



# PYTHON FOR NON-PROGRAMMER

DATAROCKIE SCHOOL



## คอร์สนี้สำหรับใคร?

- นักเรียน นักศึกษาที่อยากเริ่มเขียนโปรแกรม
- ไม่จำเป็นต้องมีพื้นฐานอะไรมาก่อนเลย
- (สนใจงานด้าน Data)



# #1 : WHY PYTHON?

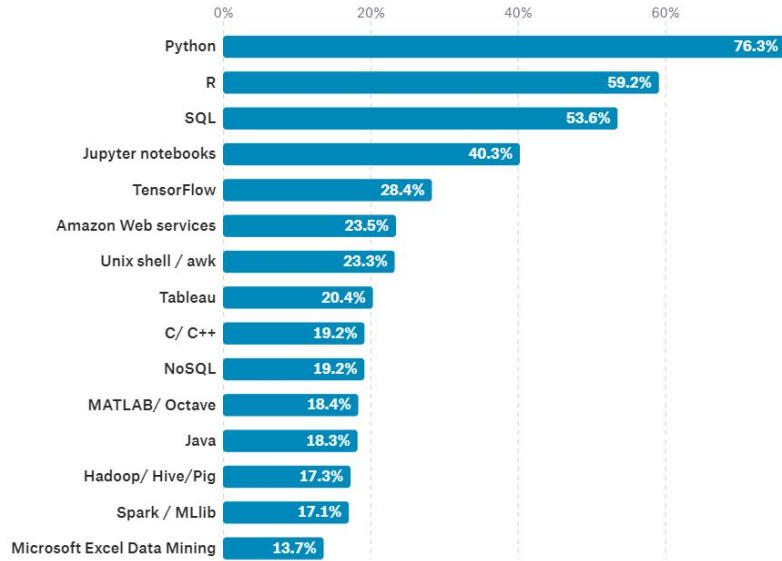


## ทำไมต้องเรียน Python?

- ทรงพลัง (powerful)
- เรียนรู้ง่าย (easy)
- กำลังเป็นที่ต้องการ (high demand)
- ใช้เยอะมากในงาน data science



# ภาษาอันดับหนึ่งของ DS/ DA



Python คือภาษาอันดับหนึ่งสำหรับงาน data science

76.3% บอกว่าใช้ Python ทุกวันเลย



# เงินเดือนไอครด้วย

## Python Developer Salaries in the United States

Salary estimated from 3,784 employees, users, and past and present job advertisements on Indeed in the past 36 months. Last updated: August 28, 2018

Location









United States

Average salary

**\$117,163** per year



## Python Developer salaries by company in the United States

Company	Average salary
 <b>Bank Of America Python Developer</b> 7 salaries Python Developer Job available	<b>\$153,895</b> per year
 <b>stanleyreid Python Developer</b> 24 salaries Python Developer Job available	<b>\$145,813</b> per year
 <b>JPMorgan Chase Python Developer</b> 8 salaries Python Developer Job available	<b>\$145,167</b> per year
 <b>SANS Consulting Services, Inc Python Developer</b> 23 salaries Python Developer Job available	<b>\$142,344</b> per year
 <b>Huxley Banking &amp; Financial Services Python Developer</b> 28 salaries Python Developer Job available	<b>\$134,941</b> per year
 <b>Hunter Bond Python Developer</b> 33 salaries	<b>\$200,000</b> per year
 <b>Inteltec Python Developer</b> 22 salaries	<b>\$157,703</b> per year
 <b>Infinite Resources, Inc. Python Developer</b> 40 salaries	<b>\$160,000</b> per year



## ทำไมต้องเรียนคอร์สนี้?

- เนื้อหาแน่นมาก โคตรดี
- ใช้เทคโนโลยีล่าสุด (2018) ในการสอนอย่าง Google Colab
- เรียนฟรีทุกอย่าง
- แอดมินสอนเก่ง 555+



# Content Curators



Toy, Data Analyst



Top, Data Scientist



Un, Data Scientist





# เนื้อหาคอร์สนี้มีอะไรบ้าง?

เนื้อหาแบ่งออกเป็น 6 บท

(ใช้เวลาเรียนประมาณสามชั่วโมง รวมทำ Quiz)

1. Basic Python
2. Data Types
3. Control Flow
4. Data Structures
5. Functions
6. Python for Data Science

## Style การสอน

- 20 วิดีโอ (full HD)
- 20 notebook ไว้ฝึกเขียนโค้ดออนไลน์
- Quiz 100 ข้อ เพื่อทดสอบความรู้ !!



# Building Blocks

## Control Flow Data Structures

ควบคุมพฤติกรรมของโปรแกรมที่เราเขียน และเข้าใจ data structures ที่สำคัญของ Python ประกอบด้วย list, tuple, dictionary

2

## Functions Data Science

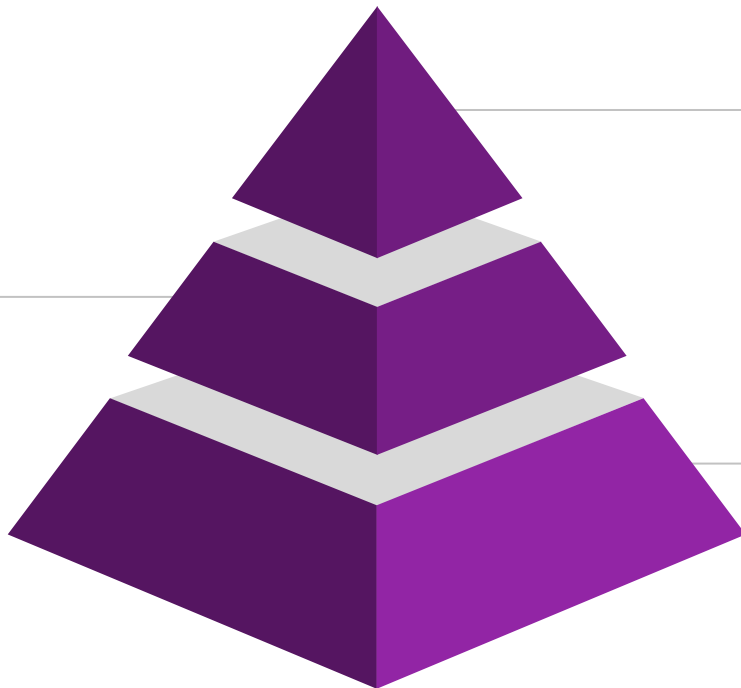
ฝึกเขียน user defined function เขียนฟังก์ชันไว้ใช้งานเอง ลอง import library ที่สำคัญสำหรับการวิเคราะห์ข้อมูลใน python เช่น numpy pandas matplotlib

3

## Variables Data Types

การเขียนโปรแกรมเบื้องต้น การสร้างตัวแปร (variable) Basic data types ของ Python ประกอบด้วย int float str bool

1





# เราจะเขียน Python ผ่านอะไร?



นักเรียนไม่ต้องติดตั้ง Python ให้ยุ่งยาก

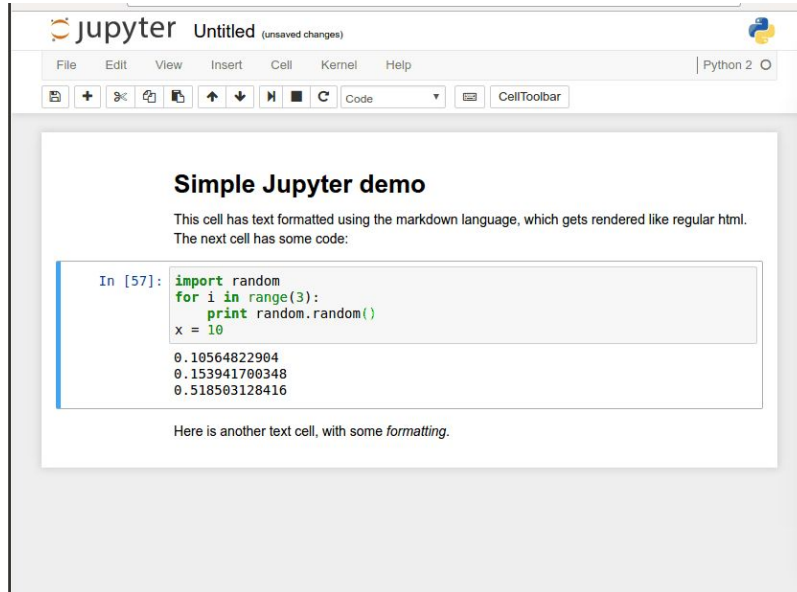
แล้วเราจะไปเขียน Python กันบน **Google Colab**  
(บน Google Chrome Browser) ใช้งานฟรี !!

สำหรับนักเรียนที่ยังไม่มี Chrome ดาวน์โหลดที่นี่  
<https://www.google.com/chrome/>

แล้วเข้าสู่ Google Colab ที่นี่  
<https://colab.research.google.com/>



# WHAT IS GOOGLE COLAB?



Python เป็น **interpreted language** เราสามารถพิมพ์โค้ด แล้วกด run เพื่อดู output ได้เลย

ปกติเราเลยเขียน Python บน Jupyter Notebook (ลงในคอมพิวเตอร์ของเราได้เลย เขียน offline ไม่ต้องต่ออินเทอร์เน็ต)

แต่เวลาลง Python จะยุ่งยากมากสำหรับมือใหม่ เราเลยสอนบน Google Colab แทน (มันคือ Jupyter version Google !!)



# HELLO WORLD

**# Your first function in any language**

**# สอนคอมพิวเตอร์ให้พูด Hello World**

```
print("Hello World")
```

```
print('Hello World')
```



## #2: PYTHON FOUNDATION



# BASIC CALCULATION

**# Python เขียน + - \* / % \*\* ( ) ได้เหมือนเครื่องคิดเลข**

- `2 + 2`
- `5 - 3`
- `3 * 3`
- `3 / 2`
- `10 % 3 # modulo`
- `3 ** 3 # power/ exponent`
- `25 ** 0.5 # sqrt`





# VARIABLES

**# สร้างตัวแปรง่ายๆด้วย =**

```
variable_name = value
```

**# ตัวอย่าง**

```
x = 100
```

```
y = 200
```

```
hello_world = 'hello world!'
```

```
my_name = 'toy'
```

```
your_name = 'andy'
```

วิธีการตั้งชื่อที่ดี

1. ใช้ตัวพิมพ์เล็ก
2. ห้ามขึ้นต้นด้วยตัวเลข
3. ถ้าชื่อยาว ให้เชื่อมคำด้วย \_



# VARIABLES

**# เอาตัวแปรมาบวกลบคูณหารกันก็ได้**

```
income = 20000  
food_expense = 5000  
saving = income - food_expense  
  
print(saving)
```



# DATA TYPES

## # เบื้องต้นมีแค่ 4 แบบเอง

- **int:** 1 2 3 4 5 6 7 8 9
- **float:** 2.4 3.5 6.9 10.2
- **string:** 'hello world' 'food' 'dog'
- **bool:** True False

เราสามารถเช็ค data type ด้วย  
ฟังก์ชัน type()



# WORK WITH STRING

String จะอยู่ในเครื่องหมายคำพูด

```
word1 = "Python"
```

```
word2 = "Hello World"
```

```
print(word1)
```

```
print(word2)
```

```
print(len(word1)) # นับความยาวของ string นั้นๆ
```



# STRING INDEX

P	Y	T	H	O	N
0	1	2	3	4	5

index ใน Python เริ่มที่ค่าศูนย์ (0)



P	Y	T	H	O	N
0	1	2	3	4	5

เราสามารถดึงค่าจาก word = "PYTHON" ได้ด้วย slice

word[0] → "P"

word[1] → "Y"

word[2] → "T"

word[-1] → "N"

word[-2] → "O"



# **SLICE [ : ]**

P	Y	T	H	O	N
0	1	2	3	4	5

```
word = "PYTHON"
```

```
word[0:3] → "PYT"
```

```
word[0:4] → "PYTH"
```

```
word[3:5] → "HO"
```



# FUNCTION vs. METHOD

[https://www.w3schools.com/python/python\\_ref\\_string.asp](https://www.w3schools.com/python/python_ref_string.asp)

## # example word

```
word = "hello world"
```

## # function

```
print(word)
len(word)
type(word)
```

## # method คือฟังก์ชันอีกประเภทหนึ่ง ที่เฉพาะเจาะจงไปที่ class ของ object นั้นๆ เช่น string

```
word.upper()
"HELLO WORLD"
```

```
word.split()
["hello", "world"]
```

```
word.find("w")
```





## STRING METHOD

"Hello".          ( )



## #3 : CONTROL FLOW



# CONTROL FLOW

# Control Flow ใช้ในการควบคุมพฤติกรรมของโปรแกรมเรา

# ใน Python มีแค่สามแบบเอง (ง่ายอะไรเบอร์นี้)

- If-else
- For loop
- While loop



# IF-ELSE

```
# student score

score = 72

if score >= 80:
    print("you passed")
else:
    print("you failed")
```



# INDENTATION MATTERS!

```
# student score
```

```
score = 72
```

```
if score >= 80:  
    print("you passed")
```

```
else:  
    print("you failed")
```

เราใช้ย่อหน้า (indent) เพื่อบอก Python  
ว่านี่คือ code block เดียวกัน



# IF-ELIF-ELSE

```
# student score
```

```
score = 72
```

```
if score >= 80:
```

```
    print("A")
```

```
elif score >= 70:
```

```
    print("B")
```

```
else:
```

```
    print("Failed")
```



# RANGE() FUNCTION

**# range() function**

```
print(range(5))
```

```
0,1,2,3,4
```

```
print(range(5,10))
```

```
5,6,7,8,9
```

```
print(range(5,20,3))
```

```
5,8,11,14,17
```



# FOR LOOP

```
# for loop is very easy  
for i in range(5):  
    print(i)
```

```
0  
1  
2  
3  
4
```





# FOR LOOP

```
# for loop is very easy  
for i in range(5):  
    print("hello world")
```

```
hello world  
hello world  
hello world  
hello world  
hello world
```



# WHILE LOOP

```
# while loop is also very easy
```

```
x = 5
```

```
while x >= 0:
```

```
    print("x")
```

```
    x = x - 1 # update variable x
```

5

4

3

2

1

0



## #4: DATA STRUCTURES



# DATA STRUCTURES

## # Basic Data Structures ที่เราใช้กันเยอะๆใน Python

- `list`: `[1, 2, 3, 4, 5]`
- `tuple`: `(1, 2, 3, 4, 5)`
- `dictionary`: `{ 'A' : 1, 'B' : 2, 'C' : 3 }`



```
# create list
```

```
list_a = [1,2,3,4,5]
```

```
list_b = ['hi', 'toy', True, False, 200]
```

```
# slice list same as string
```

```
List_b[0] # ['hi']
```

```
list_b[1] # ['toy']
```

```
list_b[1:3] # ['toy', True]
```

```
list_b[1:4] # ['toy', True, False]
```



# LIST METHOD

```
# create list
```

```
list_b = ['hi', 'toy', True, False, 200]
```

```
# .append()
```

```
list_b.append(300)
```

```
print(list_b) # ['hi', 'toy', True, False, 200, 300]
```

```
# .pop()
```

```
list_b.pop(0)
```

```
print(list_b) # ['toy', True, False, 200, 300]
```



# LIST + LIST

```
# create list_a list_b
```

```
list_a = [1,2,3]
```

```
list_b = [4,5,6]
```

```
# combine list
```

```
list_c = list_a + list_b
```

```
print(list_c)
```

```
# [1,2,3,4,5,6]
```



# WE CAN UPDATE LIST

```
# create list_a
list_a = [1,2,3]

# update values in list_a
list_a[0] = 100
list_a[1] = 200
list_a[2] = 300

print(list_a)
```



# TUPLE

# tuple เหมือนกับ list เกือบทุกอย่าง แต่เรา update ค่าใน tuple ไม่ได้

# create tuple

```
tuple_a = (1,2,3,4,5)
```

```
tuple_b = ('hi', 'toy', True, False, 200)
```

# พยายามอัปเดตค่า tuple จะขึ้น error

# immutable

```
tuple_a[0] = 100 # error !!!
```



# FOR LOOP + LIST/ TUPLE

```
list_a = [1,2,3,4,"hello"]
```

```
for element in list_a:  
    print(element)
```

1

2

3

4

hello



# DICTIONARY

**# key-value pairs**

```
dict_a = {  
    "John": 25,  
    "Jack": 30,  
    "Mary": 29  
}
```

**# extract value from dict\_a**

```
dict_a["John"] #25  
dict_a["Jack"] #30  
dict_a["Mary"] #29
```



# UPDATE DICTIONARY

## **# Update**

```
dict_a = {  
    "John": 25,  
    "Jack": 30,  
    "Mary": 29  
}  
  
dict_a["Jack"] = 33
```

## **# Add new value**

```
dict_a["Toy"] = 28
```



# PRINT DICTIONARY

```
# create dictionary
```

```
dict_a = {  
    "John": 25,  
    "Jack": 30,  
    "Mary": 29  
}
```

```
# print keys / values
```

```
dict_a.keys()  
dict_a.values()
```



# FOR LOOP DICT

```
# print key, value in dict_a using for-loop
```

```
dict_a = {  
    "John": 25,  
    "Jack": 30,  
    "Mary": 29  
}
```

```
for key, value in dict_a.items():  
    result = "key:" + key + " value:" + str(value)  
    print(result)
```

```
key:John value:25  
key:Jack value:30  
key:Mary value:29
```



# COMPARE MAIN DATA STRUCTURES

## List

- Ordered
- Mutable

## Tuple

- Ordered
- IMmutable

## Dict

- UNordered
- Mutable

**ตรงนี้สำคัญมาก !!**

Ordered แปลว่าเรียงลำดับ สามารถ slice ด้วย index ได้

Mutable แปลว่าสามารถแก้ไขข้อมูลใน data structure นั้นได้



# THE CLIFF DWELLERS

```
text = """
Between the former site of old Fort Dearborn and the present site of our newest Board of Trade there lies a restricted yet tumultuous territory through wh
High above this architectural upheaval rise yet other structures in crag-like isolation. El Capitan is duplicated time and again both in bulk and in statu
The explorer who has climbed to the shoulder of one of these great captains and has found one of the thinnest folds in the veil may readily make out the n
From such conditions as these—along with the Tacoma, the Monadnock, and a great host of other modern monsters—towers the Clifton. From the beer-hall in it
The tribe inhabiting the Clifton is large and rather heterogeneous. All told, it numbers about four thousand souls. It includes bankers, capitalists, lawy
All these thousands gather daily around their own great camp-fire. This fire heats the four big boilers under the pavement of the court which lies just be
In a word, the Clifton aims to be complete within itself, and it will be unnecessary for us to go afield either far or frequently during the present simpl
"""
```

## # Case Study

เราจะตัดคำ (tokenize) บทนำของหนังสือ The Cliff Dwellers แล้วนับคำใส่ dictionary ว่า word ไหนเกิดขึ้นเยอะสุดในบทนำนี้





## #5: USER DEFINED FUNCTION



# DEFINE FUNCTION

# เขียนฟังก์ชันใช้เองใน Python ง่ายมาก ใช้ **def** keyword

```
def function_name( ) :  
    do something
```



# DEFINE FUNCTION

```
def sum_two_nums(a, b):  
    return a+b
```

```
# test function
```

```
sum_two_nums(6, 4) #10
```

```
sum_two_nums(10, 2) #12
```



# RETURN VS. PRINT

```
def sum_two_nums1(a, b):  
    return a+b
```

```
def sum_two_nums2(a, b):  
    print(a+b)
```

```
# test function
```

```
temp1 = sum_two_nums1(10, 2)
```

```
temp2 = sum_two_nums2(10, 2)
```

```
# print temp1, temp2
```

```
print(temp1) # 12
```

```
print(temp2) # None
```

เราใช้ return เพื่อสร้าง variable ใหม่ได้



# TIME TO HONE YOUR SKILLS



การฝึกเขียนโค้ดด้วยตัวเองสำคัญมาก  
Tutorial นี้เราจะมาลองเขียน **10 functions** ด้วยกัน

แบบฝึกหัด:

<https://drive.google.com/open?id=1JYI5mTOaF68bJLDkR9PMpeVNUq3hol5P>

เฉลย:

<https://colab.research.google.com/drive/1H6A1wTtr9EFt9luFWDiR4Eetfar0dOna>

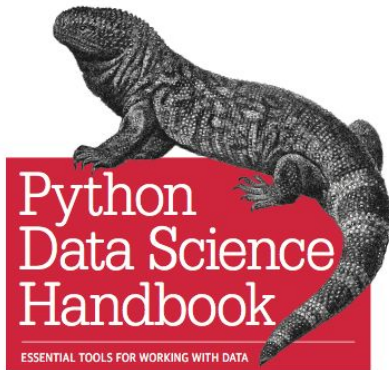


# #6: PYTHON FOR DATA SCIENCE



# PYTHON FOR DATA SCIENCE

O'REILLY



Jake VanderPlas

เราสามารถ **import library** ที่ใช้ทำงาน data science มาใส่ notebook (google colab) ของเราได้เลย

Library สำคัญที่ใช้ตลอดเลย ประกอบด้วย

- Numpy
- Pandas
- Matplotlib



# IMPORT LIBRARY

แค่พิมพ์โค้ดนี้ เพื่อ import library เข้า colab

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

ชื่อเต็มของ library

ชื่อเล่น





# INTRO TO NUMPY

```
import numpy as np
```

```
my_list = [1,2,3,4,5,6]  
arr = np.array(my_list)
```

```
np.mean(arr)  
np.median(arr)  
np.sum(arr)  
np.std(arr)  
np.var(arr)  
np.max(arr)  
np.min(arr)
```

```
arr <= 4
```

**np** . \_\_\_\_\_ ( )

วิธีการเรียกใช้งานฟังก์ชันของ numpy ต้อง  
ใส่ชื่อเล่นก่อนแล้วค่อยตามด้วยชื่อฟังก์ชัน  
np.func\_name()



# INTRO TO PANDAS

```
In [33]: data
```

```
Out[33]:
```

	Area Abbreviation	Area Code	Area	Item Code	Item	Element Code	Element	Unit	latitude	longitude	...	Y2004	Y2005	Y2006	Y2007	Y2008	Y2009
0	AF	2	Afghanistan	2511	Wheat and products	5142	Food	1000 tonnes	33.94	67.71	...	3249.0	3486.0	3704.0	4164.0	4252.0	4538.0
1	AF	2	Afghanistan	2805	Rice (Milled Equivalent)	5142	Food	1000 tonnes	33.94	67.71	...	419.0	445.0	546.0	455.0	490.0	415.0
2	AF	2	Afghanistan	2513	Barley and products	5521	Feed	1000 tonnes	33.94	67.71	...	58.0	236.0	262.0	263.0	230.0	379.0
3	AF	2	Afghanistan	2513	Barley and products	5142	Food	1000 tonnes	33.94	67.71	...	185.0	43.0	44.0	48.0	62.0	55.0
4	AF	2	Afghanistan	2514	Maize and products	5521	Feed	1000 tonnes	33.94	67.71	...	120.0	208.0	233.0	249.0	247.0	195.0
5	AF	2	Afghanistan	2514	Maize and products	5142	Food	1000 tonnes	33.94	67.71	...	231.0	67.0	82.0	67.0	69.0	71.0
6	AF	2	Afghanistan	2517	Millet and products	5142	Food	1000 tonnes	33.94	67.71	...	15.0	21.0	11.0	19.0	21.0	18.0
7	AF	2	Afghanistan	2520	Cereals, Other	5142	Food	1000 tonnes	33.94	67.71	...	2.0	1.0	1.0	0.0	0.0	0.0
8	AF	2	Afghanistan	2531	Potatoes and products	5142	Food	1000 tonnes	33.94	67.71	...	276.0	294.0	294.0	260.0	242.0	250.0
9	AF	2	Afghanistan	2536	Sugar cane	5521	Feed	1000 tonnes	33.94	67.71	...	50.0	29.0	61.0	65.0	54.0	114.0
10	AF	2	Afghanistan	2537	Sugar beet	5521	Feed	1000 tonnes	33.94	67.71	...	0.0	0.0	0.0	0.0	0.0	0.0

Column = variable  
Row = observation (data)

เราใช้ pandas จัดการ dataframe และวิเคราะห์ข้อมูลอย่างรวดเร็ว !! (pandas มัน built on-top numpy อีกที)



# INTRO TO PANDAS

```
import pandas as pd
```

```
dict_a = {  
    "year": [2017, 2018, 2019],  
    "pop": [1000, 2000, 2500],  
    "country": ["TH", "JP", "US"]  
}
```

```
# create dataframe
```

```
df = pd.DataFrame(dict_a)
```

```
# print df
```

```
df.head()
```

```
df.tail()
```

```
df.info()
```

```
df.describe()
```

pd. \_\_\_\_\_ ( )



**CONGRATULATIONS**



# COURSE SUMMARY

- ตอนนี้ทุกคนเรียนจบ ท้าวแรกของการเป็น Python Dev. แล้ว
- Basic Python
- Data Types: int float str bool
- Data Structures: list tuple dictionary
- Function vs. Method
- Control Flow
- User Defined Function
- Intro to Python for Data Science
- เรียนกันโคตรเยอะ 555+

คอร์สฟรี ออนไลน์ดี ยังมีอีกเพียบ (และจะออกใหม่ทุกเดือน)  
เข้ามาเรียนเลยที่ <https://datarockie.com>



## Our Popular Courses



### Not Just PowerPoint

เรียนเทคนิคการสร้าง Presentations อย่างมืออาชีพ



Kasidis S. (Toy)

FREE



### Intro to R Programming

ทุกอย่างที่คุณต้องรู้เกี่ยวกับพื้นฐาน R Programming



Kasidis S. (Toy)

FREE



### Intro to SQL for Data Analysis

วิเคราะห์ข้อมูลอย่างมั่นใจด้วย SQL ในระดับเริ่มต้น



Kasidis S. (Toy)

FREE



เป็นกำลังใจให้เพจเราด้วยการกด **RECOMMEND** facebook  
เพจของเราที่ <https://www.facebook.com/datarockie/>

สนับสนุนเพจ DataRockie ด้วยการกด Recommend ด้วย  
น้ำครับ พิมพ์คอมเม้น แนะนำติชมได้หมดเลยครับบบ

ชอบกด Like ใช้กด Share  
แอดรักทุกคน ชู๊บบบบ <3



**WE HOPE YOU ENJOY THE CLASS**

**DATAROCKIE SCHOOL**