places (ทศนิยมกี่ตำแหน่ง) || digit

ชนิดของตัวแปร

int(eger) (จำนวนเต็ม) **int** ตรวจสอบชนิดของข้อมูล

float (จำนวนจริง ทศนิยม) **double** เหมือน float แต่แค่ใช้พื้นที่เก็บข้อมูลมากกว่า

boolean (ตัวแปรที่มีค่า True หรือ False) ตรรกศาสตร์

complex (จำนวนเชิงซ้อน) จน.จริง + จน.จินตภาพ

เราสามารถตรวจสอบชนิดของตัวแปรด้วยคำสั่ง **type()** data type เดียวกัน + กันได้ แต่ถ้าคนละ data type + กันไม่ได้

การแปลงชนิดของตัวแปร (Variable Casting) #เปลี่ยนชนิดของตัวแปร

Operation เอาตัวแปรสองตัวมาทำอะไรกัน

(บวก (+), ลบ (-), คูณ (*),หาร (/), modulo (%), and (&), or (|), ยกกำลัง (**))

String operations #การทำงานกับข้อความ

+ -> concat(บวก), * -> repeat (ทำซ้ำ)

replace() การวางแทนที่ string # ใช้เวลาในการทำงานกับ data

split() แยก string

ชื่อไฟล์ Data Structure

list()เก็บได้ทุก Datatype

- เก็บอะไรก็ได้
- ลำดับในการเก็บ มีความสำคัญ

indexing การชี้สมาชิกใน list ด้วย เลขลำดับของ list

list ซ้อน list

การสร้าง list()

เพิ่มสมาชิกเข้าไปใน list ด้วย append()

ลบสมาชิกจาก list จะลบตัวสุดท้ายก่อน

len() ตรวจสอบจำนวนสมาชิกของ list

String is a list of characters

List Slicing :

list[จุดเริ่มต้น (ที่ป):จุดสุดท้าย (ไปรุ่ง)]

list[จุดเริ่มต้น (ที่ป):จุดสุดท้าย (ไปรุ่ง):step]

: colon # กำหนดจุดเริ่มต้นและจุดสุดท้ายของ list นับตัวแรกทางซ้ายเป็น 0 นับตัวแรกทางขวาเป็น -1

Step = list [จุดเริ่มต้น (ที่ป) : จุดสุดท้าย (ไปรุ่ง) :step]

range() # สร้างข้อมูลที่มีตัวเลขเรียงกันมาไว้ใน list

Dictionary

- คล้าย list()
- index เป็นอะไรก็ได้ ไม่จำเป็นต้องเป็นตัวเลขลำดับ
- ลำดับไม่มีความสำคัญ

สร้าง Dictionary

การชี้ สมาชิกใน Dictionary # ชี้ด้วย index ที่เราตั้งขึ้น

การเพิ่มสมาชิกลงไปใน Dictionary

Numpy Array

- เก็บได้แต่ตัวเลข
- อ้างอิงโดยใช้ พิกัด (2 มิติ ก็มองเป็น Matrix)

1D array

- `import numpy` # การเรียกใช้งาน package อันดับแรกที่ต้องทำ

2D array

```
import numpy as np # import แล้วเปลี่ยนชื่อ จาก numpy เป็น np
```

`np.zeros()` # คำสั่งพิเศษแรก พื้นหลังเป็นเลข 0

`np.ones()` คำสั่งพิเศษแรก พื้นหลังเป็นเลข 1

Matrix operations

การบวก ลบ matrix $M \pm N$ บวกลบตัวที่อยู่ตำแหน่งเดียวกัน

การคูณ matrix

- aM Scalar คูณ Matrix (Scalar คูณกระจายเข้าไปทุกตัวของ Matrix)
- $M \times N$ Matrix คูณ Matrix (แถวคูณหลัก)

```
axc,bxc,cxa,cxb
```

สร้าง identity matrix ได้ด้วย `np.eye(ขนาด)`

Matrix Slicing

ชื่อไฟล์ Basic_Programming_Concepts

def = นิยามของฟังก์ชัน

```
def function_name(input_x):  
    # do something with input_x to get output_y  
    return output_y
```

Function ไม่มี process ไม่ได้

input ของ function แบ่งเป็น 2 ชนิด (จำเป็นต้องใส่ กับ ไม่จำเป็นต้องใส่)

ตัวอย่าง function print_grade() มี input 3 ตัว จำเป็นต้องใส่ 2 ตัว คือ name กับ id และไม่จำเป็นต้องใส่ 1 ตัว คือเกรด ซึ่งมีค่า default เป็น F

Looping (for) สำหรับทำงาน ซ้ำๆ

```
for member in listEx:    ## วนที่สมาชิก ใน list  
    do_something()
```

for ชื่อตัวแปร in list ที่จะเอามาใช้เป็นสมาชิกใน ตัวแปร:
วนซ้ำ

- วนซ้ำสมาชิกใน

ใช้ for loop เพิ่มสมาชิกใน list *****

การวนลูปด้วย range() # ตัวช่วยพิเศษ

- สร้าง list 0 - ก่อนเลขนั้น
- วนลูปแบบซ้ำๆ ไม่ต้องการเปลี่ยนแปลงอะไร

Loop ซ้อน Loop (nested loop)

Loop in Function

Conditional Statement (if) เงื่อนไข

operator ที่ใช้ตรวจสอบ condition

ชื่อไฟล์ Pandas101

`import pandas as pd` ก่อนนำไฟล์เข้าต้องรันอันนี้ก่อน ถ้าไม่รันทำอย่างอื่นไม่ได้

CSV = Comma Separated Values

ในการจะใส่ค่าแต่ละค่า จะใช้ comma ในการแบ่ง

.csv = Comma Separated Values

data.go.th

นำ Data เข้า

upload to colab

```
data0 = pd.read_csv('/content/DataBKK01.csv')
data0.head(20) # คำสั่งดูหัวตาราง ดูตั้งแต่หัวไปจนถึงตัวที่ rows 20
data0.tail() # คำสั่งดูท้ายตาราง
data0.shape # คำสั่งดูขนาดของข้อมูล
data0.describe() # ค่าที่เล่าถึงค่าเหล่านี้ได้
```

link with google drive

```
from google.colab import drive
drive.mount('/content/drive') # mount = เชื่อมไดร์ฟ
data1 = pd.read_csv('/content/drive/MyDrive/DataViz23Data/DataBKK01.csv')
data1.head()
```

NaN = Not a Number # คำว่าง

DF คือ DataFrame เป็นชื่อ Data Structure ที่ Pandas ใช้

ชี้ค่าในตาราง (indexing)

มุมมองตาราง df[columns] [rows]

```
data1['Budget_resource'][690] # อยากได้ rows ที่ 690
data1['Budget_resource'][-3:-1] # ใช้ติดลบไม่ได้ แต่ใช้สไลด์ซิงได้
data1[['Approved_buget','Item_name','Department']]
```

มุมมองแบบ Array

Table Slicing ไม่ค่อยมีประสิทธิภาพ ต้องทราบพิกัด

```
data1.iloc[-20:,7:10]
```

table Query

```
small_df = data1.iloc[-5:,:] # 5 ตัวสุดท้ายของ data1
```

```
small_df
```

ป็น list of logics จากตาราง pandas

```
small_df['Approved_budget'] < 1000000
```

Missing Values

ตรวจสอบ Missing (isnull())

```
data1.isnull().any() # .any() เอาค่าความจริงใน column มา OR กัน (ใช้หาค่าว่าง)
```

```
data1.isnull().all() # .all() เอาค่าความจริงใน column มา AND กัน
```

```
data1.isnull().any().any() >> False
```

DropNA

```
data1.dropna(subset=['Fiscal_year'])
```

FillNA

```
data1.fillna('-', inplace=True)
```

การวนลูป record ในตาราง (.iterrows)

```
for row_i in range(len(data1)):
```

```
    print(data1.iloc[row_i,-2:]) # สนใจ column สุดท้าย -2
```

```
    print('-----')
```

การสร้าง columns ใหม่

df['ชื่อ column ใหม่'] = list of data # list of data มีจำนวนสมาชิกเท่ากับจำนวน row ของตาราง

```
small_df['ผู้รับผิดชอบ'] = ['ลุงตุ๋', 'ลุงตุ๋', 'ลุงป้อม', 'ซ้ซชาติ', 'พิธา']
```

การลบ columns

```
small_df.drop(axis=1, columns=['Core_budget_code']) # axis แปลว่า แถว
```

```
new_column = list()
```

```
for row in data1.iterrows():
```

```
    new_col = f'{row[1]['Department']}-{row[1]['Expenditure_category']}-{row[1]['Item_name']}'
```

```
    new_column.append(new_col)
```

การวนลูป แบบมองตารางแพนด้าส์(pandas dataframe) เป็น numpy array (.iloc)

```
for each_row in range(this_data.shape[0]):
```

```
    if (this_data.iloc[each_row,1] <= 20) and (this_data.iloc[each_row,2] == 'ขอนแก่น'): #each_row คือ ตัวเลข
```

```
print(each_row)
print(this_data.iloc[each_row,:])
```

Function ตัวช่วยใน pandas

.describe() คำนวณค่าทางสถิติของข้อมูลที่เป็นตัวเลข

.mean() คำนวณค่าเฉลี่ยของข้อมูลโดยไม่สนใจ missing

.isnull()

ชื่อไฟล์ Pandas102

```
import pandas as pd รันก่อน
import os
from google.colab import drive # เชื่อมบัญชีกับ google drive
drive.mount('/content/drive')
sep_df = pd.read_excel('/content/drive/MyDrive/DataViz23Data/sep65.xlsx')
sep_df
```

ต่อตารางแกน Y (แกน rows เพิ่มจำนวนข้อมูล)

```
sep_oct_df = pd.concat([sep_df, oct_df])
sep_oct_df
```

ต่อตารางในแนวแกน X (เพิ่มรายละเอียดของข้อมูล)

Save Table

```
condo = unit_df[unit_df['propertytype_name_en']=='Condo']
condo.head()
```

Groupby

```
unit_df.groupby('province_en').min()
```

Create Pandas table

Dictionary (row oriented)

```
import pandas as pd # เรียกใช้ package
data_list = [{ 'account': 'Jones LLC', 'Jan': 150, 'Feb': 200, 'Mar': 140 },
              { 'account': 'Alpha Co', 'Jan': 200, 'Feb': 210, 'Mar': 215 },
              { 'account': 'Blue Inc', 'Jan': 50, 'Feb': 90, 'Mar': 95 },
              ] # สร้าง list ขึ้นมา
```

List (row oriented)

```
data = [['Jones LLC',150,200,140],
        ['Alpha Co',200,210,215],
        ['Blue Inc',50,90,95]]
header =['account','Jan','Feb','Mar']
df2 = pd.DataFrame.from_records(data,columns=header) # สร้างตาราง โดยต้องมีหัวตาราง
df2
```

Dictionary (column oriented)

```
data_dict = {'account':['Jones LLC','Alpha Co','Blue Inc'],
             'Jan':[150,250,50],
             'Feb':[200,210,90],
             'Mar':[140,215,95]}
# สร้าง list โดยใช้ตามแนวคอลัมน์
```

List(column oriented)

```
data_item = [['account',['Jones LLC','Alpha Co','Blue Inc']],
             ['Jan',[150,250,50]],
             ['Feb',[200,210,90]],
             ['Mar',[140,215,95]]]
# สร้าง list
```