

# Python Data Science for Developer



# All About Me Boyd



**Sorratat Sirirattanajakarin,  
Boyd**

Data Scientist @3dsinteractive

“Data is a new UX”

## Education

- Master degree, DPU, Big Data Engineering
- Bachelor of Science, B.Sc. (Hons),  
Agro-industry, Biotechnology, KU



## ประวัติการทำงานภาคเอกชน

2018 - ปัจจุบันทำงานในตำแหน่ง Data Scientist ในเรื่องของ AI Marketing Automation ณ ที่ บริษัท 3dsinteractive

2017 - ทำงานอยู่ในบริษัท Startup ของต่างประเทศในเครือของ Capstream Ventures Inc ในตำแหน่งของ Digital marketing Executive and Data Scientist (นักการตลาดออนไลน์ นักวิทยาศาสตร์ และวิเคราะห์ด้านข้อมูล)

2016 - Vice Digital Marketing Manager บริษัท Startup จากต่างประเทศ ดูแลเกี่ยวกับการตลาดด้านการศึกษา ผ่าน Social media และ Online Channel

2013 - ประสบการณ์ทำงานในเครือ Thaibeve 3 ปี (หัวหน้าฝ่ายปฏิบัติการการผลิตเบียร์ และให้คำแนะนำฝ่ายการตลาด รวมถึงชุมชน)

2012 - ประสบการณ์ทำงาน Startup กับเครือต่างชาติและช่วยในการวิจัย และออกแบบผลิตภัณฑ์ใหม่ (ผู้ช่วยแผนกต้นแบบออกแบบผลิตภัณฑ์ และฝ่ายขาย)



## ประวัติการทำงานภาคสังคม

- อาจารย์ และวิทยากรด้านการตลาดออนไลน์ ที่บริษัท "siamhr" จัดสอนและอบรมให้กับทาง ระดับผู้จัดการ ผู้อำนวยการ และผู้ประกอบการ

- อาจารย์ และวิทยากรด้านการตลาดออนไลน์ ให้กับผู้ประกอบการที่ สำนักงานพาณิชย์ จังหวัดปทุมธานี และชุมชน BizClub จังหวัดปทุมธานี

- กรรมการด้าน สารสนเทศและเทคโนโลยี (การตลาดออนไลน์) และให้คำแนะนำเพื่อสร้าง ความแข็งแกร่งให้กับผู้ประกอบการไทย ที่ "สมาคมนักธุรกิจอาชีวิน"

- สมาชิกกลุ่ม BKK AI Volunteer มอบความรู้ด้าน AI ให้กับผู้ที่เริ่มต้นไปจนถึงขยายความรู้ ด้าน AI ให้กับวงข่าวยิ่งขึ้น



**FANPAGE - [www.facebook.com/bigdatarpg](https://www.facebook.com/bigdatarpg)**



# Morning

---

Day-01



Boyd BigData RPG - Sorratat Sirirattanajakarin

1. **Introduction to Data Science**
2. Install Anaconda and Jupyter
3. Python Programming Basics
  1. Operation
  2. Variable
  3. Flow Control
  4. Condition ( If, For, While )
  5. Function, Method
  6. List, Tuple
  7. Dictionaries and Structural Data
  8. Manipulating Strings

# Morning

---

Day-01



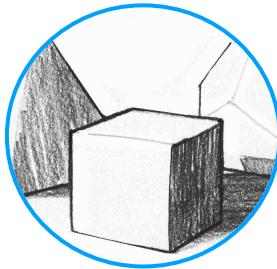
Boyd BigData RPG - Sorratat Sirirattanajakarin

1. **Introduction to Data Science**
2. Install Anaconda and Jupyter
3. Python Programming Basics
  1. Operation
  2. Variable
  3. Flow Control
  4. Condition ( If, For, While )
  5. Function, Method
  6. List, Tuple
  7. Dictionaries and Structural Data
  8. Manipulating Strings

# Introduction to Data Science



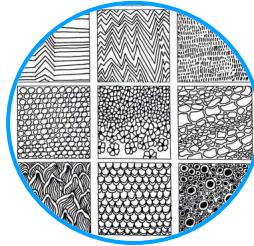
## Big Data



Volume



Velocity



Variety



Veracity

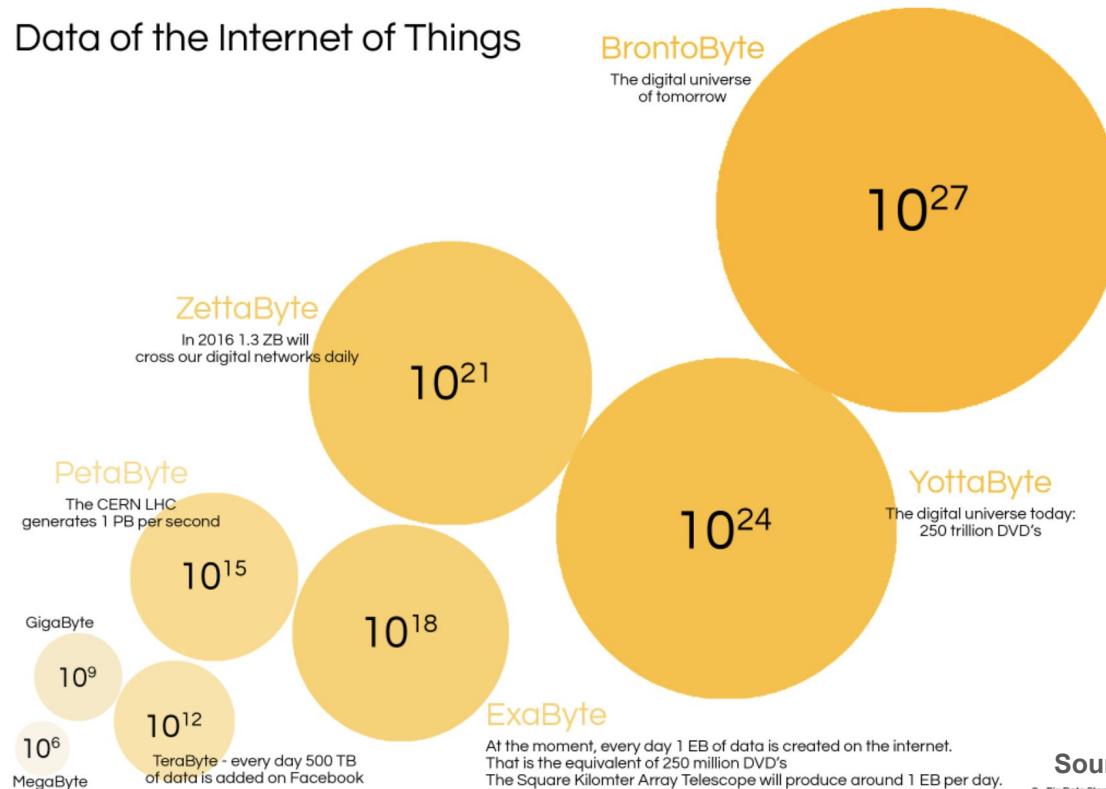


Value



# Introduction to Data Science

## Data of the Internet of Things



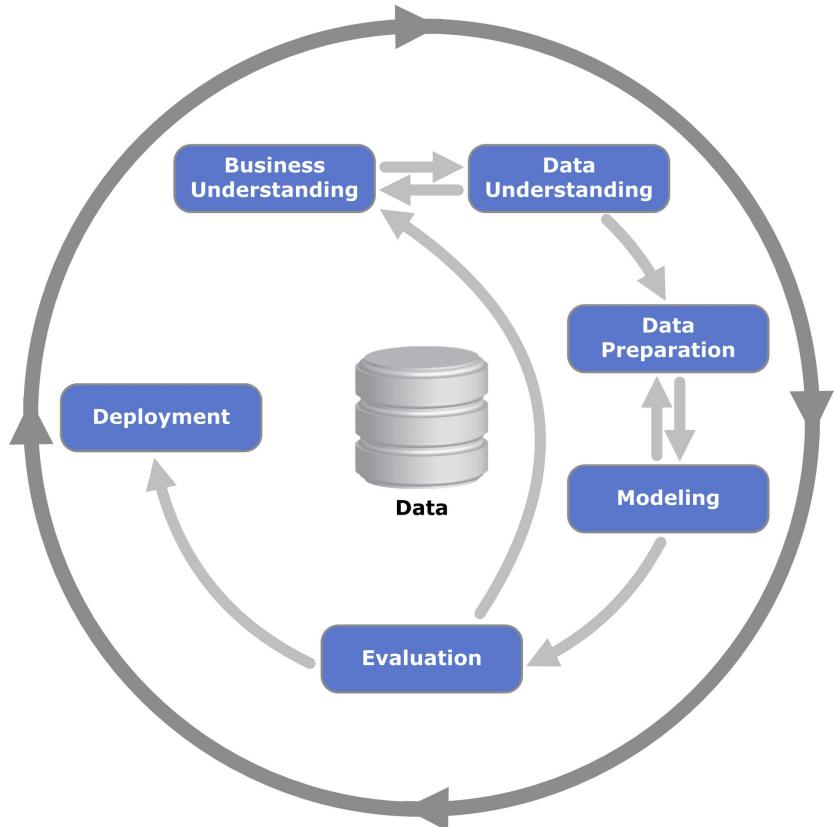
## Size of Data

Which size of data,  
you are facing with ?

Source: <https://datafloq.com/read/big-data-infographic/226>



# Introduction to Data Science



## CRISP-DM

Cross-industry standard process for data mining

Most Popular Data Science  
Thinking Framework in  
KD Nuggets website in  
2002, 2004, 2007 and 2014

Source: <https://www.kdnuggets.com/>



# Introduction to Data Science

---

## Business Understanding

ตัวอย่าง:

1. คาดการณ์ว่าหากทำการวิเคราะห์ข้อมูลลูกค้าจะทำให้ Conversion rate +50%
2. ต้องการทำ Personalized Content ให้ถูกต้องเริ่มจากการ Clustering
3. Feature อะไรที่มีผลกับ Conversion Rate มากที่สุดใน Facebook Channel
4. เวลาไหนที่เหมาะสมที่สุดในการส่ง Campiagn ให้กับแต่ละกลุ่ม Segment



# Introduction to Data Science

## Data Understanding

| Distribute Type | Description | Example                                 |                                    |
|-----------------|-------------|---|------------------------------------|
| Categorical     | Nominal     | ข้อมูลแต่ละส่วนแยกออกจากกันชัดเจน       | Zip code, Gender, City, Name       |
|                 | Ordinal     | ข้อมูลแต่ละส่วนมีความล้มพังอีกเป็นลำดับ | Grade, Class, {easy, normal, hard} |
| Numeric         | Interval    | ความแตกต่างระหว่างค่ามีความหมาย         | Temperature, Timestamp             |
|                 | Ratio       | ความแตกต่างระหว่างอัตราส่วนมีความหมาย   | Age, Weight, Length                |

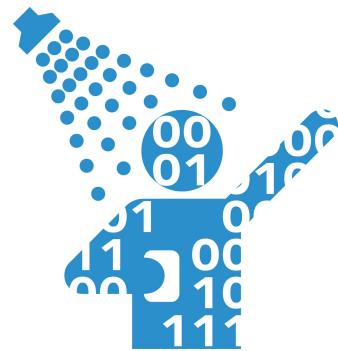


# Introduction to Data Science

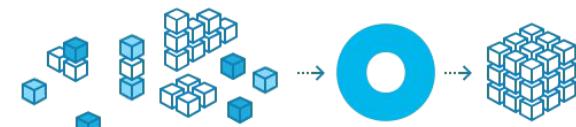
## Data Preparation



Data Selection



Data Cleaning

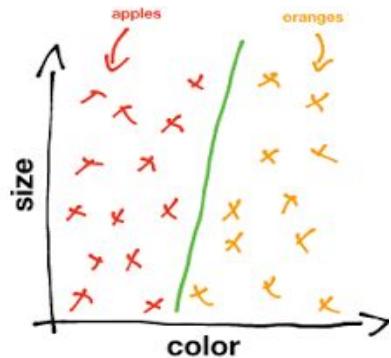


Data Transformation

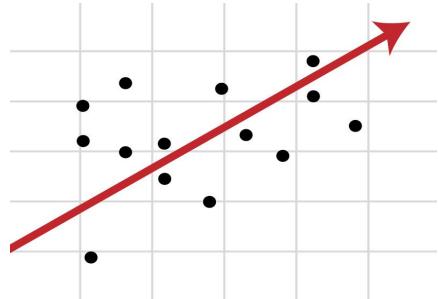


# Introduction to Data Science

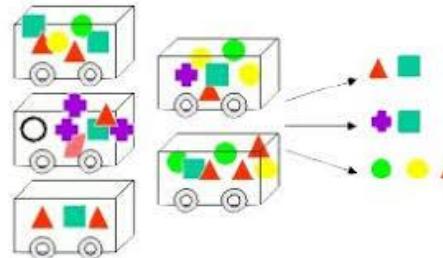
## Modeling



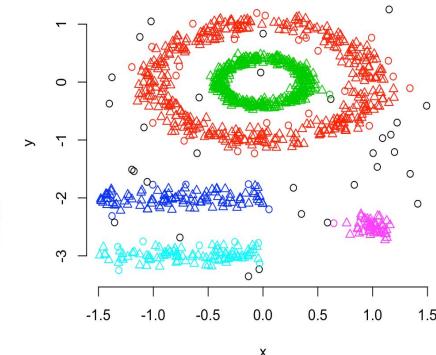
**Classification**



**Regression**



**Association  
Rule**



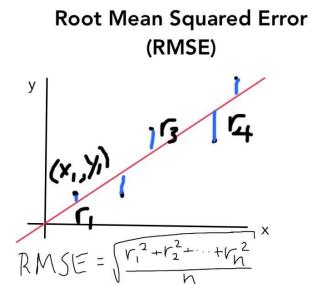
**Clustering**



# Introduction to Data Science

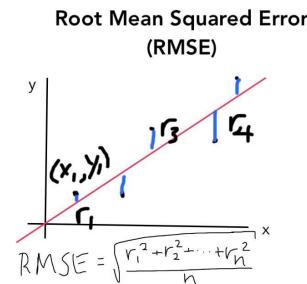
## Evaluation

|         | Predicted |         |         |    |
|---------|-----------|---------|---------|----|
|         | Class A   | Class B | Class C |    |
| Actual  | Class A   | 100     | 0       | 10 |
| Class B | 10        | 80      | 10      |    |
| Class C | 30        | 0       | 70      |    |



Rule:  $X \Rightarrow Y$

$$Support = \frac{frq(X, Y)}{N}$$
$$Confidence = \frac{frq(X, Y)}{frq(X)}$$
$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$



### Classification

**Accuracy**

### Regression

**RMSE**

### Association Rule

**Confidence,  
Support**

### Clustering

**RMSE**



# Introduction to Data Science

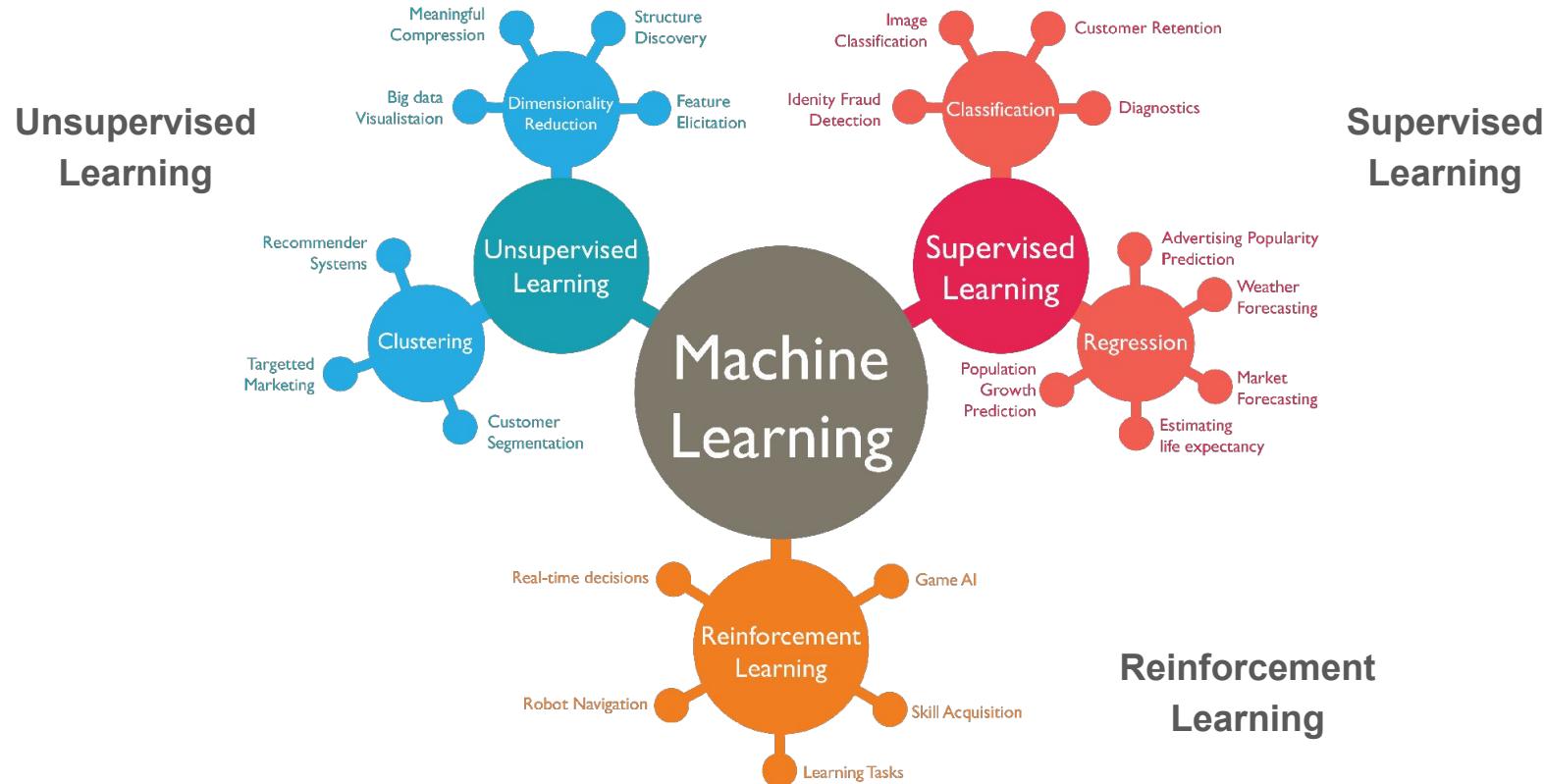
## Deployment

ตัวอย่าง:

1. คาดการณ์ว่าหากทำการวิเคราะห์ข้อมูลลูกค้าจะทำให้ Conversion rate +50%  
**สามารถทำให้ Conversion rate +100%**
2. ต้องการทำ Personalized Content ให้ถูกต้องเริ่มจากการ Clustering  
**สร้าง Clustering ขึ้นมาเพื่อทำ Personalized Content**
3. Feature อะไรที่มีผลกับ Conversion Rate มากที่สุดใน Facebook Channel  
**Feature Devices มีผลกับ Conversion Rate มากที่สุดใน Facebook Channel**
4. เวลาไหนที่เหมาะสมที่สุดในการส่ง Campiagn ให้กับแต่ละกลุ่ม Segment  
**เวลา 9:00 - 11:00 เหมาะสมที่สุดในการส่ง Campaign ให้กับ Segment A**



# Introduction to Data Science





# Descriptive Statistics

## The Central Moment of Feature

ค่าเฉลี่ยของข้อมูล

การกระจายของข้อมูล

ความเบี้ยวของข้อมูล

ความโด่งของข้อมูล

**Mean**

**Variance**

**Skewness**

**Kurtosis**

**The First Central  
Moment**

**The Second Central  
Moment**

**The Third Central  
Moment**

**The Fourth Central  
Moment**



# Descriptive Statistics

## Data Set

[ 1,1,2,3,4,5,10 ] | n = 7

ผลรวมทั้งหมด  
หาร  
จำนวนทั้งหมดของข้อมูล

**3.714**

**Mean**

ค่าที่เกิดขึ้น  
โดยมีความถี่มากที่สุด

**1**

**Mode**

ค่ากลางของข้อมูล

**3**

**Median**



# Descriptive Statistics

Data Set: [ 1,1,2,3,4,5,10 ] | n = 7

$$S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}$$

"ค่าที่ใช้วัดการกระจายของข้อมูล"

$$S.D. = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}}$$

"ระยะห่างเฉลี่ยระหว่างข้อมูลทั้งหมด กับค่ากลาง"

8.489

**Variance**

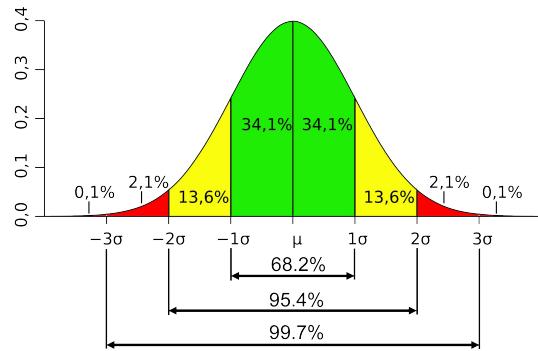
2.913

**Standard deviation**  
ค่าเบี่ยงเบนมาตรฐาน

ใช้วัดการกระจายของข้อมูล



# Descriptive Statistics



Median

Mode

Median

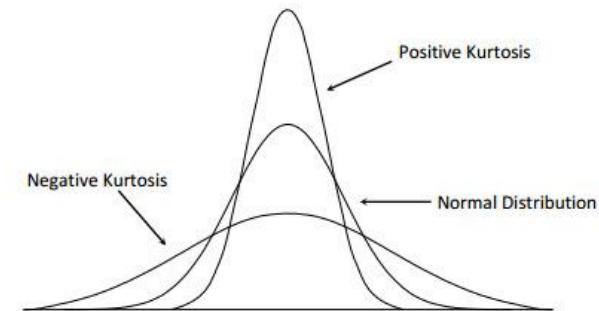
Mode

Positive Skew

Negative Skew

กราฟเมี้ยว

กราฟเบี้ยว



**Normal  
distribution**

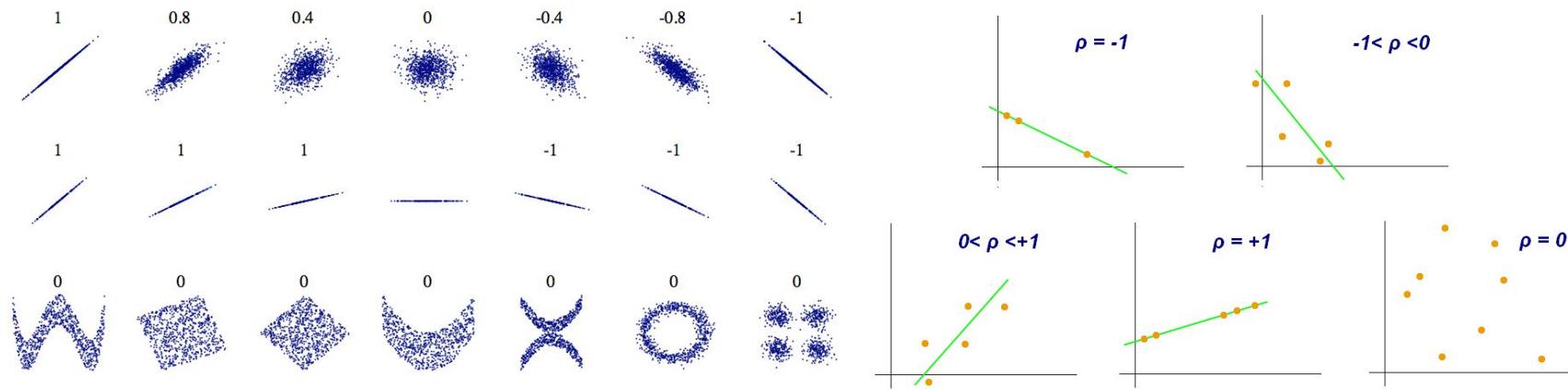
**Skewness**

**Kurtosis**



# Correlation

Correlation คือ การหาความสัมพันธ์กันของตัวแปรอิสระ (เข้าใจง่ายๆคือ ข้อมูลระหว่าง 2 Features)  
ค่า Correlation มีค่าระหว่าง  $[-1, 1]$



Pearson correlation coefficient

# Morning

---

Day-01



Boyd BigData RPG - Sorratat Sirirattanajakarin

1. Introduction to Data Science
2. **Install Anaconda and Jupyter**
3. Python Programming Basics
  1. Operation
  2. Variable
  3. Flow Control
  4. Condition ( If, For, While )
  5. Function, Method
  6. List, Tuple
  7. Dictionaries and Structural Data
  8. Manipulating Strings



# Install Anaconda and Jupyter

1. เข้าไปที่เวปไซต์ <https://www.anaconda.com/download/>
2. เลือกรอบน ปฏิบัติการที่ใช้ของเครื่องที่เราจะทำการติดตั้ง

The screenshot shows the 'Download Anaconda Distribution' page. At the top, it says 'Version 5.2 | Release Date: May 30, 2018'. Below that is a button labeled 'Download For:' with icons for Windows, macOS, and Linux. A blue circle with the number '1' is drawn around the URL in the list above. A blue circle with the number '2' is drawn around the 'Download For:' button.

**Download Anaconda Distribution**  
Version 5.2 | Release Date: May 30, 2018

Download For:

**High-Performance Distribution**  
Easily install 1,000+ [data science packages](#)

**Package Management**  
Manage packages, dependencies and environments with [conda](#)

**Portal to Data Science**  
Uncover insights in your data and create interactive visualizations

Windows   macOS   Linux



# Install Anaconda and Jupyter

## 3. ทำการติดตั้ง Python 3.6 โดยคลิกที่ Download

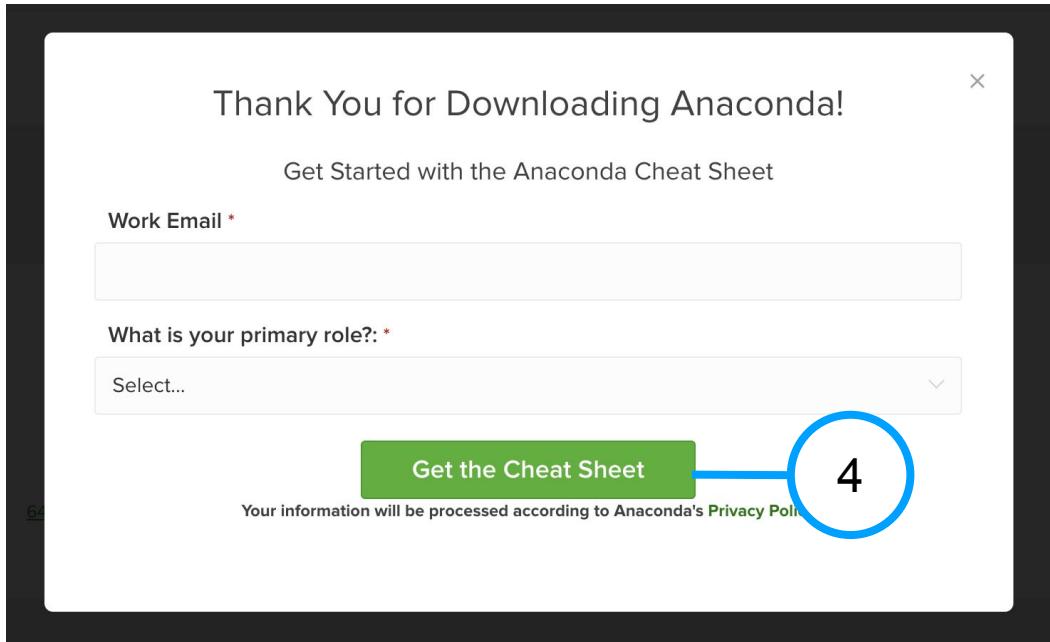
The screenshot shows the download page for the Anaconda 5.2 For macOS Installer. It features three download options:

- Python 3.6 version \***:
  - Download** (highlighted with a blue box)
  - [64-Bit Graphical Installer \(613 MB\)](#) (with a question mark icon)
  - [64-Bit Command-Line Installer \(523 MB\)](#) (with a question mark icon)
- Python 2.7 version \***:
  - Download**
  - [64-Bit Graphical Installer \(617 MB\)](#) (with a question mark icon)
  - [64-Bit Command-Line Installer \(527 MB\)](#) (with a question mark icon)



# Install Anaconda and Jupyter

4. หลังจากคลิกจะมี Popup ขึ้นมาขอข้อมูล เพื่อแลกกับ Cheat Sheet



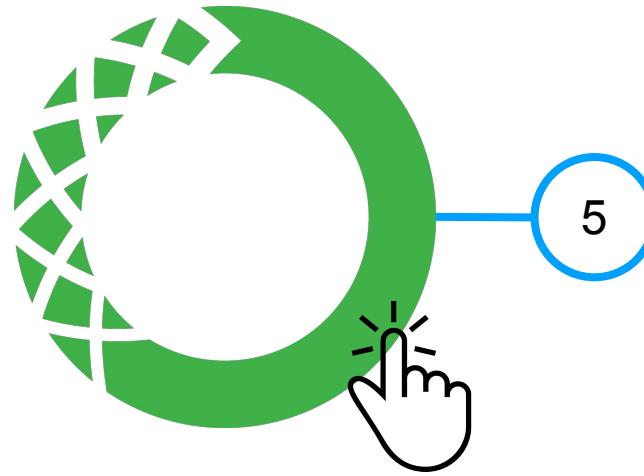
## Cheat Sheet

เปรียบเสมือนคู่มือที่รวมวารุธ์ของโปรแกรมนั้นๆ หรือคำสั่งการใช้งานต่างๆ โดยที่เราไม่ต้องไปนั่งมหาเอง



# Install Anaconda and Jupyter

5. ทำการคลิกติดตั้งตัวโปรแกรมที่ทำการโหลดมา แล้วจะได้ Icon ดังรูปด้านล่าง



Anaconda-Navigator



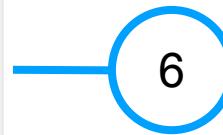
# Install Anaconda and Jupyter

## 6. หลังเข้ามาที่ Anaconda จะพบหน้าตาโปรแกรมอื่นๆ ประมาณนี้

The screenshot shows the Anaconda Navigator application interface. On the left is a sidebar with links for Home, Environments, Learning, Community, Documentation, Developer Blog, and Feedback. The main area displays a grid of application icons:

- lab** (jupyterlab): An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook Architecture. Version 0.31.12. Buttons: Launch, Install.
- jupyter** (notebook): Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. Version 5.4.1. Buttons: Launch, Install.
- IPy** (qtconsole): PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. Version 4.3.1. Buttons: Launch, Install.
- R** (rstudio): A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks. Version 1.1.423. Buttons: Launch, Install.
- spyder**: Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. Version 3.2.8. Buttons: Launch, Install.
- glueviz**: Multidimensional data visualization across files. Explore relationships within and among related datasets. Version 0.13.3. Buttons: Install.
- orange3**: Component-based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. Version 3.13.0. Buttons: Install.
- vscode**: Streamlined code editor with support for development operations like debugging, task running and version control. Version 1.25.1. Buttons: Install.

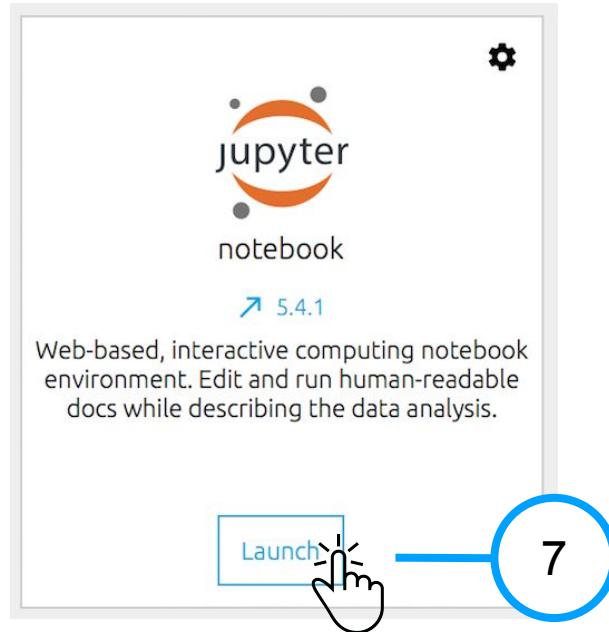
At the top right are buttons for "Upgrade Now" and "Sign in to Anaconda Cloud".





# Install Anaconda and Jupyter

7. คลิกที่ปุ่ม Launch เพื่อเริ่มใช้โปรแกรม Jupyter





# Install Anaconda and Jupyter

8. หลังจากเข้ามายังจอหน้าจอ terminal ไม่ต้องตกใจไป เป็นการรันโปรแกรม Python เนื่องหลังผ่านทาง Jupyter Notebook อีกที โดยที่จะบอกถึงวิธีการเข้า Jupyter ผ่าน URL ด้านล่าง

```
redthegx — /Users/redthegx/anaconda3/envs/tf-thesis/bin/jupyter_mac.command; exit — /Users/redthegx/anaconda3/env...
Last login: Mon Aug 13 09:45:52 on console
/Users/redthegx/anaconda3/envs/tf-thesis/bin/jupyter_mac.command ; exit;
[redthegx@x86_64-apple-darwin13 ~]$ /Users/redthegx/anaconda3/envs/tf-thesis/bin/jupyter_mac.command ; exit;
[I 22:52:47.448 NotebookApp] Serving notebooks from local directory: /Users/redthegx
[I 22:52:47.448 NotebookApp] 0 active kernels
[I 22:52:47.448 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=a707e053383c8e899693e05
1f91173a10847fbe0cf868e39
[I 22:52:47.448 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation)
[C 22:52:47.452 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=a707e053383c8e899693e051f91173a10847fbe0cf868e39
[I 22:52:47.638 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

8



# Install Anaconda and Jupyter

8

The screenshot shows the Jupyter Notebook interface at `localhost:8888/tree`. On the left, there's a sidebar with a list of files and folders. A blue circle labeled '9' highlights this sidebar. In the main area, there's a list of files with their last modified times. A blue circle labeled '11' points to the 'Upload' button at the top right of the list, and another blue circle labeled '12' points to the 'New' dropdown menu.

| Name                               | Last Modified |
|------------------------------------|---------------|
| 180630_Data_in_Action_AETAS        | 2 months ago  |
| anaconda3                          | 4 months ago  |
| Applications                       | 11 days ago   |
| checkpoint                         | an hour ago   |
| Desktop                            | 4 months ago  |
| Documents                          | 19 days ago   |
| Downloads                          | 7 months ago  |
| Ex_Files_Data_Science_of_Marketing | 3 months ago  |
| Ex_Files_Intro_Analysis_Python     | 2 months ago  |
| Ex_Files_Learning_Python           | 3 months ago  |
| Ex_Files_OpenCV_Python_Dev         | 2 months ago  |
| Ex_Files_Python_Data_Science_EssT  | 3 months ago  |
| Ex_Files_python_dpatters           | a year ago    |
| Ex_Files_Tableau10_DataSci         | 4 months ago  |
| FaceLearning                       | a month ago   |
| go-project                         | 9 days ago    |
| Google TTS                         | 2 months ago  |
| IdeaProjects                       | 5 months ago  |
| logs                               |               |

## 9. หน้าตาโปรแกรม Jupyter

Notebook หลังกด Launch จะทำการ  
ขึ้นให้อัตโนมัติ

## 10. ส่วนของ Folder

## 11. Upload ไว้สำหรับนำไฟล์เข้าไป

## 12. New ทำการสร้าง Folder ใหม่ หรือทำการสร้าง Sheet สำหรับเขียน ภาษา Python

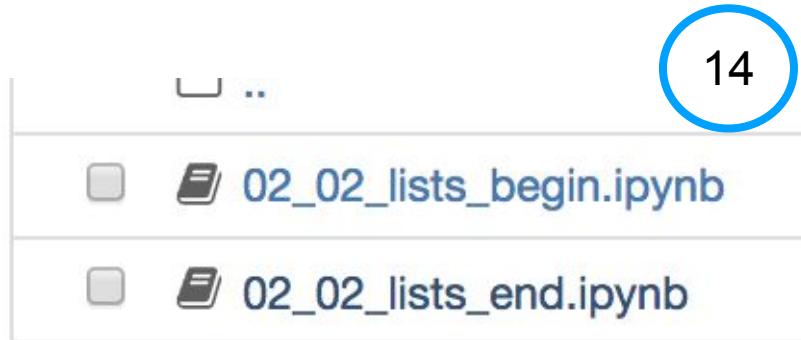


# Install Anaconda and Jupyter

13. ทำการค้นหา Folder “ML\_for\_your\_DS\_with\_Python”

14. หลังจากนั้นก็กดเข้าไปทีละขั้นแล้วลุยไปด้วยกันได้เลย !!

14



ไฟล์ .ipynb สำหรับรันภาษา Python  
ผ่าน Jupyter Notebook

## Interactive Development Environment

ส่วนของหน้าจอที่ให้เราทำการเขียนโค้ดเพื่อทำการประมวลผลได้ ในที่นี้เราจะใช้ Jupyter Notebook เป็น IDLE เพื่อทำการเขียนโค้ดผ่านทีละกรอบ ๆ แทนหน้าจอยังเป็นมิตร ดีกว่าต้องมาเขียนผ่าน Terminal หน้าจอสีดำๆ คงลำบากແຍ່ເລຍ

```
In [2]: nephews = ["Huey", "Dewey", "Louie"]

In [3]: nephews
Out[3]: ['Huey', 'Dewey', 'Louie']

In [4]: nephews[0]
Out[4]: 'Huey'

In [5]: nephews[2]
Out[5]: 'Louie'

In [6]: len(nephews)
Out[6]: 3
```

Jupyter Notebook

```
In [1]: jupyter      %1
In [2]: python (python3.6) %2
Last login: Fri Aug 17 22:54:29 on ttys000
[redtheGx@RedTheGx-MacBook-Pro ~ ]$ python
Python 3.6.5 |Anaconda custom (64-bit)| (default, Mar 29 2018, 13:14:23)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Terminal



# Morning

---

Day-01



Boyd BigData RPG - Sorratat Sirirattanajakarin

1. Introduction to Data Science
2. Install Anaconda and Jupyter
3. **Python Programming Basics**
  1. Operation
  2. Variable
  3. Flow Control
  4. Condition ( If, For, While )
  5. Function, Method
  6. List, Tuple
  7. Dictionaries and Structural Data
  8. Manipulating Strings



# Python Programming Basics

---

1. Operation
2. Variable
3. Flow Control
4. Condition ( If, For, While )
5. Function, Method
6. List, Tuple
7. Dictionaries and Structural Data
8. Manipulating Strings



# Operation

| Operator | Operation       | Example   | Result |
|----------|-----------------|-----------|--------|
| **       | ยกกำลัง         | $2 ** 4$  | 8      |
| %        | เก็บเศษที่เหลือ | $24 \% 5$ | 4      |
| //       | หารแบบไม่เอาเศษ | $24 // 5$ | 4      |
| /        | หาร             | $24 / 5$  | 4.8    |
| *        | คูณ             | $2 * 5$   | 10     |
| -        | ลบ              | $5 - 2$   | 3      |
| +        | บวก             | $2 + 5$   | 8      |

# Code Session

In คือส่วนของการ Input Code ของเราเข้าไปทำงาน

Out คือส่วนผลลัพธ์ที่เราทำการสั่งทำงาน

```
In [2]: 2 + 3 * 6
```

```
Out[2]: 20
```

```
In [3]: (2 + 3) * 6
```

```
Out[3]: 30
```

```
In [4]: 48565878 * 578453
```

```
Out[4]: 28093077826734
```

```
In [5]: 2 ** 8
```

```
Out[5]: 256
```

```
In [6]: 23 / 7
```

```
Out[6]: 3.2857142857142856
```

```
In [7]: 23 // 7
```

```
Out[7]: 3
```

```
In [9]: 23 % 7
```

```
Out[9]: 2
```

```
In [10]: 2 + 2
```

```
Out[10]: 4
```

```
In [11]: (5 - 1) * ((7 + 1) / (3 - 1))
```

```
Out[11]: 16.0
```

# ERROR

TIP

คือส่วนหนึ่งของชีวิต ดังนั้นไม่ต้องไปกลัวมัน และ Error ใน Python ก็เข้าใจได้ไม่ยาก เพราะมันจะบอกเราเลยว่า Error อะไรและที่บรรทัดไหนกันแน่ !!



Error ในชีวิตจริง

```
In [1]: 10 +
      File "<ipython-input-1-e6d1cf46367f>", line 1
          10 +
              ^
SyntaxError: invalid syntax

In [2]: 40 + 5 + * 2
      File "<ipython-input-2-ef6a7ba9d20f>", line 1
          40 + 5 + * 2
              ^
SyntaxError: invalid syntax
```

Error ในPython





# Operation

ลำดับการทำงานของ Operation ໄລ່ທຳຈາກງານເລີບໃຫ້ເສື້ອຈຸດ່ອນ

**Start**



$$(5 - 1) * ((7 + 1) / (3 - 1))$$

$$4 * ((7 + 1) / (3 - 1))$$

$$4 * (8) / (3 - 1)$$

$$4 * (8) / (2)$$

$$4 * 4.0$$

$$16.0$$



**Result**



# Data types

---

| Data Type             | Examples                                  |
|-----------------------|---|
| Integer               | -2, -1, 0, 1, 2, 3, 4, 5                  |
| Floating-point number | -1.25, -1.0, -0.5, 0.0, 0.5, 1.0, 1.25    |
| Strings               | 'ab', 'ฉัน', 'เธอ', 'Hello!', '25 เรื่อง' |

# String

TIP

สำหรับ String เป็น Type ของข้อมูลรูปแบบหนึ่ง การจะบอกว่าสิ่งที่คือ String นั้น จำเป็นต้องใช้ลักษณ์ Single Quote ("") หรือ Double Quote ก็ได้ ("")

Single Quote

'ไม่พูดเยอะเจ็บคอ !!!'

Double Quote

"ผมชอบปิดทีวีตอนเย็นวันศุกร์"



# Comment

TIP

สำหรับการเขียนโค้ด หากเราเขียนได้แล้ว เรายังคงอยากระบุว่าต้องการให้บันทึกไว้ข้างๆ โค้ดที่เราเขียนเอง เห็นแล้วคงอุ่นใจกว่าที่จะต้องมาคอยัดคนละที่ จริงไหมครับ !! โดยโค้ดนั้นมี 2 แบบ โดยที่ระบบจะไม่นำโค้ดมาทำการคำนวณ

```
In [1]: 1 + 1 # Code สำหรับการบวก  
Out[1]: 2
```

```
In [2]: """  
ต่อไปนี้จะทำการเขียนโค้ดสำหรับการ หารเท่านั้น  
"""  
  
Out[2]: '\nต่อไปนี้จะทำการเขียนโค้ดสำหรับการ หารเท่านั้น\n'
```

Comment โดยใช้ # หรือ Sharp

Comment โดยใช้ """" """"  
หรือ Triple Double Quote



# Morning

---

Day-01



Boyd BigData RPG - Sorratat Sirirattanajakarin

1. Introduction to Data Science
2. Install Anaconda and Jupyter
3. **Python Programming Basics**
  1. Operation
  2. **Variable**
  3. Flow Control
  4. Condition ( If, For, While )
  5. Function, Method
  6. List, Tuple
  7. Dictionaries and Structural Data
  8. Manipulating Strings

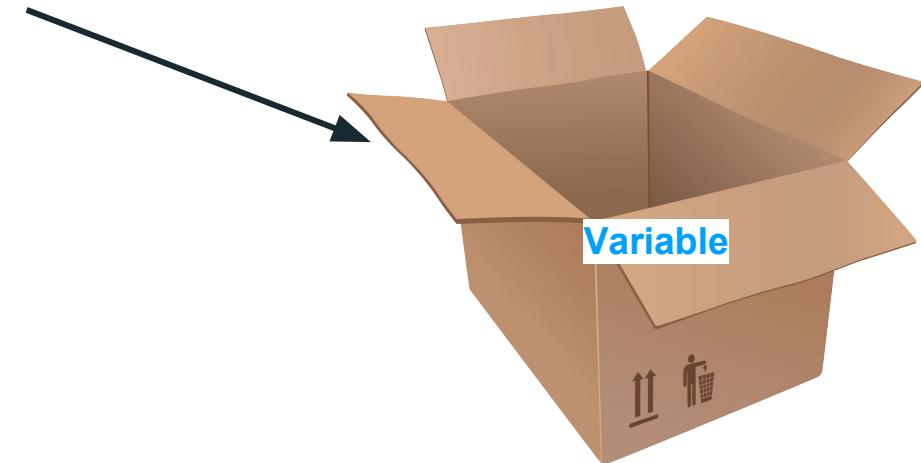


# Variable

การสร้างตัวแปรมารับค่าบางอย่างไป

## "ผนชอนปิดทีวีตอนเย็นวันศุกร์"

ค่าที่เราต้องการเก็บ



ส่งไปเก็บที่กล่องซื้อตัวแปร

bin



# Variable

หากเราต้องการเรียกใช้งานค่าดังกล่าว ก็ไม่ต้องพิมพ์ซ้ำใหม่อีก ดีไหมล่ะ  
ก็แค่พิมพ์ชื่อคล่องที่เราเก็บไว้ มันก็จะปล่อยผลลัพธ์ออกมา



**"ผู้ช่วยปิดทีวีตอนเย็นวันศุกร์"**

ผลลัพธ์ของการเรียกชื่อตัวแปร

**bin**



# Variable

ตัวแปรสามารถรับค่าเก็บได้เพียงค่าเดียว ดังนั้นหากทำการโยนของใหม่เข้าไป  
ของเก่าก็จะถูกเท ออกจากกล่อง และเปลี่ยนค่าที่เก็บเป็นตัวใหม่ล่าสุดแทน

"ไม่พูดเยอะเจ็บคอ !!!"

ค่าที่ถูกนำเข้าไปเก็บใหม่

"ผนชובปิดทีวิตอนเย็นวันศุกร์"

ค่าที่ถูกเอาออก หมดความหมาย



bin

# Name the Variable

| Valid variable names | Invalid variable names | Reasons                           |
|----------------------|------------------------|-----------------------------------|
| -                    | --                     | ไม่สามารถใช้ hyphen ได้           |
| ตารางส่วนตัว         | 112hawaiian            | ใช้ตัวเลขนำหน้าไม่ได้             |
| bin                  | def                    | ใช้ชื่อเฉพาะไม่ได้ ตรงกับฟังก์ชัน |
| df_common            | df common              | เว้นวรรคชื่อไม่ได้                |
| bacon                | 44                     | ใช้ตัวเลขไม่ได้                   |
| pizza                | piz\$a                 | ใช้สัญลักษณ์พิเศษไม่ได้           |
| _pizza               | '_pizza'               | ใช้ String ไม่ได้                 |



# Code Session

สร้างกล่องชื่อ bin ນารับค่า

ทำการแสดงผลตัวแปรชื่อ bin

ส่งค่าใหม่ให้ตัวแปรเดิมที่ชื่อ bin

ทำการแสดงผลตัวแปรชื่อ bin



```
In [1]: bin = "ผมชอบปิดทีวีตอนเย็นวันศุกร์"
```

```
In [2]: bin
```

```
Out[2]: ' ผมชอบปิดทีวีตอนเย็นวันศุกร์ '
```

```
In [3]: bin = "ไม่พูด เยอะ เจ็บคอ !! "
```

```
In [4]: bin
```

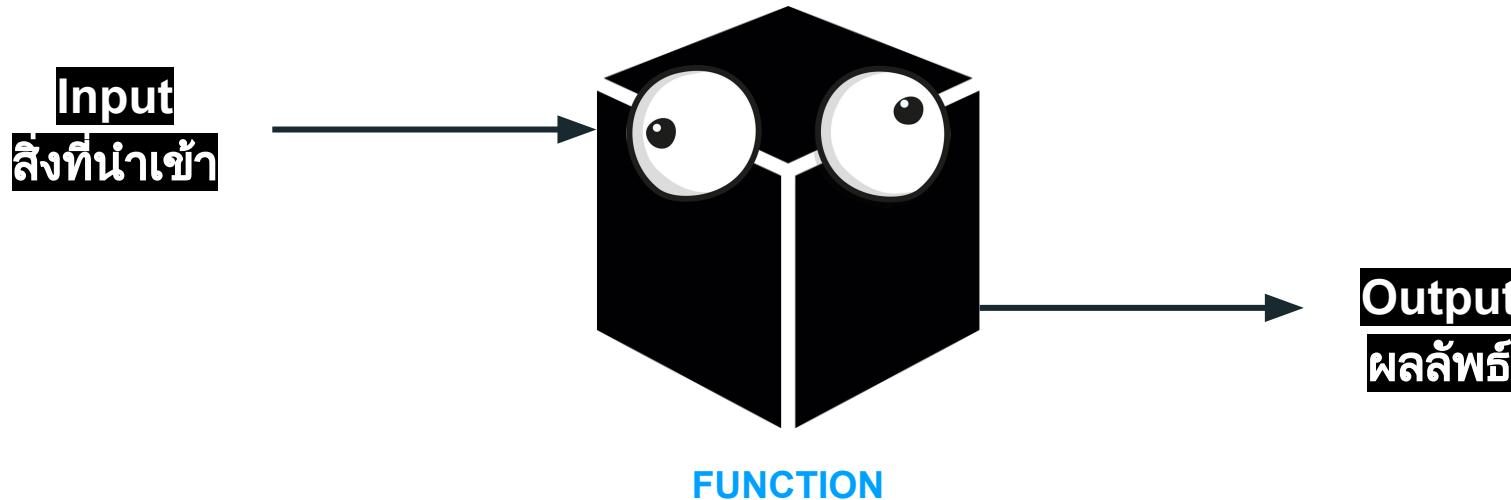
```
Out[4]: ' ไม่พูด เยอะ เจ็บคอ !! '
```

```
In [ ]:
```





# Function



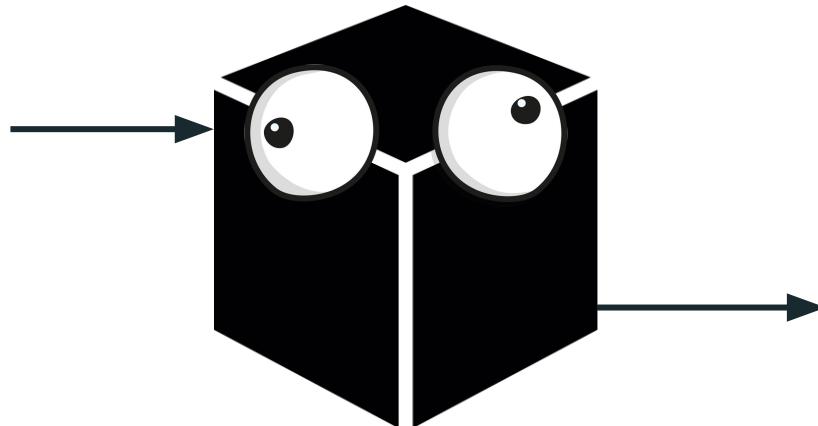
โดยปกติ 1 Function จะทำงานเฉพาะ 1 อย่าง  
เปรียบเสมือนโปรแกรมประดิษฐ์  
มีกลไกบางอย่างเพื่อการเปลี่ยนรูปหรือคำนวณเป็นต้น



# Function

ตัวอย่าง Function คำนวณ BMI ด้วยไมโครเกลียร์

รับ 2 ค่า  
1. ค่าน้ำหนัก  
2. ค่าส่วนสูง



INPUT

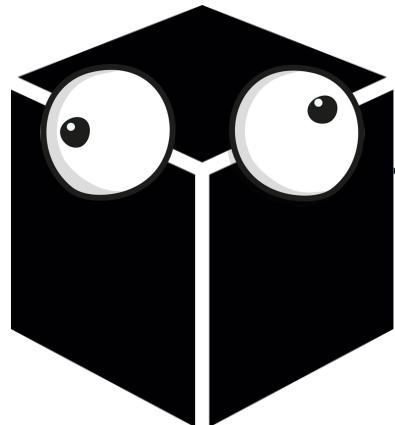
FUNCTION BMI

OUTPUT

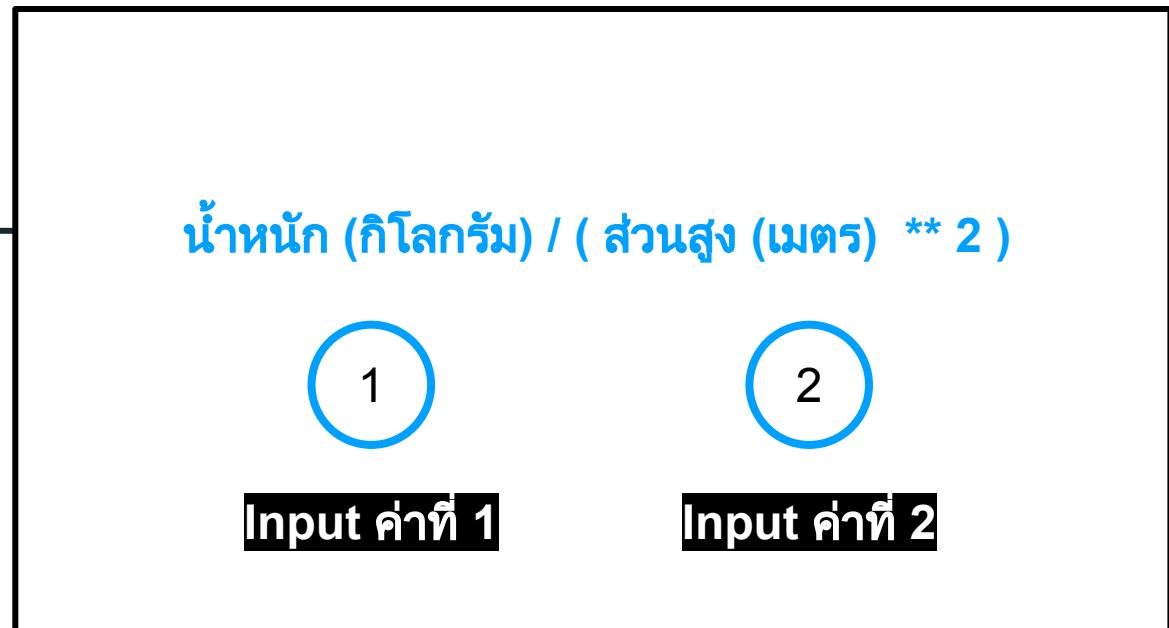


# Function

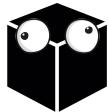
มาดูภายใน Function คำนวณ BMI ดั้งนีมวกลาย กันดีกว่า !?



FUNCTION BMI



# Function



โครงสร้างการเขียน Function

ประกาศเริ่มฟังก์ชันทุกรอบ  
ต้องเขียนด้วย def

ประกาศชื่อฟังก์ชัน

ประกาศตัวแปรที่รับเข้าฟังก์ชัน

```
def function(v1, v2):  
    result = v1 / ( v2 ** 2 )  
    return result
```

ส่วนการทำงานของ  
ฟังก์ชัน

ผลลัพธ์ที่ทำการส่งออกมานะ

ใส return เพื่อทำการส่ง  
ผลลัพธ์ออกจากฟังก์ชัน

# Code Session

ทำการสร้าง พิ้งก์ชัน ขึ้นมาสำหรับคำนวณ BMI

ทำการตั้งค่าให้กับตัวแปรที่จะใส่เข้าใน พิ้งก์ชัน

ทำการใช้งาน พิ้งก์ชัน

ผลลัพธ์ของ พิ้งก์ชัน

```
In [1]: def BMICalculator(weight, height):  
    bmi = weight / (height ** 2)  
    return bmi
```

```
In [2]: weight = 75  
        height = 1.8
```

```
In [3]: BMICalculator(weight, height)
```

```
Out[3]: 23.148148148148145
```



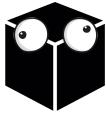


# Function

Function พื้นฐานที่เรียกใช้งานได้เลย

| Function | Result  |
|----------|---|
| print()  | ทำการดูค่าตัวแปร หรือค่าที่ส่งเข้ามาในฟังก์ชัน    |
| input()  | ทำการใส่ข้อมูลเข้าไปในฟังก์ชัน                    |
| len()    | ทำการนับจำนวนตัวอักษรของค่าที่ใส่เข้ามาในฟังก์ชัน |
| str()    | ทำการแปลงค่าข้อมูลให้อยู่ในชนิดของ String         |
| int()    | ทำการแปลงค่าข้อมูลให้อยู่ในชนิดของ Int            |
| float()  | ทำการแปลงค่าข้อมูลให้อยู่ในชนิดของ Float          |

# Function Print



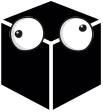
**INPUT**

```
print('Hello Prayud!')
```

**OUTPUT**

```
Hello Prayud!
```

# Function Input



**INPUT**

```
input('กรุณาใส่จำนวนนาฬิกา')
```

**OUTPUT**

กรุณาใส่จำนวนนาฬิกา

“ช่องสำหรับใส่ข้อมูล เพื่อทำการแสดงผลต่อ”

# Function Len



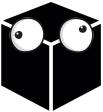
**INPUT**

```
len('Hello Prayud!')
```

**OUTPUT**

```
13
```

# Function String



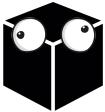
**INPUT**

**str(25)**

**OUTPUT**

**'25'**

# Function Int



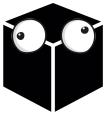
INPUT

`int(24.99)`

OUTPUT

`24`

# Function Float



**INPUT**

**float(25)**

**OUTPUT**

**25.0**

# Code Session

ผลลัพธ์ print ฟังก์ชัน

ผลลัพธ์ input ฟังก์ชัน

ผลลัพธ์ len ฟังก์ชัน

ผลลัพธ์ str ฟังก์ชัน

ผลลัพธ์ int ฟังก์ชัน

ผลลัพธ์ float ฟังก์ชัน

```
In [1]: print('Hello Prayud!')
```

Hello Prayud!

```
In [3]: input("กรุณาใส่จำนวนนาพิกา")
```

กรุณาใส่จำนวนนาพิกา25

```
Out[3]: '25'
```

```
In [5]: len('Hello Prayud!')
```

```
Out[5]: 13
```

```
In [7]: str(25)
```

```
Out[7]: '25'
```

```
In [8]: int(24.99)
```

```
Out[8]: 24
```

```
In [9]: float(25)
```

```
Out[9]: 25.0
```

# My First Program

---



เปิด Jupyter Notebook ไปที่ไฟล์

**python101-01-07-myFirstProgram.ipynb**

# Evaluation Step Function

ลำดับการแสดงผลของฟังก์ชัน และการทำงาน

## Step

```
print('อายุคุณจะ ' + str(int(myAge) + 1) + ' ในปีนี้ ถ้าคุณยังไม่ถึงวันเกิดมาก่อนในปีเสียก่อนนะ !!')
print('อายุคุณจะ ' + str(int('60') + 1) + ' ในปีนี้ ถ้าคุณยังไม่ถึงวันเกิดมาก่อนในปีเสียก่อนนะ !!')
print('อายุคุณจะ ' + str(60+1) + ' ในปีนี้ ถ้าคุณยังไม่ถึงวันเกิดมาก่อนในปีเสียก่อนนะ !!')
print('อายุคุณจะ ' + str(61) + ' ในปีนี้ ถ้าคุณยังไม่ถึงวันเกิดมาก่อนในปีเสียก่อนนะ !!')
print('อายุคุณจะ ' + '61' + ' ในปีนี้ ถ้าคุณยังไม่ถึงวันเกิดมาก่อนในปีเสียก่อนนะ !!')
print('อายุคุณจะ 61' + ' ในปีนี้ ถ้าคุณยังไม่ถึงวันเกิดมาก่อนในปีเสียก่อนนะ !!')
print('อายุคุณจะ 61 ในปีนี้ ถ้าคุณยังไม่ถึงวันเกิดมาก่อนในปีเสียก่อนนะ !!')
```

## Output

อายุคุณจะ 61 ในปีนี้ ถ้าคุณยังไม่ถึงวันเกิดมาก่อนในปีเสียก่อนนะ !!



# Homework Session 1

1. ข้อใดคือ Operator และข้อใดคือ Value

- a. \*
- b. 'hello'
- c. -88.8
- d. -
- e. /
- f. +
- g. 5

2. ข้อใดคือ Variable และข้อใดคือ String

- a. spam
- b. 'spam'

3. ลองบอกชื่อชนิดของข้อมูลมา สาม ชนิด



# Homework Session 1

3. หากทำการรันโค้ดหลังจากนี้ จะได้ผลลัพธ์เท่ากับเท่าไหร่

```
watch = 24
```

```
watch = watch + 1
```

4. ทำไมโค้ดชุดนี้ถึงรันแล้วเกิด Error

```
'ผมยืมนาฬิกาเพื่อนมา ' + 99 + ' เรือน.'
```



# Morning

---

Day-01

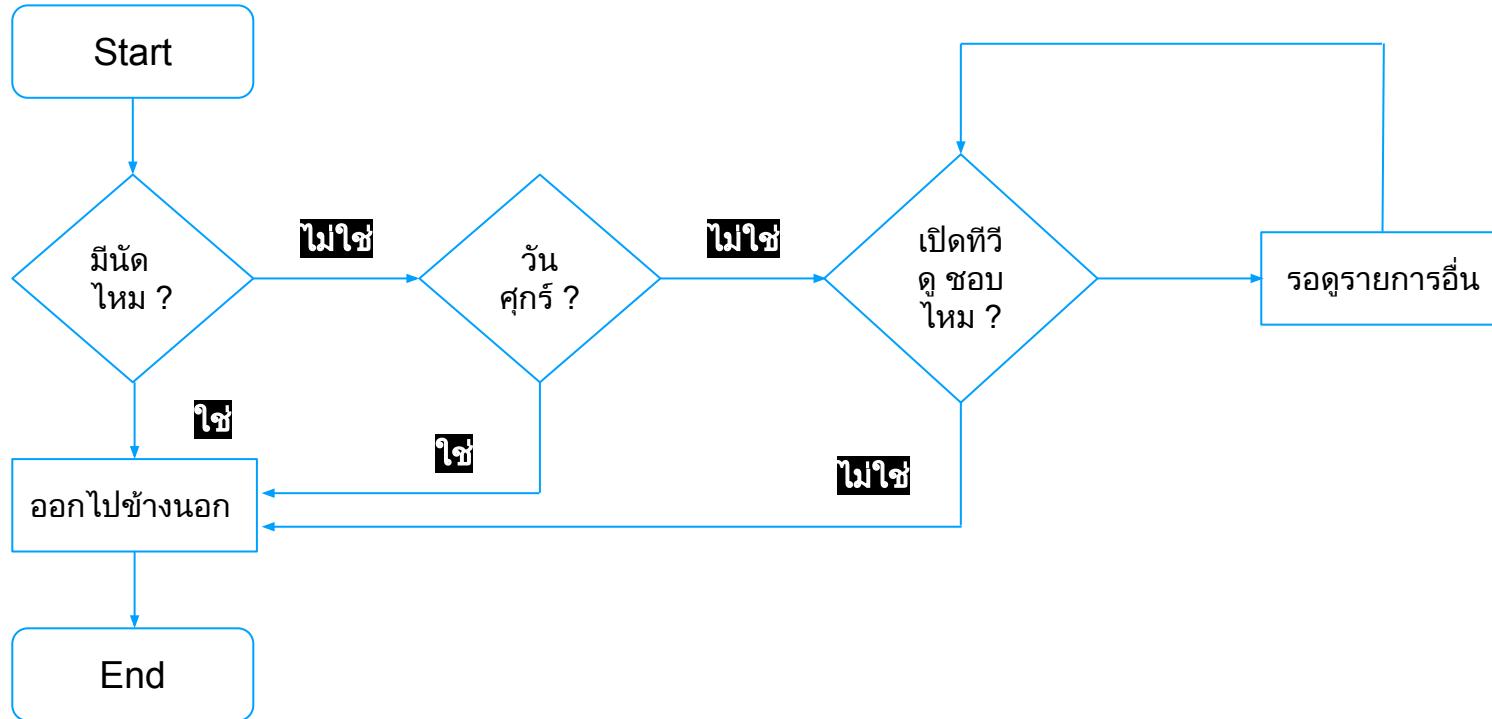


Boyd BigData RPG - Sorratat Sirirattanajakarin

1. Introduction to Data Science
2. Install Anaconda and Jupyter
3. **Python Programming Basics**
  1. Operation
  2. Variable
  3. **Flow Control**
  4. Condition ( If, For, While )
  5. Function, Method
  6. List, Tuple
  7. Dictionaries and Structural Data
  8. Manipulating Strings



# Flow Control





# Boolean Values

---

True

มีค่าเท่ากับ 1

False

มีค่าเท่ากับ 0

# The Difference Between == and =

TIP

ความแตกต่างระหว่างเครื่องหมาย == และ =

=

การ Assign ค่า

==

การเปรียบเทียบค่า





# Comparison Operator

เครื่องหมายการเปรียบเทียบ

| Operator           | Meaning             |
|--------------------|---------------------|
| <code>==</code>    | เทียบเท่ากัน        |
| <code>!=</code>    | ไม่เท่ากัน          |
| <code>&lt;</code>  | น้อยกว่า            |
| <code>&gt;</code>  | มากกว่า             |
| <code>&lt;=</code> | น้อยกว่าหรือเท่ากัน |
| <code>&gt;=</code> | มากกว่าหรือเท่ากัน  |

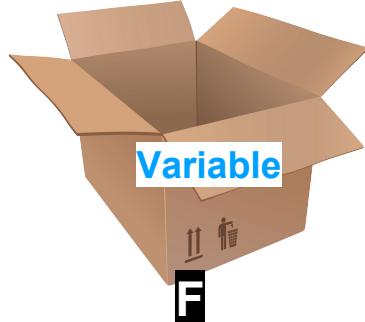


# Comparison Operator

ก่อนอื่นเลย ให้เรา Assign ค่าให้ T และ F ดังภาพ



"True"



"False"



# And

---

| Variable A | Operator | Variable B | Result |
|------------|----------|------------|--------|
| T          | and      | T          | True   |
| T          | and      | F          | False  |
| F          | and      | T          | False  |
| F          | and      | F          | False  |



# Or

---

| Variable A | Operator | Variable B | Result |
|------------|----------|------------|--------|
| T          | or       | T          | True   |
| T          | or       | F          | True   |
| F          | or       | T          | True   |
| F          | or       | F          | False  |



# Not

---

| Not | Variable A | Result |
|-----|------------|--------|
| not | T          | False  |
| not | F          | True   |



# Mixing Boolean and comparison operators

| Values A     | Operator | Values A    | Result |
|--------------|----------|-------------|--------|
| ( 44 > 2 )   | ==       | ( 46 - 0 )  | False  |
| ( 44 < 4 )   | ==       | ( 40 - 4 )  | False  |
| ( 10 != 20 ) | and      | ( 44 == 2 ) | False  |
| ( 1 == 2 )   | !=       | ( 2 == 1 )  | False  |

# Code Session

String ที่ไม่ได้มีความเท่ากัน ผลลัพธ์คือ False

String ที่ไม่ได้มีความเท่ากัน ผลลัพธ์คือ True

```
In [15]: 'hello' == 'สวัสดี'
```

```
Out[15]: False
```

```
In [16]: 'dog' != 'cat'
```

```
Out[16]: True
```

```
In [17]: 'ทั้มปี' != 'ประหยัด'
```

```
Out[17]: True
```

```
In [18]: T != F
```

```
Out[18]: True
```

```
In [19]: 42 == 42.0
```

```
Out[19]: True
```

```
In [20]: 44 == '44.0'
```

```
Out[20]: False
```

```
In [23]: จำนวนไข่ = 42  
จำนวนไข่ <= 42
```

```
Out[23]: True
```

```
In [24]: จำนวนนาฬิกา = 29  
จำนวนนาฬิกา >= 10
```

```
Out[24]: True
```



# Morning

---

Day-01



Boyd BigData RPG - Sorratat Sirirattanajakarin

1. Introduction to Data Science
2. Install Anaconda and Jupyter
3. **Python Programming Basics**
  1. Operation
  2. Variable
  3. Flow Control
  4. **Condition ( If, For, While )**
  5. Function, Method
  6. List, Tuple
  7. Dictionaries and Structural Data
  8. Manipulating Strings

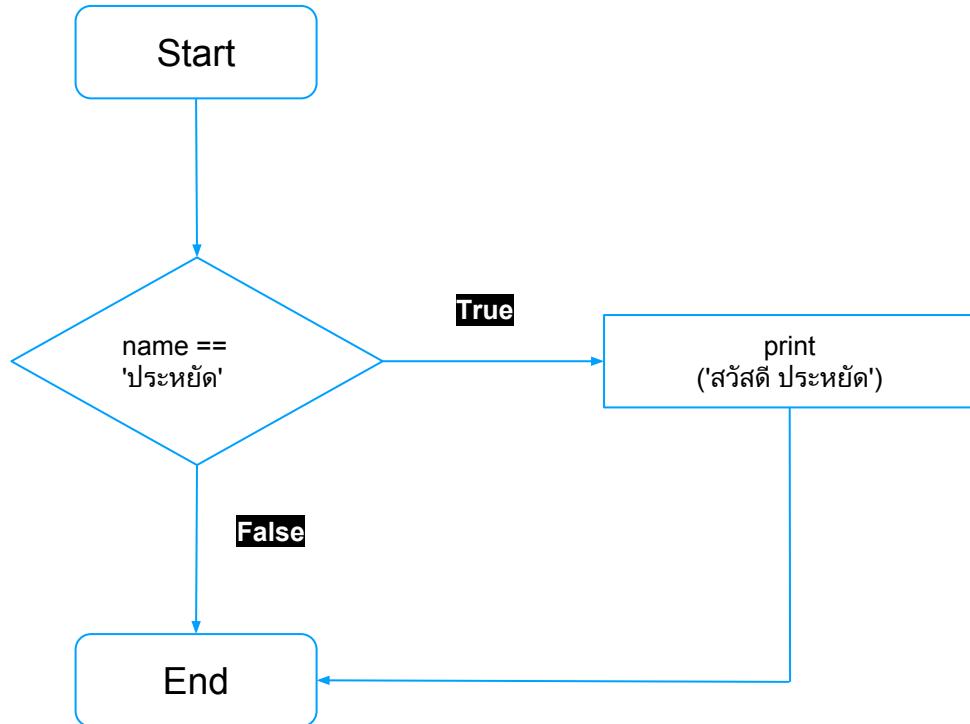


# Condition

การสร้างเงื่อนไขจาก Boolean ว่าถ้าเกิด True ให้ทำอะไรต่อ และถ้าเกิด False ให้ทำอะไรต่อ โดยเจ้าเงื่อนไขนี้เองที่เราเอาไปสร้าง Flow Diagram



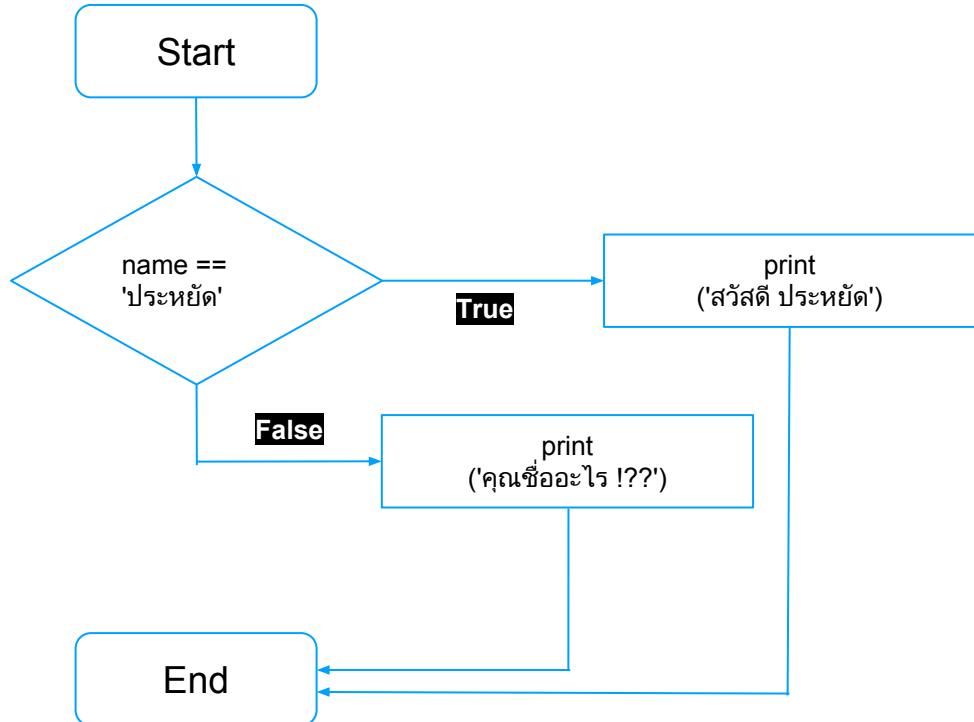
# IF



```
if name == 'ประ hayd':  
    print('สวัสดี ประ hayd')
```



# Else



**if name == 'ประ hayd':**

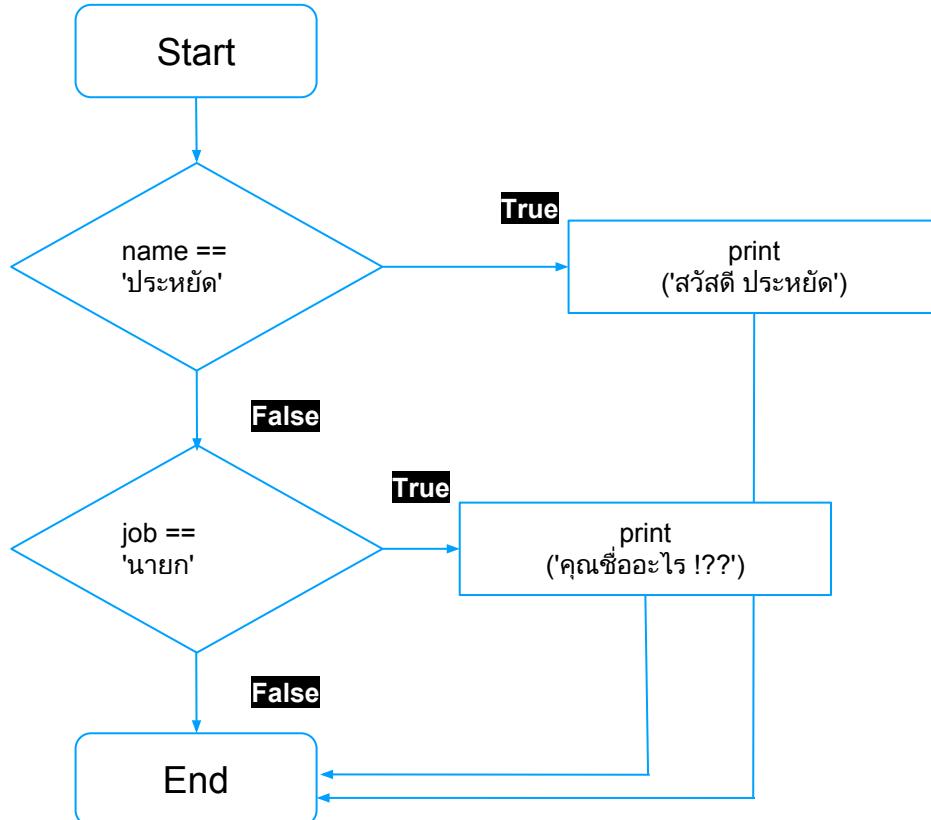
**print('สวัสดี ประ hayd')**

**else:**

**print('คุณชื่ออะไร !??')**



# Elif



**if name == 'ประ hayd':**

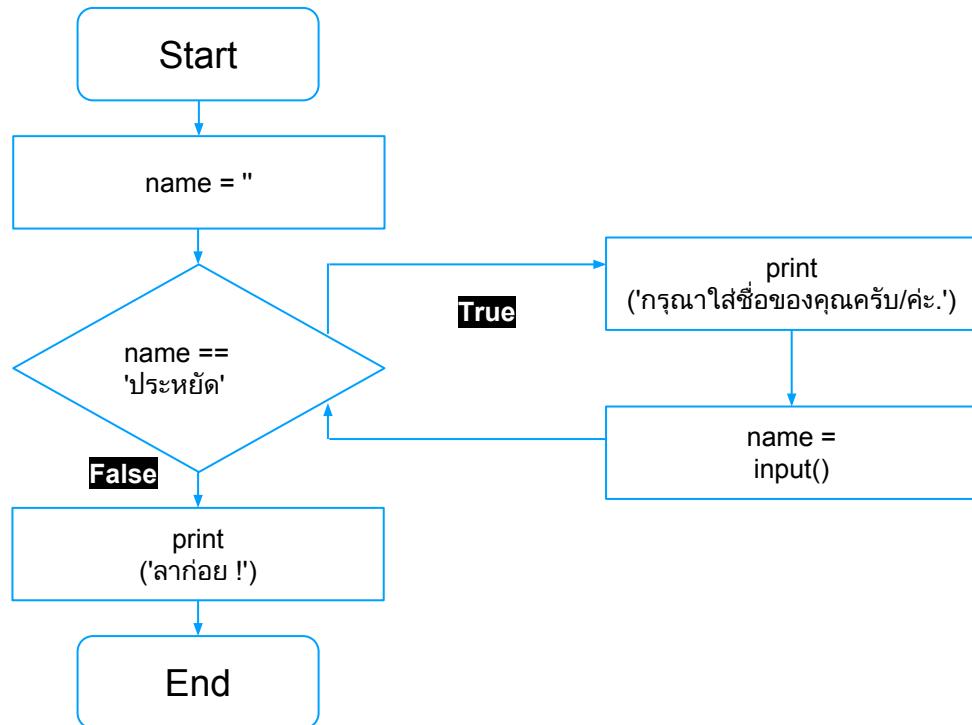
**print('สวัสดี ประ hayd')**

**elif Job == 'นายก':**

**print('คุณชื่ออะไร !??')**



# While loop



**name = "**

**while name != 'ประ hayd':**

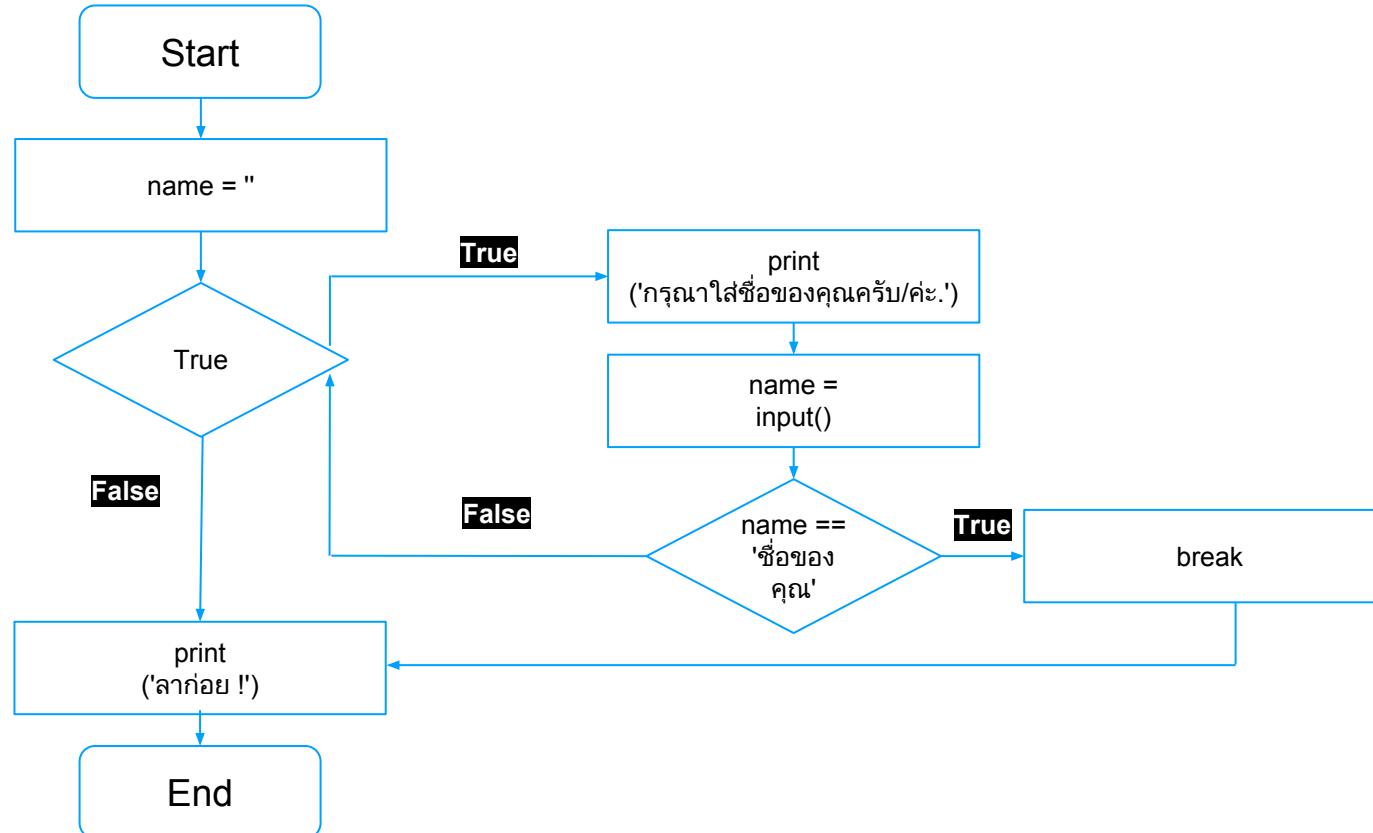
**print('กรุณาใส่ชื่อของคุณครับ/ค่ะ.')**

**name = input()**

**print('ลาก่อน!')**



# Break





# Break

---

```
while True:
```

```
    print('ลองใส่ชื่อคุณดูสิ')
```

```
    name = input()
```

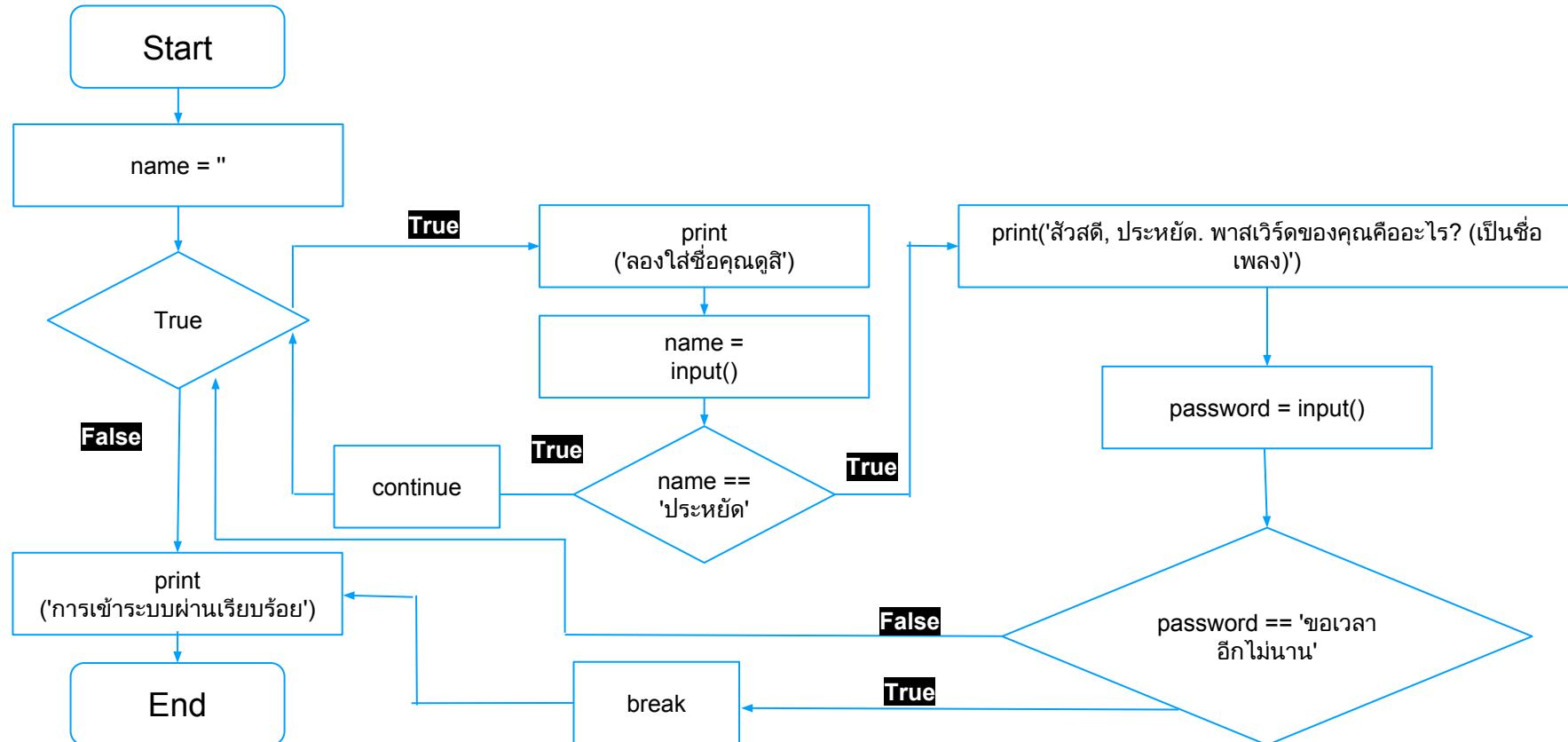
```
    if name == 'ชื่อของคุณ':
```

```
        break
```

```
    print('ยินดีที่ได้รู้จักนะ !')
```



# Continue





# Continue

---

**while True:**

**print('ลองใส่ชื่อคุณดูสิ')**

**name = input()**

**if name != 'ปะหยัด':**

**continue**

**print('สวัสดี, ปะหยัด. พาสเวิร์ดของคุณคืออะไร? (เป็นชื่อเพลง)')**

**password = input()**

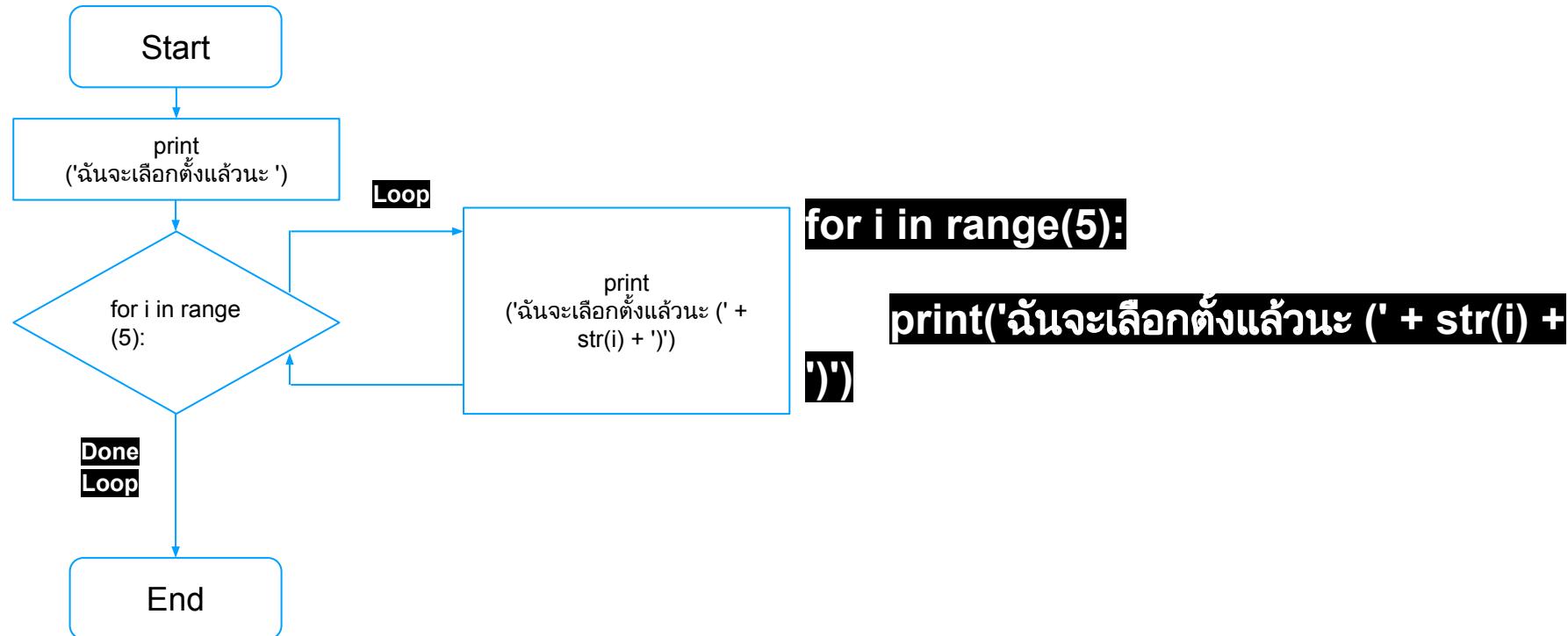
**if password == 'ขอเวลาอีกไม่นาน':**

**break**

**print('การเข้าระบบผ่านเรียบร้อย')**



# For loop





# For loop

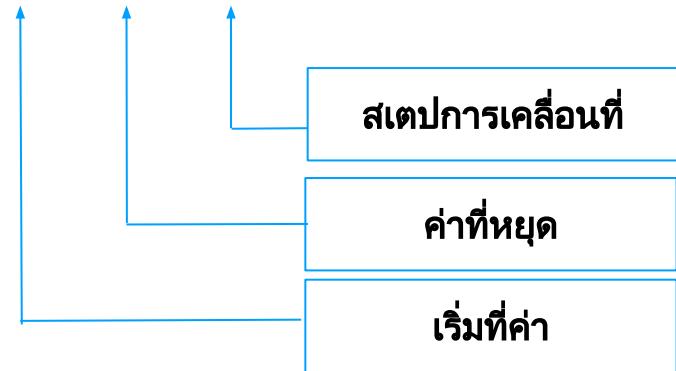
INPUT

```
for i in range(0,10,2):
```

```
    print('i')
```

OUTPUT

```
0,2,4,6,8
```



# Code Session

สร้างกล่องขึ้นมา รับค่าชนิดของ String

ใช้ If Condition

ใช้ If + else Condition

ใช้ If + elif Condition

ใช้ While loop หากไม่เท่ากับ 'ประหยด' ออก loop

'prayad' != 'ประหยด' ทำการใส่ค่าใหม่

'ประหยด' == 'ประหยด' ทำการออกจาก loop

```
In [3]: name = ''
```

```
In [4]: if name == 'ประหยด':  
    print('สวัสดี ประหยด')
```

```
In [6]: if name == 'ประหยด':  
    print('สวัสดี ประหยด')  
else:  
    print('คุณชื่ออะไร !??')
```

สวัสดี ประหยด

```
In [7]: if name == 'ประหยด':  
    print('สวัสดี ประหยด')  
elif Job == 'นายก':  
    print('คุณชื่ออะไร !??')
```

สวัสดี ประหยด

```
In [5]: while name != 'ประหยด':  
    print('กรุณาใส่ชื่อของคุณครับ/ค่ะ.')  
    name = input()  
print('ลาก่อຍ !')
```

กรุณาใส่ชื่อของคุณครับ/ค่ะ.

prayad

กรุณาใส่ชื่อของคุณครับ/ค่ะ.

prayut

กรุณาใส่ชื่อของคุณครับ/ค่ะ.

ประหยด

ลาກ่อຍ !

# Homework Session 2

1. Boolean มีค่าอะไรได้บ้าง
2. ค่าด้านล่างเหล่านี้มีผลลัพธ์อย่างไรบ้าง
  - a.  $(5 > 4) \text{ and } (3 == 5)$
  - b.  $\text{not } (5 > 4)$
  - c.  $(5 > 4) \text{ or } (3 == 5)$
  - d.  $\text{not } ((5 > 4) \text{ or } (3 == 5))$
  - e.  $(\text{True and True}) \text{ and } (\text{True == False})$
  - f.  $(\text{not False}) \text{ or } (\text{not True})$
3. อธิบายความแตกต่างของ = กับ ==



# Homework Session 2

4. ลองสร้าง for loop โดยที่มี range ดังต่อไปนี้
  - a. range(10)
  - b. range(0,10)
  - c. range(0,10,1)
5. เรากำหนดอย่างไรเพื่อป้องกันการเกิด Infinite Loop



# Morning

---

Day-01



Boyd BigData RPG - Sorratat Sirirattanajakarin

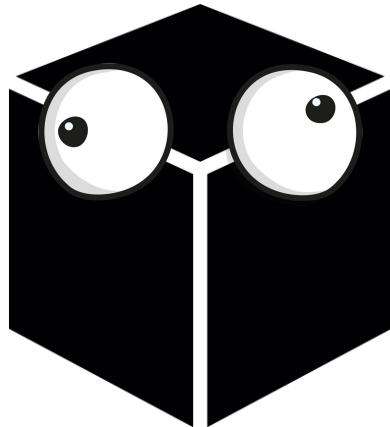
1. Introduction to Data Science
2. Install Anaconda and Jupyter
3. **Python Programming Basics**
  1. Operation
  2. Variable
  3. Flow Control
  4. Condition ( If, For, While )
  5. **Function, Method**
  6. List, Tuple
  7. Dictionaries and Structural Data
  8. Manipulating Strings



# Function and Method

Function ฟังก์ชัน ก็เหมือนกับการสร้าง โปรแกรมขนาดเล็กๆ  
ไว้ใช้งานเองแบบง่ายๆ

Method เป็นการเรียกใช้งาน Function ย่อๆ ที่อยู่ใน Module ใหญ่



FUNCTION



METHOD



# Function and Method

---

ความแตกต่างในการเรียกใช้งาน ระหว่าง Function และ Method

Function

**BMIcalculator(weight,  
height)**

Method

**weightAndHeight.BMIca  
lculator()**



# Function

---

## Create

```
def happy():  
    print('สวัสดีลันมาจากโลกอนาคต!')  
    print('เราจะทำให้พากເຮືອສຸກ')  
    print('และມີຄວາມສຸຂຈນລັ້ນອອກມາ')
```



# Function

---

**สวัสดีจันมาจากโลกอนาคต!**

## Result

**เราจะทำให้พากเนอสนุก**

**และมีความสุขจนลืนอุกมา**



# Function

ทำไมเราถึงต้องสร้าง Function ด้วยล่ะ!? ก็เพื่อหลีกเลี่ยงการทำงาน หรือการเขียนคำสั่งที่ใช้บ่อยๆและซ้ำๆนั่นเอง

1

**happy()**

2

**print('สวัสดีลั่นมาจากโลกอนาคต!')**

**print('เราจะทำให้พากເຮືອສຸກ')**

**print('และມີຄວາມສຸຂຈນລັ້ນອອກນາ')**

ผลลัพธ์เดียวกันแต่ด้านซ้ายเขียนสั้นกว่าเห็นๆ !!



# Fortune Cookie Program

มาลองสร้างโปรแกรม Cookie เลี้ยงหายกันดีกว่า ^^



# Code Session

สร้างกล่องขึ้นมา รับค่าชนิดของ String

ใช้ If Condition

ใช้ If + else Condition

ใช้ If + elif Condition

ใช้ While loop หากไม่เท่ากับ 'ประยัด' ออก loop

'prayad' != 'ประยัด' ทำการใส่ค่าใหม่

'ประยัด' == 'ประยัด' ทำการออกจาก loop

In [5]:

```
import random

def คุกเกี้เลี่ยงไทย(เลี่ยงไทย):
    if เลี่ยงไทย == 1:
        return 'แอบมองเธอว้อยู่นะจี๊'
    elif เลี่ยงไทย == 2:
        return 'แต่เธอว่าไม่สนใจ'
    elif เลี่ยงไทย == 3:
        return 'แอบส่งใจให้นิดๆ'
    elif เลี่ยงไทย == 4:
        return 'แต่ดูเธอช่างเฉยเมย'
    elif เลี่ยงไทย == 5:
        return 'คนน่ารักก็เยอะนะ'
    elif เลี่ยงไทย == 6:
        return 'Yeah Yeah Yeah'
    elif เลี่ยงไทย == 7:
        return 'เหมือนว่าฉันนั้นเครังค้าง'
    elif เลี่ยงไทย == 8:
        return 'รักไม่รักก็ต้องเลี่ยง'
    elif เลี่ยงไทย == 9:
        return 'อยากบอกเธอว่ารัก'

r = random.randint(1, 9)
fortune = คุกเกี้เลี่ยงไทย(r)
print(fortune)
```

แอบมองเธอว้อยู่นะจี๊

# Local and Global Variable

ความแตกต่างระหว่าง Local and Global Variable



**bin**

ตัวแปรแบบ Global  
เรียกใช้งานที่ไหนก็ได้



**bin**

ตัวแปรแบบ Local  
เรียกใช้งานได้เฉพาะที่เท่านั้น



# Morning

---

Day-01



Boyd BigData RPG - Sorratat Sirirattanajakarin

1. Introduction to Data Science
2. Install Anaconda and Jupyter
3. **Python Programming Basics**
  1. Operation
  2. Variable
  3. Flow Control
  4. Condition ( If, For, While )
  5. Function, Method
  6. **List, Tuple**
  7. Dictionaries and Structural Data
  8. Manipulating Strings



# List and Tuple

List และ Tuple คือรูปแบบของ Container ที่ทำการบรรทุกค่าต่างๆ ได้หลายค่า

List

```
[“Prayad” , 10 , 44.44 , “PrawitZa”]
```

Tuple

```
(“Prayad” , 10 , 44.44 , “PrawitZa”)
```



# List

ค่าต่างๆจะถูกจัดเรียงตามลำดับใน List ด้วย เช่นกัน เรียกว่า Index



**INDEX**

**0**

**1**

**2**

**3**



# List

---

```
bins = [ "Prayad" , 10 , 44.44 , "PrawitZa" ]
```

"Prayad"                    10                    44.44                    "PrawitZa"

**bins[0]**

**bins[1]**

**bins[2]**

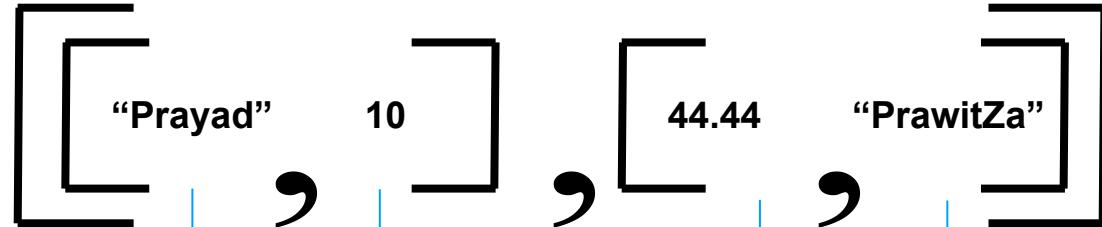
**bins[3]**



# List in list

**newBins**

=



**INDEX ชั้นที่ 1**

0

1

00

01

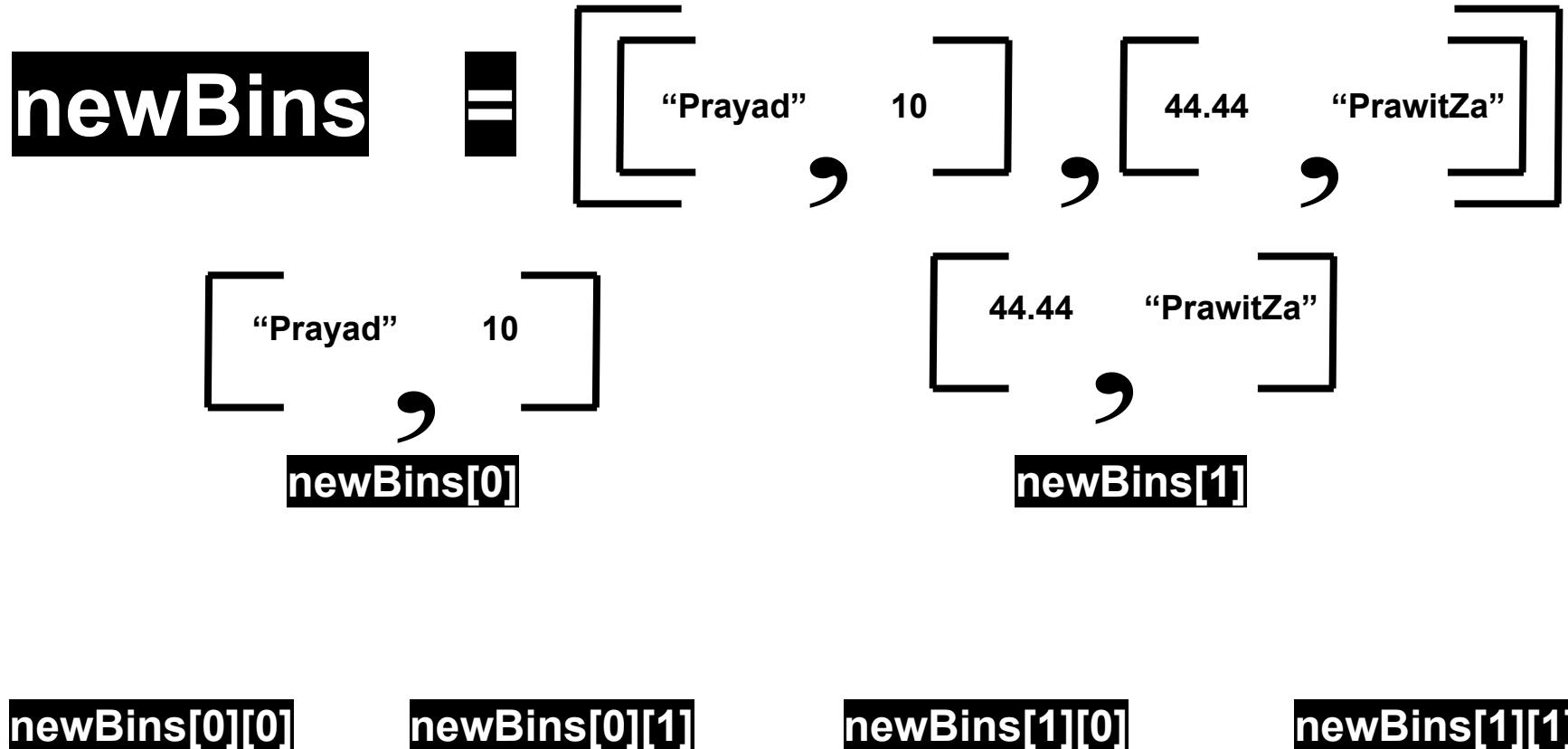
10

11

**INDEX ชั้นที่ 2**

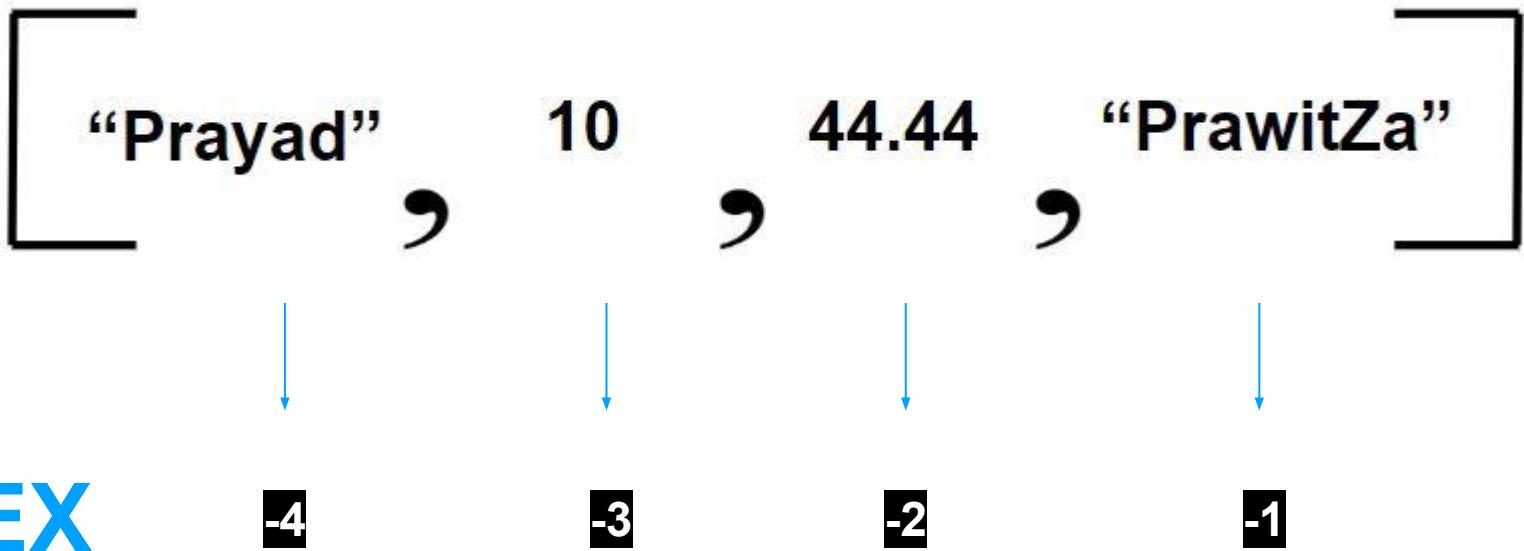


# List in list





# Negative Index





# Change Values in List

List คือ Container ที่สามารถทำการเปลี่ยนแปลงค่าภายในได้

'ประจำ'

ค่าที่ถูกนำเข้าไปเก็บใหม่

'pranop'

ค่าที่ถูกเอาออก หมดความหมาย



**bins[3]**



# Add Values in List

```
[“Prayad”, 10, 44.44, “PrawitZa”, “prawat”]
```

,

,

,

,

]



```
bins.append('prawat')
```



# List Concatenation

```
[“Prayad”, 10, 44.44, “PrawitZa”] + [“Prayad”, 10, 44.44, “PrawitZa”]
```

```
[“Prayad”, 10, 44.44, “PrawitZa”, “Prayad”, 10, 44.44, “PrawitZa”]
```



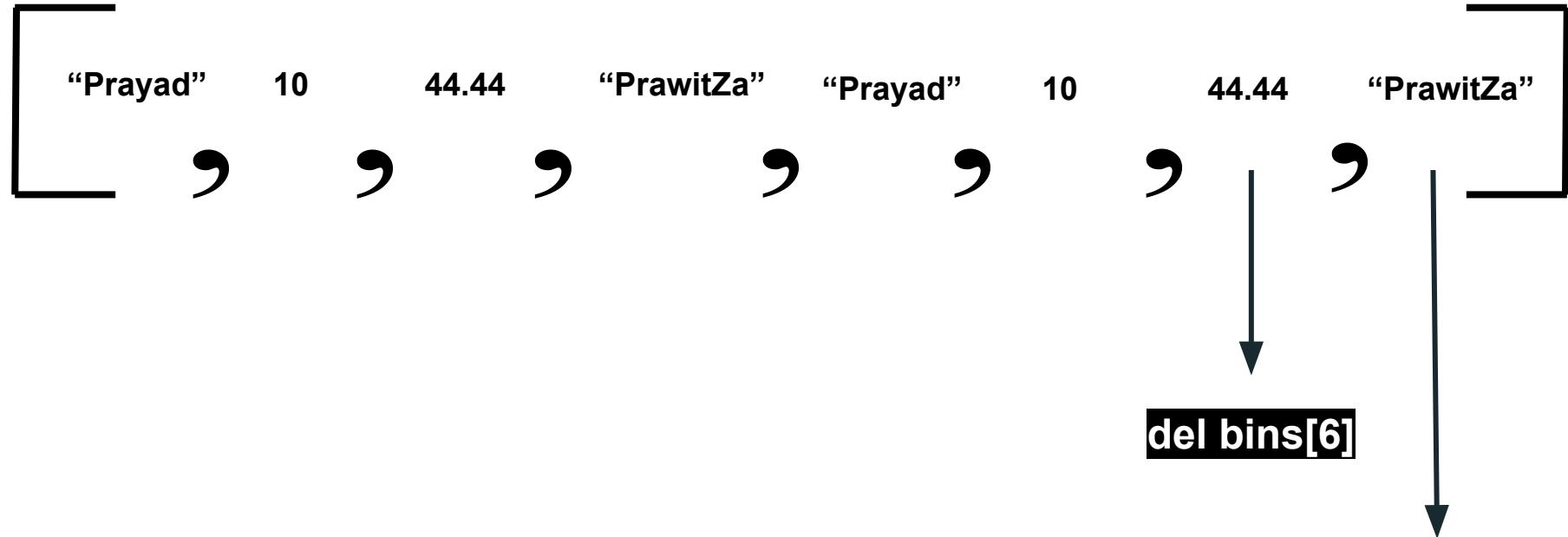
# List Multiply

```
[“Prayad”, 10, 44.44, “PrawitZa”] * 2
```

```
“Prayad”, 10, 44.44, “PrawitZa”, “Prayad”, 10, 44.44, “PrawitZa”  
,
```

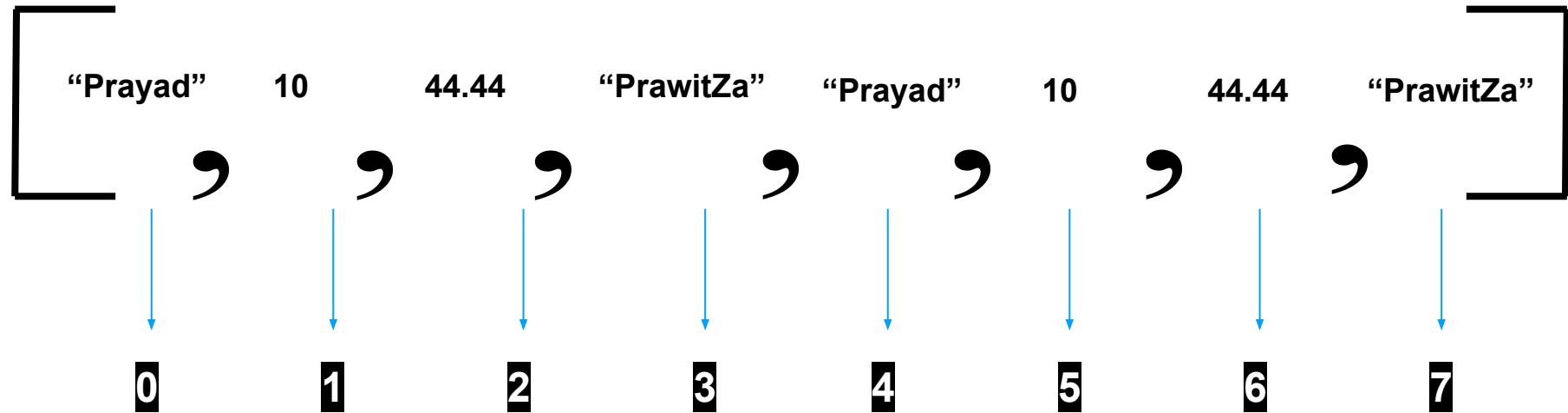


# Removing Values





# Slicing



`bins[:]`

หากใส่เครื่องหมาย : คือการเลือกทุกตำแหน่งมาใช้งาน



# Slicing

**bins[2:5]**

เลขตัวหน้าแสดงถึงเริ่มนับจาก  
ตำแหน่ง index นั้นเพื่อทำการนำค่า  
มาใช้

เลขตัวหลังแสดงถึงตำแหน่งที่ทำการ  
หยุด และไม่ได้นำค่าตำแหน่งนั้นมา  
ใช้



# Slicing

---

44.44

“PrawitZa”

“Prayad”

,

,

2

3

4

**bins[2:5]**



# Tuple

Tuple นั้นสามารถเก็บข้อมูลได้ คล้ายกับ List แต่ไม่สามารถเปลี่ยนแปลงค่า, เพิ่มค่า หรือทำการลบค่าได้

( “Prayad” , 10 , 44.44 , “PrawitZa” )

แล้วทำอย่างไรดีล่ะ ให้สามารถเปลี่ยนแปลงข้อมูลได้ ?



# Converting List and Tuple

---

`tuple(['yolo'])`



**LIST to TUPLE**

`list(('yolo'))`



**TUPLE to LIST**

# Code Session

ทำการเปลี่ยน list เป็น tuple

ทดลองเรียกค่า กล้ายเป็น tuple แล้ว

ทำการใช้ slice เพื่อเข้าถึงข้อมูลได้

ทดลองทำการเปลี่ยนแปลงค่าดูที่ตำแหน่ง index = 4

ต้องการเปลี่ยนแปลงค่าตำแหน่ง index = 4 แต่ไม่เป็นผล เพราะ tuple ไม่มีคุณสมบัติดังกล่าว

## TUPLE

```
In [30]: binsTuple = tuple(bins)
```

```
In [31]: binsTuple
```

```
Out[31]: ('ประจวบ',
           'ประจวบ',
           'prayad',
           'prawiit',
           'prawiit',
           'prawat',
           'prajuap',
           'prajuap')
```

```
In [33]: binsTuple[:-4]
```

```
Out[33]: ('ประจวบ', 'ประจวบ', 'prayad', 'prawiit')
```

```
In [34]: binsTuple[4] = 'prayette'
```

```
-----  
TypeError  
<ipython-input-34-a75a67e42342> in <module>  
----> 1 binsTuple[4] = 'prayette'
```

```
TypeError: 'tuple' object does not support
```

# Morning

---

Day-01



Boyd BigData RPG - Sorratat Sirirattanajakarin

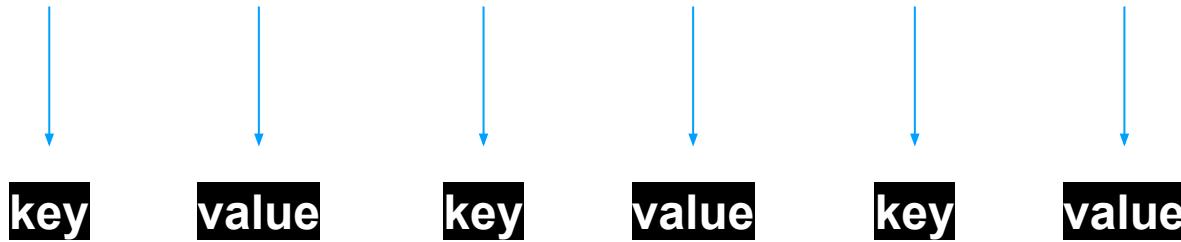
1. Introduction to Data Science
2. Install Anaconda and Jupyter
3. **Python Programming Basics**
  1. Operation
  2. Variable
  3. Flow Control
  4. Condition ( If, For, While )
  5. Function, Method
  6. List, Tuple
  7. **Dictionaries and Structural Data**
  8. Manipulating Strings



# Dictionaries

Dictionary คล้ายกับ List แต่แทนที่จะเข้าถึงตำแหน่งข้อมูลนั้นๆจากการใช้ Index เราเปลี่ยนมาใช้รูปแบบที่เรียกว่า Key-Value แทน

```
myDict = {'อุ่ง': 'อร', 'หลาม': 'ปณ', 'กวน': 'ไข่มุก'}
```





# Dictionaries vs List

---

```
myDict = {'อึ้ง': 'อร', 'หลาม': 'ปัญ', 'กวิน': 'ไข่มุก'}
```

```
yourDict = {'กวิน': 'ไข่มุก', 'อึ้ง': 'อร', 'หลาม': 'ปัญ'}
```

myDict == yourDict

TRUE



# Dictionaries vs List

---

**myList = ['ອົກ', 'ປ່ານ', 'ໄຟ້ມູກ']**

**yourList = ['ປ່ານ', 'ໄຟ້ມູກ', 'ອົກ']**

**myList == yourList**

**FALSE**



# Access Key Value and Item

```
myDict = {'อุ่ง': 'อร', 'หลาน': 'ป้อม', 'กวน': 'ไข่มุก'}
```



```
myDict.keys()
```



```
'อุ่ง', 'หลาน', 'กวน'
```



# Access Key Values and Item

```
myDict = {'อุ่ง': 'อร', 'หลาม': 'ป้อม', 'กิวิน': 'ไข่มุก'}
```



```
myDict.values()
```



```
'อร', 'ป้อม', 'ไข่มุก'
```



# Access Key Value and Item

```
myDict = {'อุ่ง': 'อร', 'หลาม': 'ป้อม', 'กิวิน': 'ไข่มุก'}
```



```
myDict.item()
```



```
('อุ่ง','อร'), ('หลาม','ป้อม'), ('กิวิน','ไข่มุก')
```



# Checking Whether a Key or Value Exists

```
myDict = {'อุ่ง': 'อร', 'หลาม': 'ปัญ', 'กวน': 'ไข่มุก'}
```



```
'อุ่ง' in myDict.keys()
```



TRUE



# Checking Whether a Key or Value Exists

```
myDict = {'อุ่ง': 'อร', 'หลาม': 'ป้อม', 'กวน': 'ไข่มุก'}
```



```
'กุหลาบไฟ' in myDict.keys()
```



**FALSE**



# Checking Whether a Key or Value Exists

```
myDict = {'อุ่ง': 'อร', 'หลาม': 'ปัญ', 'กิวิน': 'ไข่มุก'}
```



```
'ไข่มุก' in myDict.values()
```



TRUE



# Checking Whether a Key or Value Exists

```
myDict = {'อุ่ง': 'อร', 'หลาม': 'ป้อม', 'กวิน': 'ไข่มุก'}
```



```
'prayad' in myDict.values()
```



**FALSE**



# get() Method

**myDict.get('key' [, default values])**

ใส่ ชื่อ dictionary ที่ทำการ  
ใช้งานลงใน

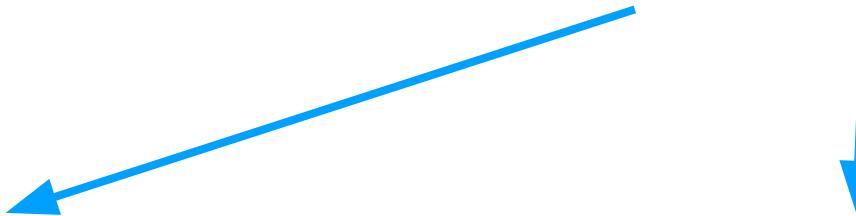
ใส่ key ที่ต้องการค้นหา  
เพื่อดึง value ออกมาใช้

หากทำการค้นหา key ที่ใส่  
ไปแล้วไม่เจอก็จะได้ค่า 0  
กลับมาแทน



# setdefault() Method

**myDict.setdefault('key', 'value')**



ใช้ในการใส่ key ใหม่ที่ยังไม่เคยมีใน dict ที่เราเรียกใช้ method

ใส่ค่า value เริ่มต้นให้สำหรับ key ใหม่ที่ทำการถูกสร้างขึ้นมาใน dict ที่เราเรียกใช้ method



# Example setdefault() Method

```
dict = {}
```

```
dict.setdefault( "dog", 0 )
```



Set ค่า Default

```
dict.setdefault( "cat", 5 )
```



Set ค่า Default

```
dict[ "dog" ] = dict[ "dog" ]+1
```



ผลลัพธ์ = 1

```
dict[ "cat" ] = dict[ "cat" ]+1
```



ผลลัพธ์ = 6

# Code Session

นำ message มารับค่า string

สร้าง dictionary มารองรับไว้ก่อน

สร้าง for loop เพื่อทำการนับพยัญชนะที่บลละตัว

ทำการใช้ setdefault เพื่อดักจับพยัญชนะ ตัวใหม่

หากเจอตัวเดิมจะทำการเพิ่ม value ให้กับ key  
พยัญชนะตัวนั้น +1 ซึ่งอัปเดทภายใน count {}

หลังจาก for loop เสร็จทำการ print ว่าภายใน dict มี  
พยัญชนะอะไรบ้าง จำนวนเท่าไหร

## Word Count ¶

In [31]:

```
message = """แอบมองเหรออยู่นะจ๊ะ
แต่เชօไม่รู้บังเลย
แอบส่งใจให้นิดๆ
แต่ดูเชօช่างเฉยเมย
ເອົາລະເຕີຍນິໃຈໄວ້หน້ອຍ
ມັນຈະຫວັກຂອຍດ້ອງເສີຍກັນ
Yeah Yeah Yeah""""
```

```
count = {}
```

```
for character in message:
    count.setdefault(character, 0)
    count[character] = count[character] + 1
```

```
print(count)
```

```
{'ອ': 3, 'ີ': 3, 'ມ': 5, 'ຍ': 8, 'ບ': 3,
 'ຳ': 2, 'ທ': 3, 'ະ': 4, 'ັ': 6, 'ດ': 2,
 'ສ': 2, 'ົ': 3, 'ແ': 4, 'ີ': 2, '\n': 6,
 'ນ': 5, 'າ': 3, 'Y': 3, "": 1, 'ຈ': 4,
 'ີ': 1, 'ຈ': 1, 'ໜ': 1, 'ີ': 3, 'ກ': 3,
 'ວ': 2, 'ເ': 9, 'e': 3, 'ໆ': 1, 'ຮ': 2,
 'ອ': 11, "": 6, 'ດ': 4, 'ລ': 2, 'ໄ': 2,
 'ຫ': 3, "": 9, 'ກ': 2}
```

# Morning

---

Day-01



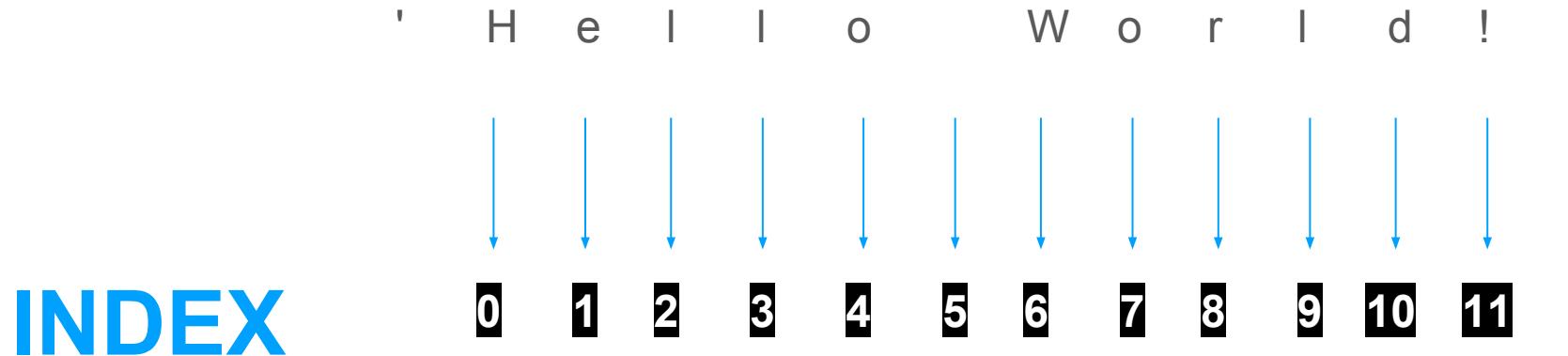
Boyd BigData RPG - Sorratat Sirirattanajakarin

1. Introduction to Data Science
2. Install Anaconda and Jupyter
3. **Python Programming Basics**
  1. Operation
  2. Variable
  3. Flow Control
  4. Condition ( If, For, While )
  5. Function, Method
  6. List, Tuple
  7. Dictionaries and Structural Data
  8. **Manipulating Strings**



# Manipulating Strings

สำหรับบทนี้จะเน้นไปยัง method ที่เข้ามาใช้ร่วมงานกับ string แต่หลักๆเลย string ก็มี index ของตัวเองเช่นกันนะยกตัวอย่างคำว่า 'Hello World!'





# Manipulating Strings

การใช้งานเข้าถึง String มีคำสั่งค่อนข้างคล้ายกับ List เพราะสามารถเรียกใช้ตาม Index และ Slice ได้

**hello** = 'Hello World!'

**INDEX**

**hello[4]** = 'o'

**SLICE**

**hello[:5]** = 'Hello'



# Escape Character

ตัวสัญลักษณ์พิเศษที่ช่วยอะไรบางอย่างภายใน String

| Escape Character | Result                                |
|------------------|---------------------------------------|
| \'               | ทำการใส่ single quote                 |
| \"               | ทำการใส่ double quote                 |
| \n               | ทำการใส่ new line (ขึ้นบันท Ruddใหม่) |
| \t               | ทำการใส่ tab (เว้นวรคแบบ กด แทป)      |
| \\\              | ทำการใส่ Backslash                    |



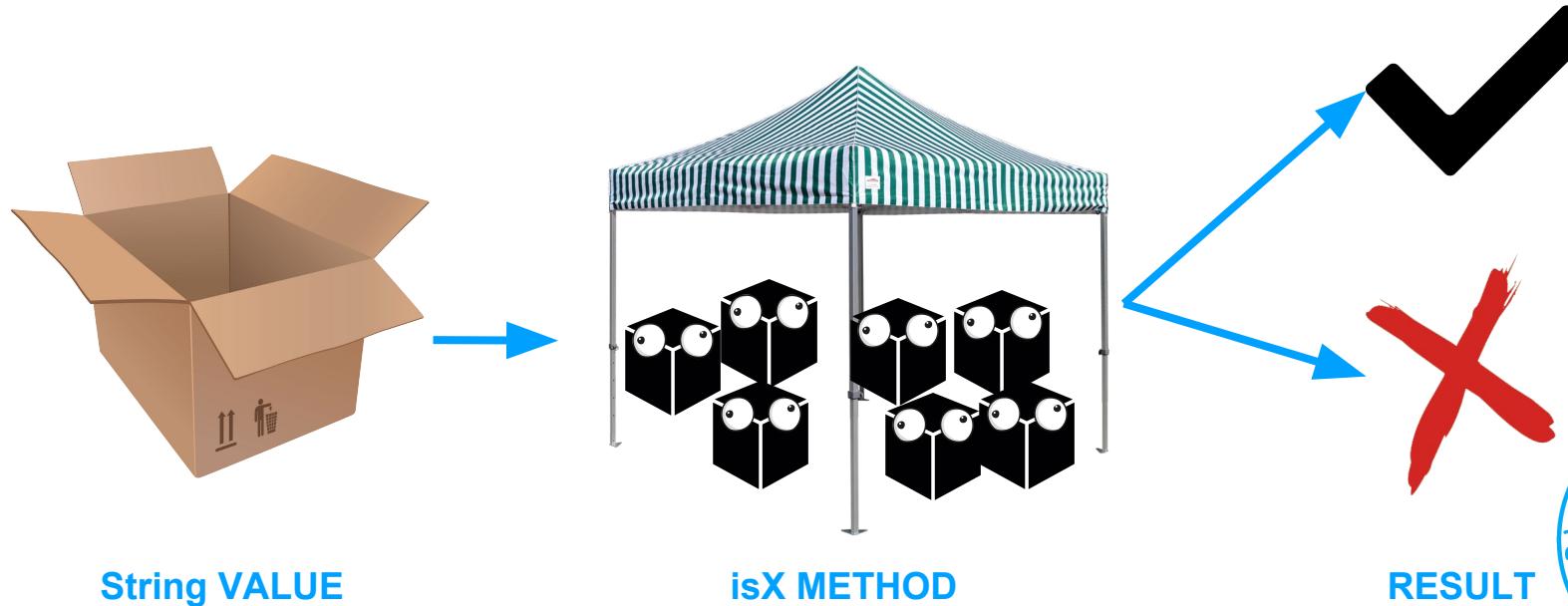
# Useful String Method

| Method      | Result   |
|-------------|--|
| upper()     | ใช้ในการปรับ String ที่เลือกให้มีขนาดใหญ่ขึ้น  |
| lower()     | ใช้ในการปรับ String ที่เลือกให้มีขนาดเล็กลง  |
| isupper()   | ใช้ในการเช็คว่า String ที่เลือกทุกตัวนั้น มีขนาดใหญ่หรือไม่                              |
| islower()   | ใช้ในการเช็คว่า String ที่เลือกทุกตัวนั้น มีขนาดเล็กหรือไม่                              |
| isalpha()   | เช็ค String ว่ามีเฉพาะ ตัวอักษร อยู่หรือไม่ ถ้าใช่จะได้ True ไม่ใช่เป็น False            |
| isalnum()   | เช็ค String ว่ามีเฉพาะ ตัวอักษร และ ตัวเลข อยู่หรือไม่ ถ้าใช่จะได้ True ไม่ใช่เป็น False |
| isdecimal() | เช็ค String ว่ามีเฉพาะ ตัวเลข ถ้าใช่จะได้ True ไม่ใช่เป็น False                          |
| isspace()   | เช็ค String ว่ามีเฉพาะ ' ', '\n', '\t' อยู่หรือไม่ ถ้าใช่จะได้ True ไม่ใช่เป็น False     |
| istitle()   | เช็ค String ว่ามี ตัวอักษรแรกขึ้นต้นด้วยตัวใหญ่หรือไม่ ถ้าใช่จะได้ True ไม่ใช่เป็น False |

# isX Method on String

TIP

ข้อสังเกตุผลลัพธ์ของ is Method ที่เราพบจะเห็นว่าเป็น Boolean ไว้ใช้สำหรับเช็คค่าของชุดของ String





# join() Method จาก list >> string

ในการทำงานกับ String สิ่งที่ขาดไปไม่ได้เลย คือการใช้งาน 2 Method นี้ อันได้แก่ join() และ split() นั่นคือการ เชื่อมคำ และทำการตัดคำ ตามลำดับ

**STEP 1**

```
', '.join(['ชั้น', 'คือ', 'data scientist'])
```

**STEP 2**

```
'ชั้น' 'คือ' 'data scientist'
```

**STEP 3**

```
'ชั้น, คือ, data scientist'
```

เชื่อมด้วย comma ,



# join() Method จาก list >> string

---

**STEP 1**

```
' '.join(['ชั้น', 'คือ', 'data scientist'])
```

**STEP 2**

```
'ชั้น' 'คือ' 'data scientist'
```

**STEP 3**

```
'ชั้น คือ data scientist'
```

เชื่อมด้วยเว้นวรคแบบ spacebar



# split() Method จาก string >> list

---

STEP 1

'ชั้น คือ data scientist'.split()

STEP 2

'ชั้น' 'คือ' 'data' 'scientist'

STEP 3

['ชั้น', 'คือ', 'data', 'scientist']

แยกคำด้วย comma spacebar



# split() Method จาก string >> list

---

STEP 1

'ชั้น คือ data scientist'.split('data')

STEP 2

'ชั้น คือ ' ' scientist'

STEP 3

['ชั้น คือ ', ' scientist']

แยกคำด้วยคำว่า 'data'



# rjust(), ljust(), and center()

กลุ่มสมาชิก Method ที่ไว้ใช้ในการจัดการ ปรับการจัดวาง String ให้อยู่ในรูปแบบที่เราต้องการ

```
'hello'.rjust('10','=')
```

String ที่เรานำมาใช้

ทำการสร้างพื้นที่ 10 หน่วย  
โดยเลื่อน string ที่เราใช้ไป  
ทางขวา ( เพราะเราใช้ rjust )

เติมส่วนที่หายไปจากพื้นที่ที่  
ขยายด้วย สัญลักษณ์ หรือ  
string ที่เราต้องการเพิ่ม



# rjust()

---

ใช้งาน

```
'hello'.rjust('10','=')
```

ผลลัพธ์

```
=====hello
```



# Ijust()

---

ใช้งาน

```
'hello'.Ijust('10','=')
```

ผลลัพธ์

```
hello=====
```



# center()

---

ใช้งาน

```
'hello'.center('10','=')
```

ผลลัพธ์

```
====hello====
```



# strip(), rstrip(), and lstrip()

Method กลุ่มที่ช่วยในการกำจัด Space ออกไป หรือกำจัดคำที่ไม่ต้องการออกไป ก็ทำได้เช่นกัน

**'hello'.rjust('10','=')**

String ที่เรานำมาใช้

ทำการสร้างพื้นที่ 10 หน่วย  
โดยเลื่อน string ที่เราใช้ไป  
ทางขวา ( เพราะเราใช้ rjust )

เติมส่วนที่หายไปจากพื้นที่ที่  
ขยายด้วย สัญลักษณ์ หรือ  
string ที่เราต้องการเพิ่ม



# rstrip()

---

ใช้งาน

```
'Hello'.rstrip()
```

ผลลัพธ์

```
'Hello'
```



# Istrip()

---

ใช้งาน

' Hello '.strip()

ผลลัพธ์

'Hello '



# strip()

---

ใช้งาน

' Hello '.strip()

ผลลัพธ์

'Hello'

# Code Session

สร้าง While Loop True เพื่อรันโปรแกรม

ทำการใส่ค่าอายุเข้าไปในฟังก์ชัน

ถ้าตัวเลขเป็นจำนวนเต็มจะหยุดโปรแกรมทันที

ถ้าใส่ค่าอื่นๆ โปรแกรมจะยังทำงานต่อ

สร้าง While Loop True เพื่อรันโปรแกรม แต่คราวนี้ เป็นโปรแกรมใส่พาสเวิร์ด

ถ้าเกิดพาสเวิร์ดเป็นตัวอักษรและตัวเลขจะหยุด โปรแกรมทันที แต่ถ้าไม่ใช่จะทำงานให้ใส่ใหม่อีกครั้ง

## Validation Program

```
while True:  
    print('กรุณาระบุอายุของคุณ: ')  
    age = input()  
    if age.isdecimal():  
        break  
    print('กรุณาระบุเป็นจำนวนเต็มครับ/ค่ะ')
```

```
กรุณาระบุอายุของคุณ:  
20.4  
กรุณาระบุเป็นจำนวนเต็มครับ/ค่ะ  
กรุณาระบุอายุของคุณ:  
25
```

```
while True:  
    print('กรุณาระบุพาสเวิร์ดใหม่ (ใช้ตัวอักษรและตัวเลขร่วมกัน): ')  
    password = input()  
    if password.isalnum():  
        break  
    print('พาสเวิร์ดต้องเป็นตัวอักษรและตัวเลขเท่านั้น')
```

```
กรุณาระบุพาสเวิร์ดใหม่ (ใช้ตัวอักษรและตัวเลขร่วมกัน):  
prayad yolo  
พาสเวิร์ดต้องเป็นตัวอักษรและตัวเลขเท่านั้น  
กรุณาระบุพาสเวิร์ดใหม่ (ใช้ตัวอักษรและตัวเลขร่วมกัน):  
prayad1234
```

# Code Session

สร้าง พิงก์ชัน รับ 3 ค่า

ทำการแสดงค่าด้านบน PICNIC ITEM

ทำการ for loop ดึงค่า key value ออกมากำหนด

จัดรูปแบบแสดงผลด้วย ljust และ rjust

สร้าง dict ที่บรรจุค่าต่างๆดังกล่าว

ทำการเรียกใช้ พิงก์ชัน และใส่ 3 ตัว แปรเพื่อจัดความกว้างด้านซ้ายรับ 12 ตัว และด้านขวารับได้ 5 ตัว

```
def printPicnic(itemsDict, leftWidth, rightWidth):
    print('PICNIC ITEMS'.center(leftWidth + rightWidth, '-'))
    for k, v in itemsDict.items():
        print(k.ljust(leftWidth, '.') + str(v).rjust(rightWidth))
```

```
picnicItems = {'sandwiches': 4, 'apples': 12, 'cups': 4, 'cookies': 8000}

printPicnic(picnicItems, 12, 5)

---PICNIC ITEMS---
cookies..... 8000
sandwiches.. 4
cups........ 4
apples..... 12
```



# Homework Session 3

1. อะไรคือ escape character
2. \t กับ \t หมายถึงอะไร
3. ถ้าไม่ใช้ \t ใน string เราจะวิธีเขียนแบบอื่นใหม่ เขียนได้อย่างไรบ้าง
4. ค่าเหล่านี้ได้ผลลัพธ์อย่างไรบ้าง
  - a. 'Hello world'[0]
  - b. 'Hello world'[4:5]
  - c. 'Hello world'[:2]
  - d. 'Hello world'[3:]
  - e. 'Hello'.title()
  - f. 'Hello'.upper().isupper()
  - g. 'Hello'.upper().lower()



# Homework Session 3

5. ค่าเหล่านี้ได้ผลลัพธ์อย่างไรบ้าง

- a. 'กвин อุ่ง สลอธ'.split()
- b. '---'.join('กвин อุ่ง สลอธ'.split())

6. มีวิธีลบช่องว่างที่เกิดขึ้นด้านหน้า string กับด้านหลัง string อย่างไรได้บ้าง



# Python Data Science for Developer

Thank you

