

Basic Programming and Database



ร.ต.สมโภชน์ กุลธารารมณ

ผู้สอน

ร.อ. สมโภชน์ กุลธารารมณธ์ อาจารย์แผนกการศึกษา ประจำแผนกปฏิบัติการที่ 1 กองปฏิบัติการที่ 2 สำนักปฏิบัติการทางไซเบอร์ นหบ.ทหาร

• ประวัติการศึกษา

• ปริญญาโท

- วท.ม. สาขาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

• ปริญญาตรี

- วท.บ. สาขาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

▶ ประวัติการทำงาน

▶ พ.ศ. 2549- 2551

- ▶ ตำแหน่ง Developer บริษัท AISOFT (บริษัทรับพัฒนาแอปพลิเคชัน
ด้านการท่องเที่ยว)

▶ พ.ศ. 2551- 2561

- ▶ หัวหน้างานพัฒนาซอฟต์แวร์ สำนักวิทยบริการและเทคโนโลยีสารสนเทศมหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

▶ พ.ศ. 2561 –ปัจจุบัน

- ▶ อาจารย์แผนกการศึกษา กองวิทยาการศูนย์ไซเบอร์ทหารกองบัญชาการกองทัพไทย

Certificate

CompTIA: Classroom Trainer (CTT+), Project+, A+, Network+ , Security+

ORACLE: Oracle Database 11g Administrator Certified Associate

ADOBE: Visual Communication using Adobe Photoshop® CS6

NSTDA (สวทช): Information Technology Professionals Examination Council: ITPEC

Microsoft: Microsoft Certified Trainer, Microsoft Certified Solution Expert

จุดประสงค์รายวิชา

1. เพื่อให้ผู้เรียนเข้าใจโครงสร้างข้อมูล และการออกแบบโปรแกรมที่เหมาะสมในการจัดการข้อมูลได้
2. เพื่อให้ผู้เรียนสามารถวิเคราะห์ ออกแบบ และเป็นแนวทางการพัฒนาโปรแกรมตามกรอบที่กำหนดได้
3. เพื่อให้ผู้เรียนทราบถึงโครงสร้างในการเขียนโปรแกรม
4. เพื่อให้ผู้เรียน เรียนรู้คำสั่ง ตัวแปร และวิธีการใช้งานในรูปแบบต่าง ๆ ได้

แผนการสอน

- Day 1-2

- Intro & Pretest

- รู้จักโครงสร้างข้อมูลและอัลกอริทึม

- รู้จักภาษา Python

- Intro & Install

- Syntax,, Comment

- Python Variable,

- Python Data Type,

- Python Coding Example

- **อาร์เรย์ (Array)**

- Python Collections (Arrays)

- List

- Tuple

- Set

- Dictionary

- Python Coding Example

• Day 3

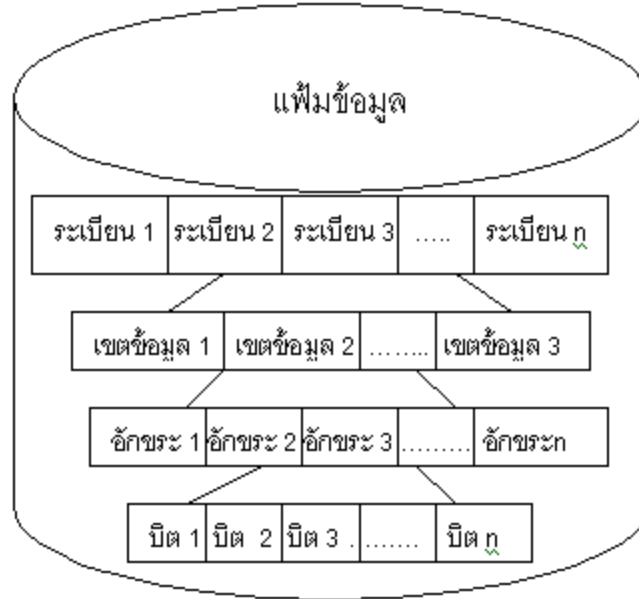
- ลิงคิสต์ (Linked-List)
- สแตก (Stack)
- Python Coding Example
- Test เก็บคะแนน
- Assignment (Mini Project) เก็บคะแนน
- คิว (Queue)
- ต้นไม้ (Tree)
- Python Coding Example

- Day 4

- กราฟ (Graph)
- การค้นหา (Searching)
- Python Coding Example
- การจัดเรียง (Sorting)
- Python Coding Example
- Test เก็บคะแนน

โครงสร้างข้อมูล

- โครงสร้างข้อมูล (File Structure) หมายถึง ลักษณะการจัดแบ่งพิกัดต่าง ๆ ของข้อมูล สำหรับแต่ละระเบียน (Record) ในแฟ้มข้อมูลเพื่อให้คอมพิวเตอร์สามารถรับไปประมวลผลได้ ประกอบด้วยส่วนต่าง ๆ ดังนี้



โครงสร้างข้อมูล (ต่อ)

- เป็นวิธีจัดเก็บข้อมูลในคอมพิวเตอร์ เพื่อให้สามารถนำมาใช้ได้อย่างมีประสิทธิภาพ
- การใช้งานโครงสร้างข้อมูล ต้องมีขั้นตอนวิธีที่เหมาะสม จึงจะสามารถใช้งานได้มีประสิทธิภาพ
- การออกแบบโครงสร้างข้อมูลที่ดีจะช่วยลดเวลาในการกระทำการ และลดการใช้งานในพื้นที่ความจำด้วย

การจัดเก็บข้อมูล

เป็นการนำข้อมูลที่มีอยู่เข้าไปเก็บในเครื่องคอมพิวเตอร์ อาทิ

- ประวัติส่วนตัว
- สินค้าคงคลัง
- การขาย
- การเงิน
- บุคลากร
- เป็นต้น

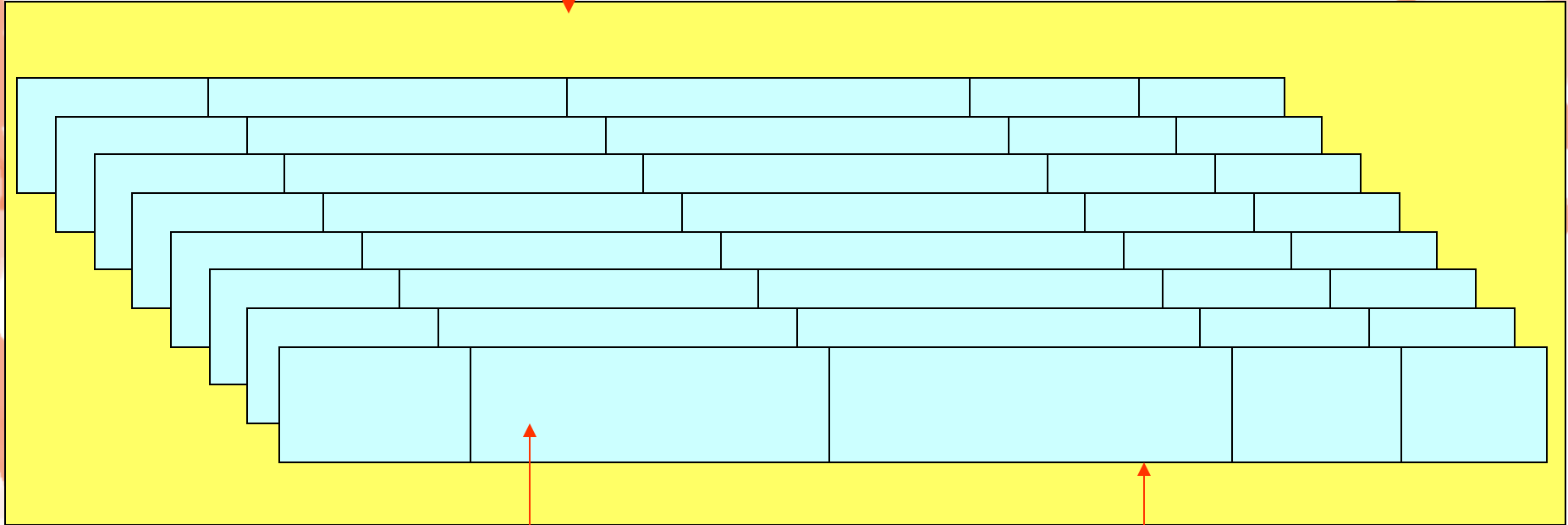
สมมติมีข้อมูลจำนวนหนึ่งประกอบด้วยชื่อ ที่อยู่ หมายเลขโทรศัพท์
หากต้องการเขียน โปรแกรมเพื่อจัดการกับข้อมูลเหล่านี้ จะต้องพิจารณาสิ่งต่างๆ

- วิธีการจัดเก็บลงในหน่วยความจำ
- วิธีการดึงกล่าวสามารถใช้ได้กับข้อมูลปริมาณมากเพียงใด
- สามารถเพิ่มข้อมูลใหม่ ลบข้อมูลเก่าได้อย่างรวดเร็วหรือไม่
- สามารถค้นหาข้อมูลที่ต้องการ ได้เร็วเพียงใด
- ถ้าต้องการจัดข้อมูลให้เรียงตามตัวอักษร จะใช้วิธีใด

หน่วยของข้อมูล

- บิต (bit)
- อักขระ (character)
- เขต (field)
- ระเบียน (record)
- แฟ้ม (file)
- ฐานข้อมูล (database)

แฟ้มข้อมูล (Data file)



เขต (Field)

ระเบียน (Record)

ระเบียบนักเรียน

เลขประจำตัว (8)

ระดับคะแนนเฉลี่ย (4)



ชื่อ (10)

สกุล (15)

หน่วยกิตสะสม (3)

ระเบียบสินค้า

รหัสสินค้า (4)

ระดับคะแนนเฉลี่ย (4)



ชื่อสินค้า(20)

สกุล (15)

ราคา (5)

00101010 *
00101011 +
00101100 /
00101101 -
00101110 .
00101111 /
00110000 0
00110001 1
00110010 2
00110011 3
00110100 4
00110101 5
00110110 6
00110111 7
00111000 8
00111001 9
00111010 :
00111011 ;
00111100 <

00110101 00110000 00110001 001100011

บิต

5

0

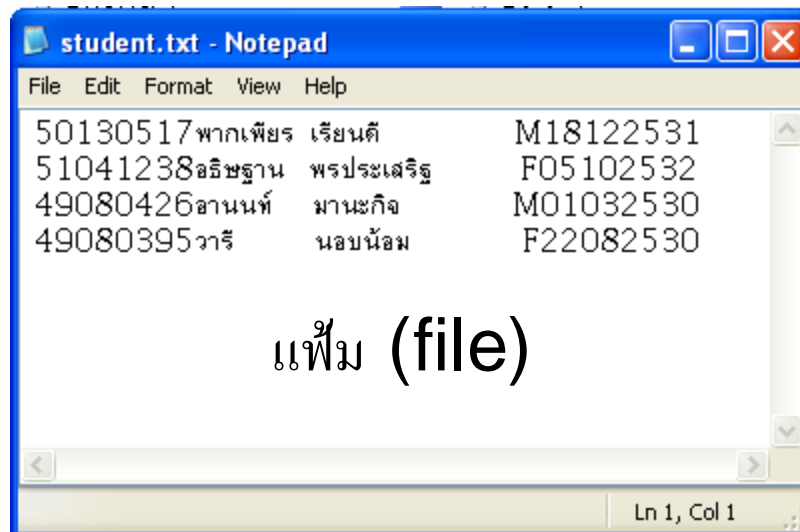
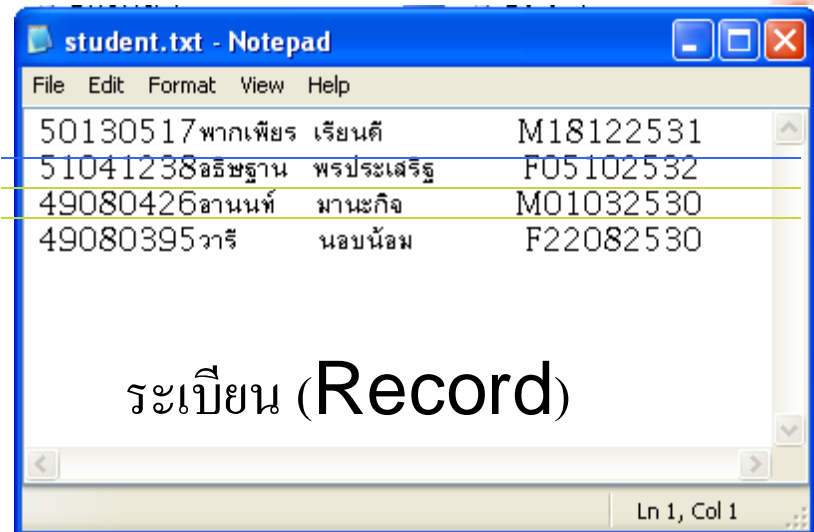
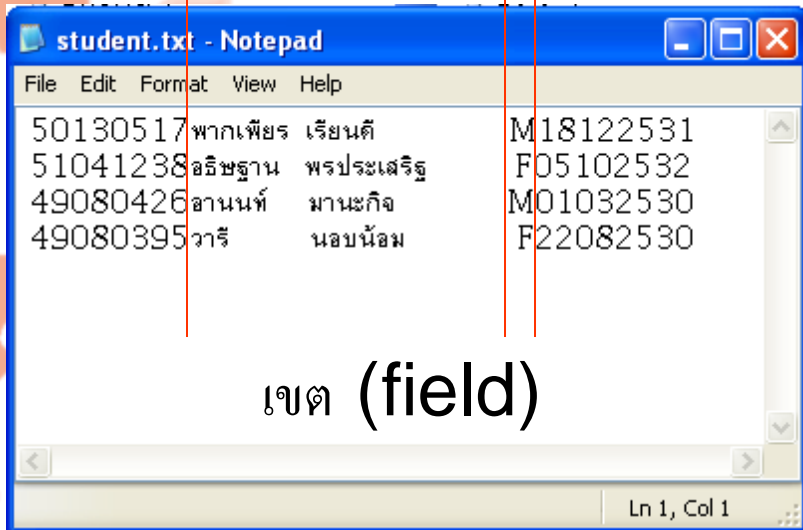
1

3

อักขระ

5013

เขต




```
student.txt - Notepad
File Edit Format View Help
50130517 พากเพียร เรียนดี M18122531
51041238 อธิษฐาน พรประเสริฐ F05102532
49080426 อานนท์ มานะกิจ M01032530
49080395 วรวิ นอบน้อม F22082530
Ln 1, Col 1
```

```
studentgrade.txt - Notepad
File Edit Format View Help
501305171 514181123.5
501305171 514171112.5
510412381 513871114.0
490802061 519990111.5
490803951 519990112.0
```

```
faculty.txt - Notepad
File Edit Format View Help
01Agriculture
02Fisheries
03Forestry
04Science
05Engineering
06Education
07Economics
08social science
09Humanities
10Agro-Industry
|
```

ฐานข้อมูล (Database)

แฟ้ม (File)

แฟ้มเป็นที่รวบรวมข้อมูล หรือข่าวสารต่างๆ
เพื่อให้คอมพิวเตอร์ใช้กระทำการ โดยจะต้อง
มีการกำหนดชื่อให้แฟ้ม เพื่อให้สามารถ
เรียกใช้ได้

เพิ่มข้อมูล (Data files)

ประกอบด้วยข้อมูล ได้แก่

- ตัวอักษร
- ข้อความ
- ตัวเลข
- รูปภาพ
- เสียง

faculty1 - Notepad

File Edit Format View Help

01 Agriculture

02 Fisheries

03 Forestry

04 Science

05 Engineering

06 Education

07 Economics

08 Social science

09 Humanities

10 Agro-Industry

11 Architecture

12 Veterinary Medicine

13 Business administration

student - Notepad

File Edit Format View Help

5410405550	อรชร	ยิ้มแย้ม	F 0819411115
5410307771	ธารินทร์	ประเสริฐกุล	M0819527766
5310806669	ศิริพร	สุขเสมอ	F 0898642225
5310806677	ปราโมทย์	มีไมตรี	M0851933066
5410609080	สมพล	ชยันดี	M0851076776
5411308039	มนัส	นิยมสุข	M0819211744
5410705445	จตุพล	ถาวรเกียรติ	M0847675089
5411053103	ลดาวัลย์	บวรกิจ	F 0857529003

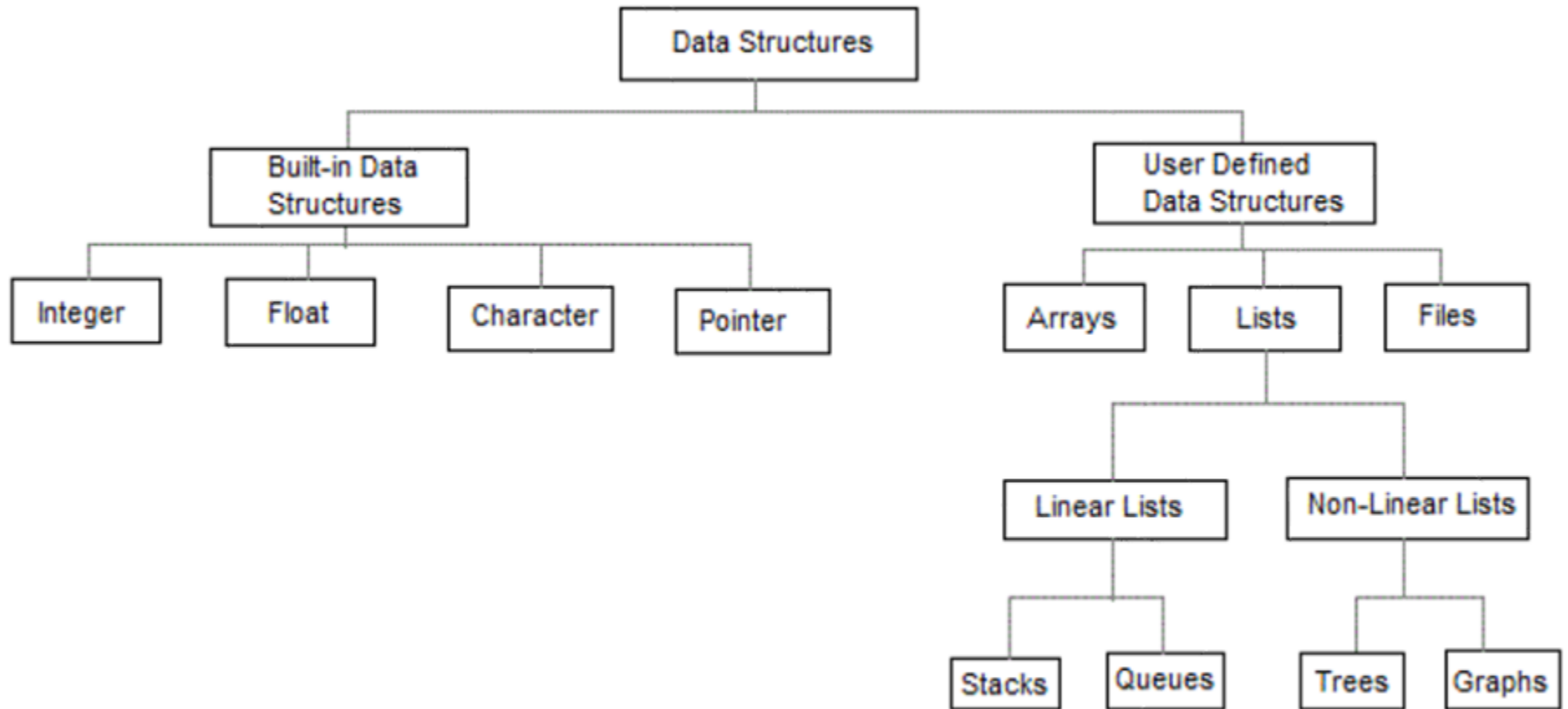
grade - Notepad

File Edit Format View Help

5410807231	542014181120138710201175101
5410307771	542014170140135511001387111
5310806669	542019990110141811201355102
5310806677	542014181130117511201355114
5411308039	542014182150138711401175102
5410609080	542014171110138721201175108
5410705445	542014181110135511301175105
5411053103	542014181140199901101355103

ประเภทโครงสร้างข้อมูล

- พื้นฐาน (Base data structures)
 - Primitive
- เชิงเส้น (Linear data structures)
 - Array
 - List
- ไม่เป็นเส้นตรง (Non Linear data structures)
 - Graph
 - Tree



INTRODUCTION TO DATA STRUCTURES

โครงสร้างข้อมูลพื้นฐาน (Base Data Structures)

ได้แก่

- จำนวนเต็ม ...-3, -2, -1, 0, 1, 2, 3
- จำนวนจริงที่มีทศนิยม 3.75, 3.75×10^0
- บูล มีค่าเป็น จริง หรือ เท็จ true, false
- อักขระ A, a, +, &, #, %,
- สายอักขระ (ข้อความหรือ **string**) I am lazy

ระเบียบวิธี

สายอักขระ

เลขประจำตัว (10)

จำนวนจริง

ระดับคะแนนเฉลี่ย (4)



ชื่อ (10)

สายอักขระ

สกุล (15)

สายอักขระ

หน่วยกิตสะสม (3)

จำนวนเต็ม

เพศ (1)

อักขระ

ระเบียบনীติ

สายอักขระ

เลขประจำตัว (10)

จำนวนจริง

ระดับคะแนนเฉลี่ย (4)

5310807231	สันติภาพ	เอกธำรงกุล	M	2.65	108
------------	----------	------------	---	------	-----

ชื่อ (10)

สายอักขระ

สกุล (15)

สายอักขระ

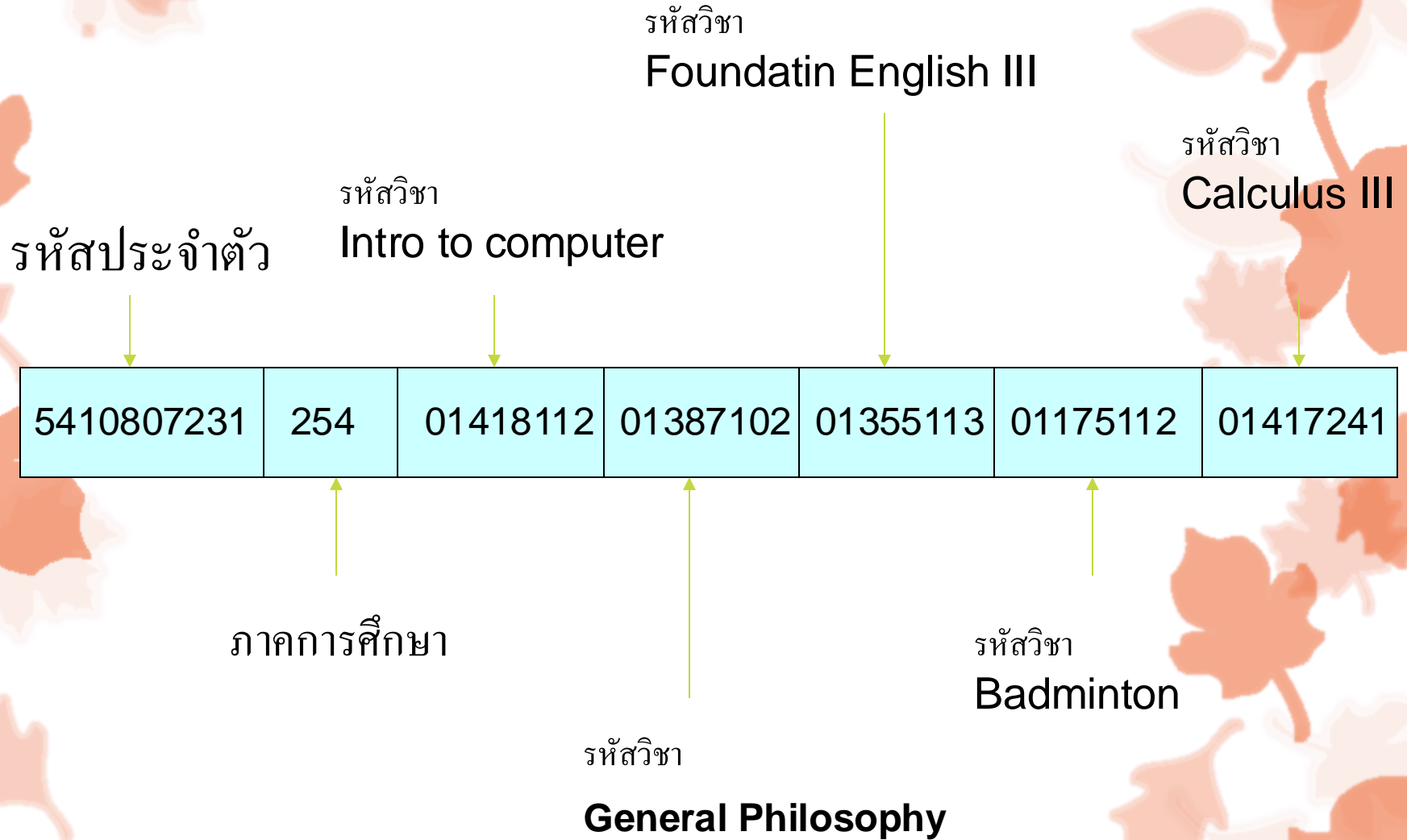
หน่วยกิตสะสม (3)

จำนวนเต็ม

เพศ (1)

อักขระ

ระเบียบหลักสูตร



รหัสวิชา
Foundatin English III

รหัสวิชา
Calculus III

รหัสวิชา
Intro to computer

รหัสประจำตัว

5410807231	254	01418112	01387102	01355113	01175112	01417241
------------	-----	----------	----------	----------	----------	----------

ภาคการศึกษา

รหัสวิชา
Badminton

รหัสวิชา
General Philosophy

รหัสประจำตัว

Grade-com ... Grade-calculus

5410807231	254	2.5	3.0	1.5	4.0	2.0
------------	-----	-----	-----	-----	-----	-----

ภาคการศึกษา

get Intro_computer
 get grade-com
 get Philosophy
 get grade-philo
 get English
 get grade-eng
 get Badminton
 get grade-bad
 get Calculus
 get grade-calculus

อัลกอริทึม (ALGORITHM)

เป็นวิธีการแสดงลำดับขั้นตอนในการทำงานหรือแก้ไขปัญหาอย่างใดอย่างหนึ่ง เช่นการกำหนดขั้นตอนเพื่อแก้ปัญหาการจัดเรียงเอกสารในแฟ้มข้อมูล หรือการกำหนดอัลกอริทึมในการค้นหาข้อมูลในแฟ้มข้อมูลทั้งหมด เป็นต้น

อัลกอริทึมที่ดีควรมีคุณสมบัติ ดังนี้

1. มีลำดับขั้นตอนทำงานก่อน-หลัง ชัดเจน
2. เข้าใจง่ายและไม่กำกวม
3. สามารถประมวลผลการทำงานด้วยคอมพิวเตอร์ได้
4. การทำงานของอัลกอริทึมจะต้องสิ้นสุด หลังจากดำเนินงานตามระยะเวลาที่กำหนด

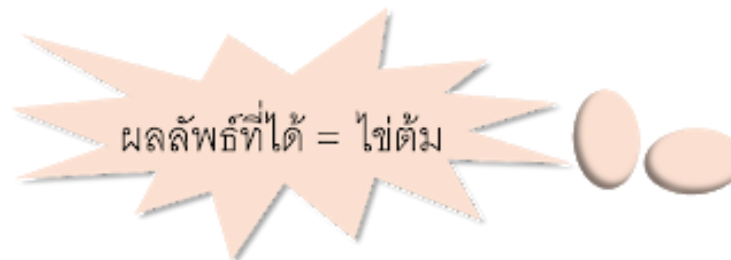
ตัวอย่างอัลกอริทึม : การต้มไข่ไก่

• วิธีที่ 1

- 1 ต้มน้ำให้เดือด
- 2 ใส่ไข่
- 3 รอ 10 นาที
- 4 ดับไฟ / ปิดเตา
- 5 ปอกไข่

• วิธีที่ 2

- 1 ต้มน้ำให้เดือด
- 2 ใส่ไข่
- 3 รอ 5 นาที
- 4 ดับไฟ / ปิดเตา



Python Intro

- Python คือ programming language ที่ได้รับความนิยมมาก ถูกพัฒนาโดย Guido van Rossum ในปี 1991
- ใช้ในเรื่อง
 - web development (server-side),
 - software development,
 - mathematics,
 - system scripting.
- Python ทำอะไรได้บ้าง
 - used on a server to create web applications
 - used alongside software to create workflows.
 - connect to database systems. It can also read and modify files.
 - used to handle big data and perform complex mathematics
 - used for rapid prototyping, or for production-ready software development

Python Intro

- ทำไมต้องใช้ล่ะ ?
 - works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
 - simple syntax similar to the English language
 - syntax that allows developers to write programs with fewer lines than some other programming languages.
 - runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick
 - treated in a procedural way, an object-oriented way or a functional way.

Python Install

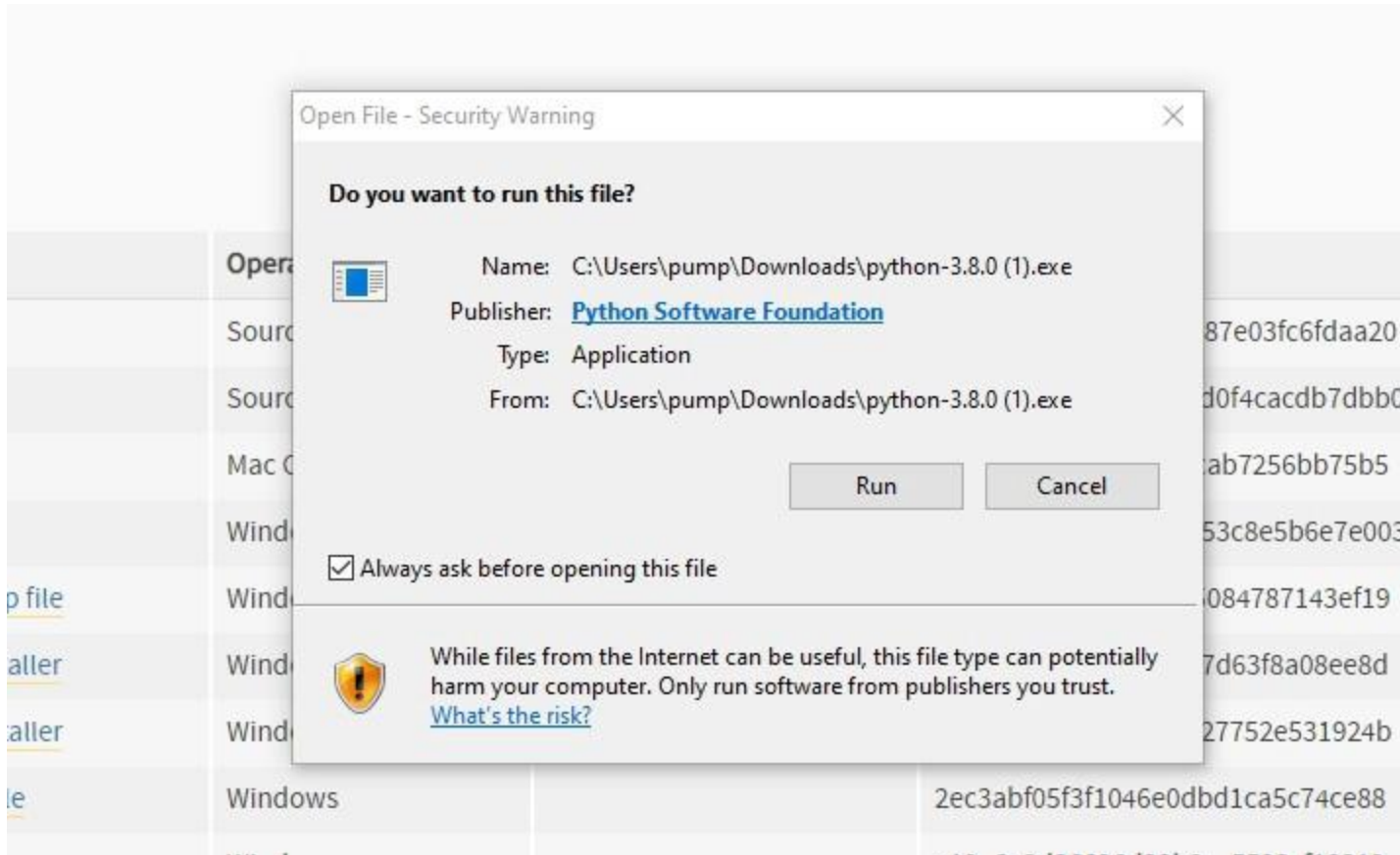
- ขั้นตอนการติดตั้งไพทอน เวอร์ชัน 3.8 ลงบน Windows 10
 - 1. ดาวน์โหลดไฟล์ติดตั้งของไพทอน
จาก <https://www.python.org/downloads/>
 - 2. เลือก "Python 3.8" จากนั้นกด "Download"
 - 3. เมื่อกดคำว่า "Download" แล้วให้เลือก "Windows x86-64 executable Installer"

Files

Version	Operating System	Description	MD5 Sum	File Size	PGP
Gzipped source tarball	Source release		e18a9d1a0a6d858b9787e03fc6fdaa20	23949883	SIG
XZ compressed source tarball	Source release		dbac8df9d8b9edc678d0f4cacdb7dbb0	17829824	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	f5f9ae9f416170c6355cab7256bb75b5	29005746	SIG
Windows help file	Windows		1c33359821033ddb3353c8e5b6e7e003	8457529	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	99cca948512b53fb165084787143ef19	8084795	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	29ea87f24c32f5e924b7d63f8a08ee8d	27505064	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	f93f7ba8cd48066c59827752e531924b	1363336	SIG
Windows x86 embeddable zip file	Windows		2ec3abf05f3f1046e0dbd1ca5c74ce88	7213298	SIG
Windows x86 executable installer	Windows		412a649d36626d33b8ca5593cf18318c	26406312	SIG
Windows x86 web-based installer	Windows		50d484ff0b08722b3cf51f9305f49fdc	1325368	SIG

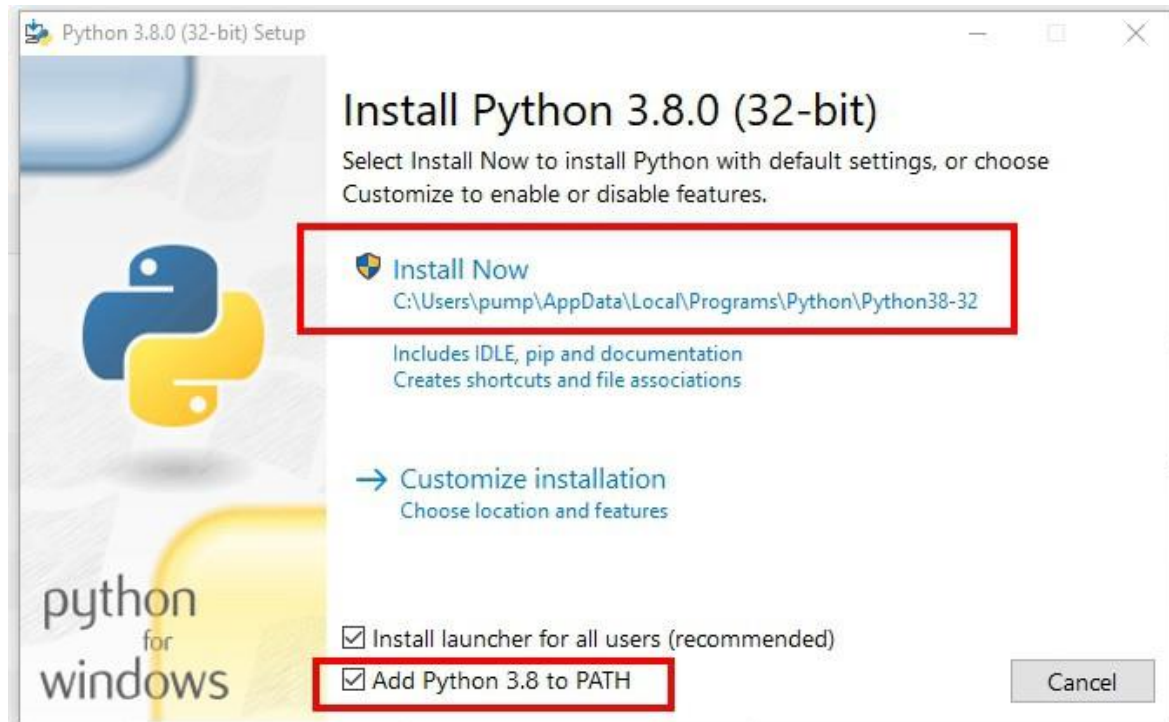
Python Install (ต่อ)

- 4. ติดตั้งกด run



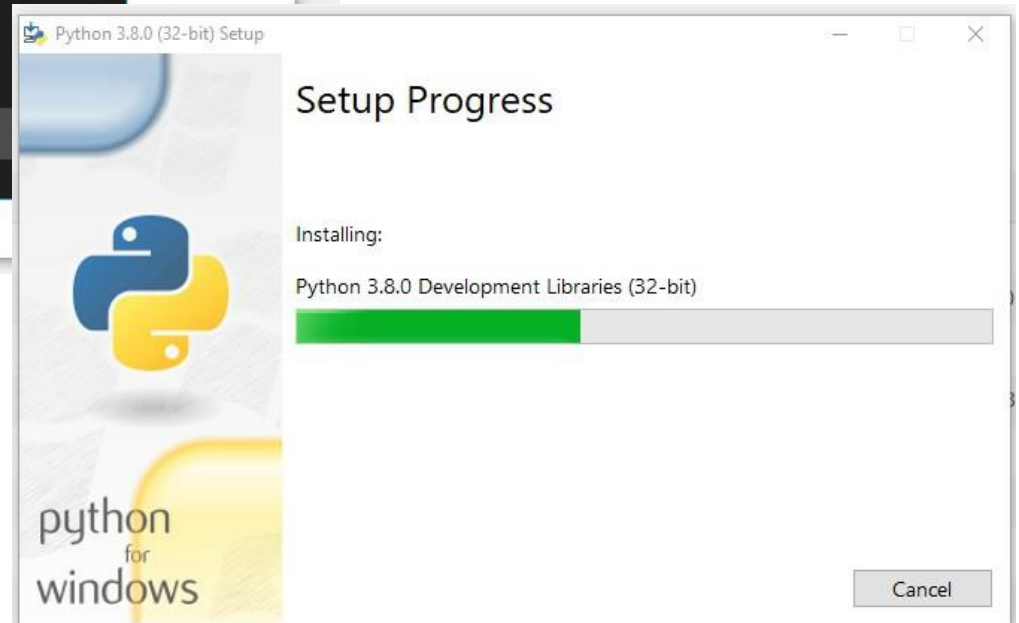
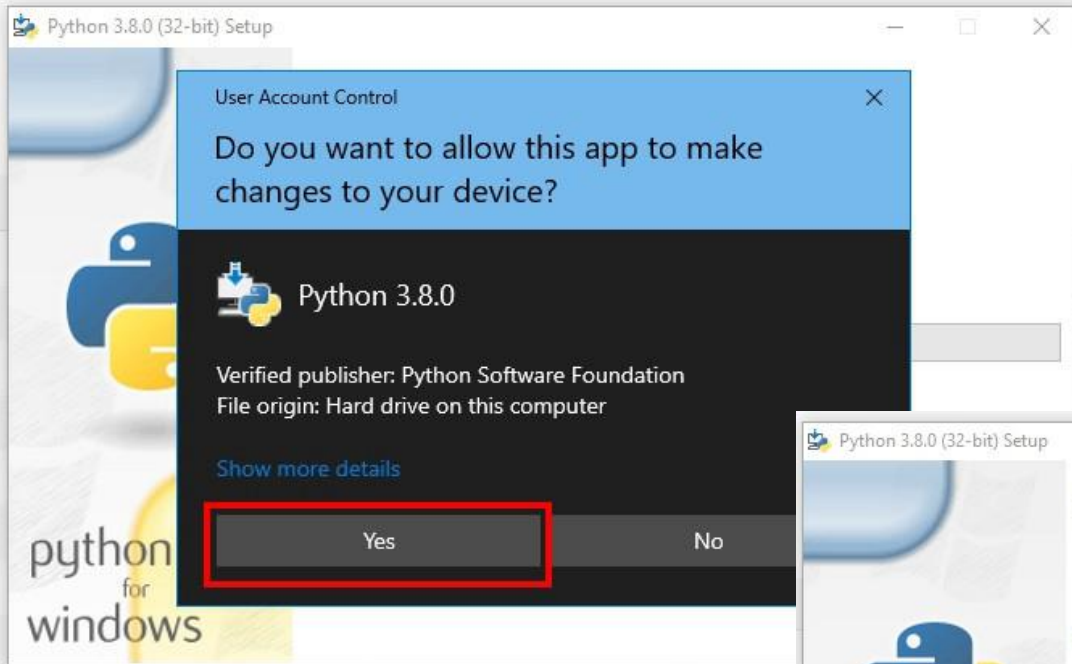
Python Install (ต่อ)

- หลังจากนั้นหน้าต่างของการติดตั้งจะปรากฏขึ้นมา คลิกเลือกที่ “Add Python 3.8 to PATH” เพื่อให้ระบบทำการกำหนด PATH เพื่อให้ภาษา Python สามารถทำงานได้กับ Command line อัตโนมัติในทุกที่ คลิกที่ “Install now “ เพื่อเริ่มการติดตั้งภาษา Python คุณอาจจะเปลี่ยนแปลงตัวเลือกสำหรับการติดตั้งด้วยตัวเองโดยเลือกที่ “Customize install” เช่น เปลี่ยนสถานที่ที่ต้องการติดตั้ง เป็นต้น



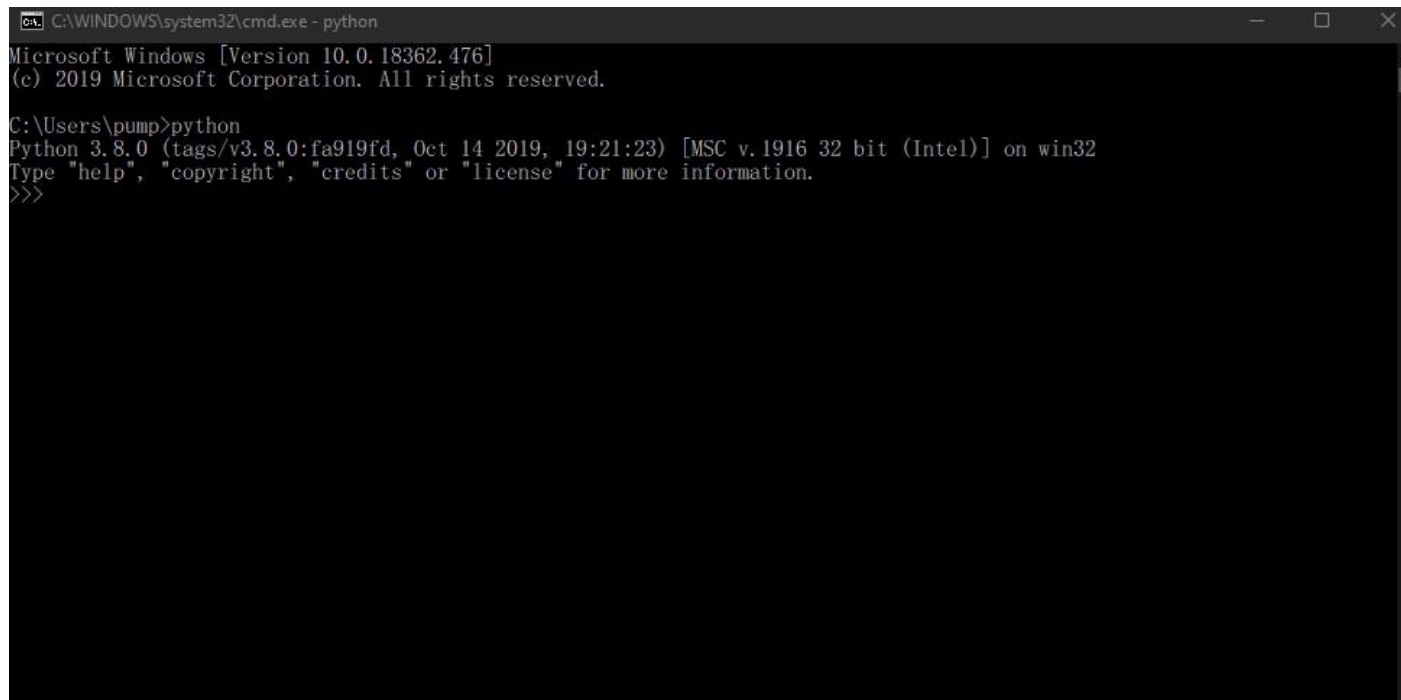
Python Install (ต่อ)

- กด Yes เพื่อยอมรับการติดตั้ง



Python Install (ต่อ)

- เราได้ติดตั้งภาษา Python เรียบร้อยแล้วและคุณพร้อมที่จะเขียนโปรแกรมในภาษา Python ต่อไปเราจะทำการตรวจสอบการติดตั้ง โดยให้คุณเปิด Command line ขึ้นมา และพิมพ์คำสั่ง “python” ซึ่งจะแสดงผลดังนี้ ซึ่งถือว่าการติดตั้งเสร็จสมบูรณ์ และจะเป็นการเข้าสู่ Interactive shell ของภาษา Python ที่คุณสามารถพิมพ์คำสั่งต่างๆ และดูผลลัพธ์การทำงานได้ทันที

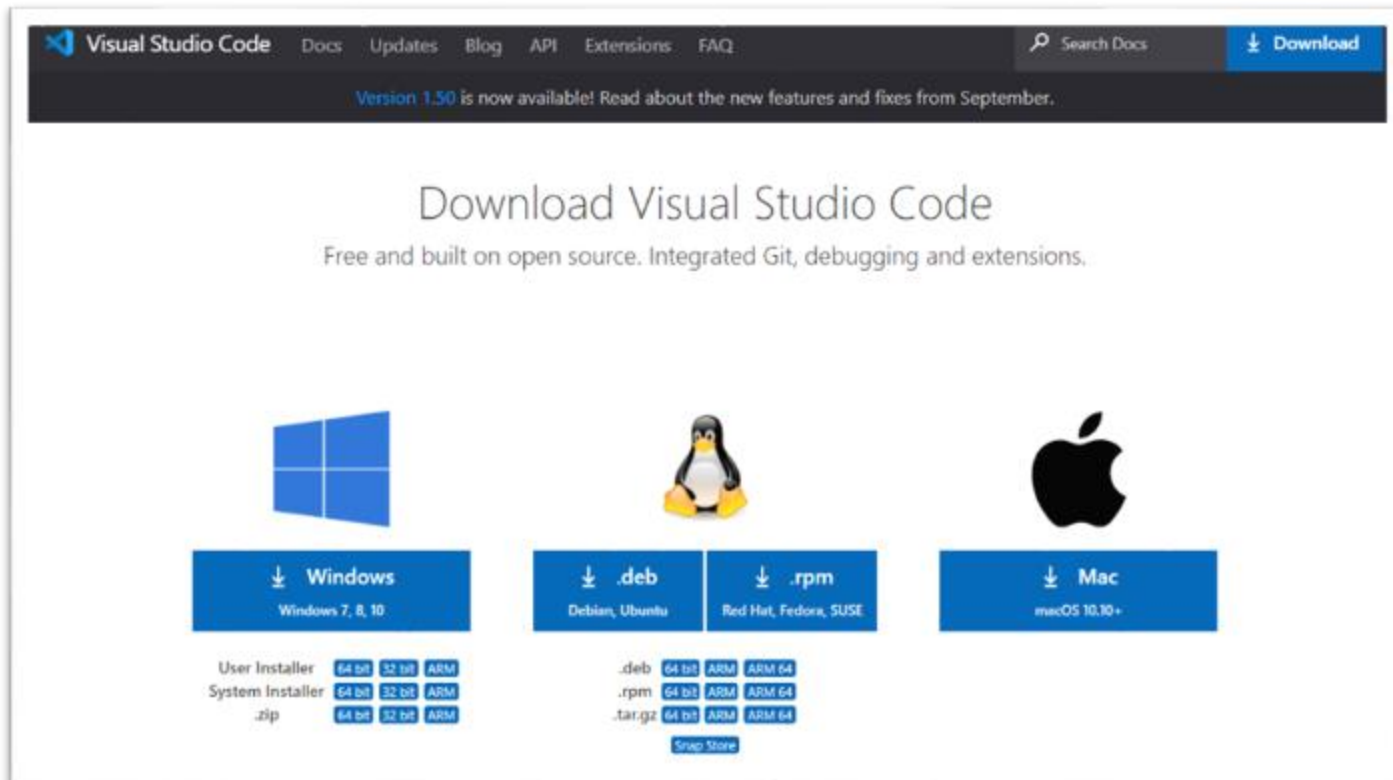


```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pump>python
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Install Visual Studio Code

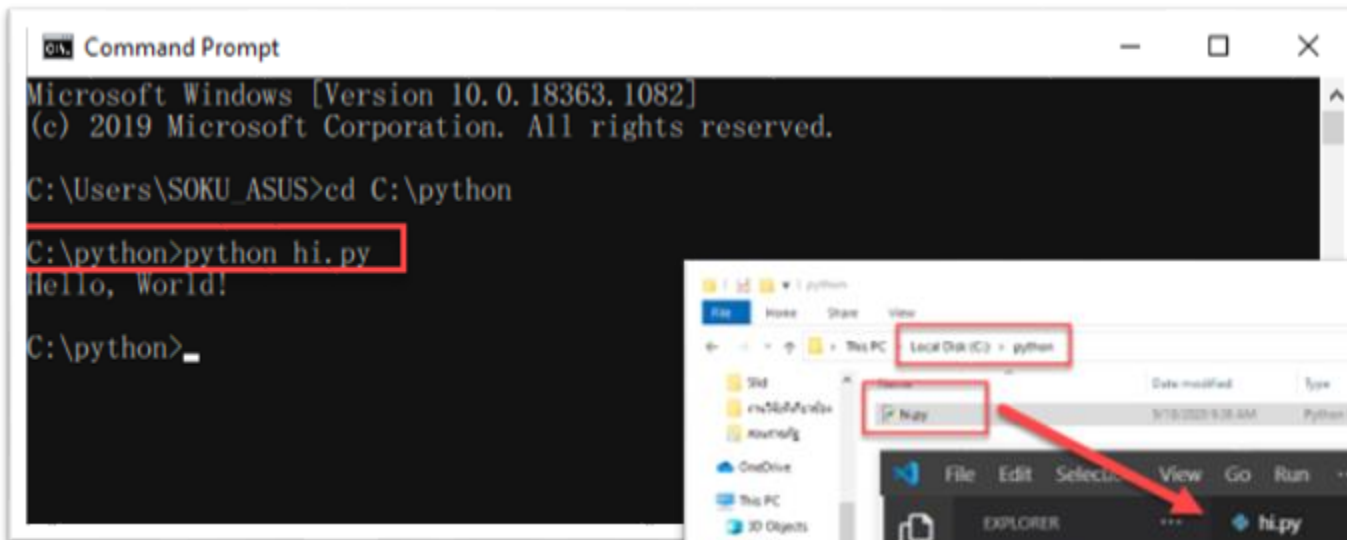
- ง่าย text editor & Code editor
- Free & Support windows, linux, mac os
- Download <https://code.visualstudio.com/download>



The screenshot shows the Visual Studio Code website's download page. At the top, there is a navigation bar with links for 'Visual Studio Code', 'Docs', 'Updates', 'Blog', 'API', 'Extensions', and 'FAQ'. A search bar for 'Search Docs' and a 'Download' button are also present. A banner below the navigation bar announces 'Version 1.50 is now available! Read about the new features and fixes from September.' The main heading is 'Download Visual Studio Code', followed by the tagline 'Free and built on open source. Integrated Git, debugging and extensions.' Below this, there are three main sections for operating systems: Windows, Linux, and Mac. The Windows section features the Windows logo and a blue button with a download icon, 'Windows', and 'Windows 7, 8, 10'. Underneath, it lists 'User Installer', 'System Installer', and '.zip' with buttons for '64 bit', '32 bit', and 'ARM'. The Linux section is split into two columns: one for '.deb' (Debian, Ubuntu) and one for '.rpm' (Red Hat, Fedora, SUSE). Each column lists '.deb', '.rpm', and '.tar.gz' with buttons for '64 bit', 'ARM', and 'ARM 64'. A 'Snap Store' button is located at the bottom of the Linux section. The Mac section features the Apple logo and a blue button with a download icon, 'Mac', and 'macOS 10.10+'. The entire page is framed by a thin border.

การรันโปรแกรมภาษา Python

- การรันโปรแกรมที่เขียนในภาษา Python นั้นไฟล์จะต้องถูกบันทึกในนามสกุล .py เพื่อโปรแกรมในภาษา Python เปิด Command line และใช้คำสั่ง python และตามด้วยที่อยู่ของไฟล์ดังกล่าวอย่างข้างล่างนี้
- python c:\python\hi.py (ต้องมีไฟล์ hi.py อยู่ก่อน)

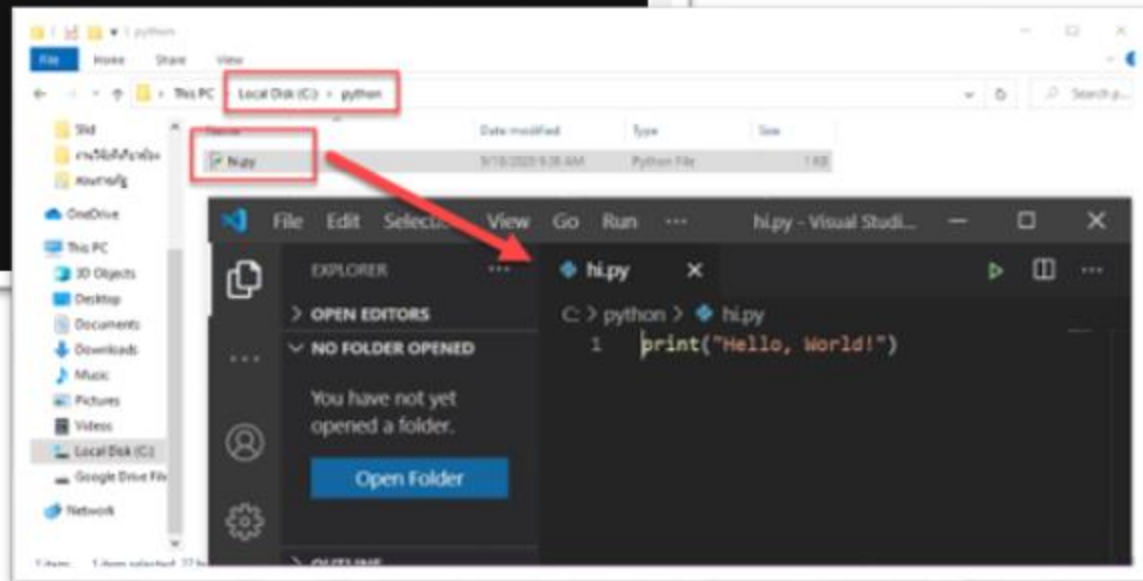


```
Command Prompt
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\SOKU_ASUS>cd C:\python

C:\python>python hi.py
Hello, World!

C:\python>
```



Python Syntax

- Python Indentation
- การเว้นวรรคของภาษา python สำคัญมาก

```
if 5 > 2:  
    print("Five is greater than two!")
```

Syntax Error:

```
if 5 > 2:  
print("Five is greater than two!")
```

```
if 5 > 2:  
    print("Five is greater than two!")  
if 5 > 2:  
    print("Five is greater than two!")
```



```
Five is greater than two!  
Five is greater than two!
```

```
if 5 > 2:  
    print("Five is greater than two!")  
    print("Five is greater than two!")
```

Error

```
File "demo_indentation2_error.py", line 3  
    print("Five is greater than two!")  
    ^  
IndentationError: unexpected indent
```

Python Comments

```
#This is a comment.  
print("Hello, World!")
```



```
Hello, World!
```

```
print("Hello, World!") #This is a comment.
```



```
Hello, World!
```

```
""  
This is a comment  
written in  
more than just one line  
""  
print("Hello, World!")
```



```
Hello, World!
```


Python Variables

```
x = 5  
y = "John"  
print(x)  
print(y)
```



```
5  
John
```

```
x = 4  
x = "Sally"  
print(x)
```



```
Sally
```

```
x = "John"  
print(x)  
#double quotes are the same as single quotes:  
x = 'John'  
print(x)
```



```
John  
John
```


Python Variables (ต่อ)

- ชื่อตัวแปรเป็น case-sensitive คือต้องระวังในการตั้งชื่อ

```
#Legal variable names:  
myvar = "John"  
my_var = "John"  
_my_var = "John"  
myVar = "John"  
MYVAR = "John"  
myvar2 = "John"  
  
#Illegal variable names:  
2myvar = "John"  
my-var = "John"  
my var = "John"
```

- สามารถใส่ค่าแบบเรียงกันได้

```
x, y, z = "Orange", "Banana", "Cherry"  
  
print(x)  
print(y)  
print(z)
```




```
Orange  
Banana  
Cherry
```

Python Variables (ต่อ)

- หรือใส่ค่าเดียวกันทั้ง 3 ตัวได้


```
x = y = z = "Orange"
print(x)
print(y)
print(z)
```



```
Orange
Orange
Orange
```

- การแสดง output สามารถนำ text + variable ได้

```
x = "awesome"
print("Python is " + x)
```



```
Python is awesome
```

- สามารถกำหนดตัวแปรอื่นมารับค่าใหม่ก่อนได้

```
x = "Python is "
y = "awesome"
z = x + y
print(z)
```



```
Python is awesome
```

Python Variables (ต่อ)

- ข้อความบวกตัวเลขต้องระวังจะเกิด Error ได้

```
x = 5
y = "John"
print(x + y)
```



```
TypeError: unsupported operand type(s) for +: 'int'
```

- มารดสร้างตัวแปรแบบ Global ได้

```
x = "awesome"

def myfunc():
    print("Python is " + x)

myfunc()
```



```
Python is awesome
```

- ตัวแปร Global บางตัวอาจจะทำหน้าที่เป็นตัวแปรธรรมดาในฟังก์ชันอื่นก็ได้ เพราะฉะนั้นต้องระวัง

```
x = "awesome"

def myfunc():
    x = "fantastic"
    print("Python is " + x)

myfunc()

print("Python is " + x)
```

```
Python is fantastic
Python is awesome
```

Python Variables (ต่อ)

การแก้ปัญหาตัวแปร Global คือกำหนดเลยว่าเป็น Global

```
x = "awesome"

def myfunc():
    global x
    x = "fantastic"

myfunc()

print("Python is " + x)
```



```
Python is fantastic
```

Python Data Types

Data Type ใน Python จะถูก Built-in มาให้ใช้งานจะแยกตามหมวดหมู่ดังภาพด้านล่าง

```
Text Type:      str
Numeric Types:  int, float, complex
Sequence Types: list, tuple, range
Mapping Type:   dict
Set Types:      set, frozenset
Boolean Type:   bool
Binary Types:   bytes, bytearray, memoryview
```

สามารถตรวจสอบ type ได้

```
x = 5
print(type(x))
```



```
<class 'int'>
```

Python Data Types (๓๑)

Example	Data Type
<code>x = "Hello World"</code>	str
<code>x = 20</code>	int
<code>x = 20.5</code>	float
<code>x = 1j</code>	complex
<code>x = ["apple", "banana", "cherry"]</code>	list
<code>x = ("apple", "banana", "cherry")</code>	tuple
<code>x = range(6)</code>	range
<code>x = {"name" : "John", "age" : 36}</code>	dict
<code>x = {"apple", "banana", "cherry"}</code>	set
<code>x = frozenset({"apple", "banana", "cherry"})</code>	frozenset
<code>x = True</code>	bool
<code>x = b"Hello"</code>	bytes
<code>x = bytearray(5)</code>	bytearray
<code>x = memoryview(bytes(5))</code>	memoryview

Python Data Types (ต่อ)

สามารถกำหนด data type เลย

Example	Data Type
<code>x = str("Hello World")</code>	str
<code>x = int(20)</code>	int
<code>x = float(20.5)</code>	float
<code>x = complex(1j)</code>	complex
<code>x = list(("apple", "banana", "cherry"))</code>	list
<code>x = tuple(("apple", "banana", "cherry"))</code>	tuple
<code>x = range(6)</code>	range
<code>x = dict(name="John", age=36)</code>	dict
<code>x = set(("apple", "banana", "cherry"))</code>	set
<code>x = frozenset(("apple", "banana", "cherry"))</code>	frozenset
<code>x = bool(5)</code>	bool
<code>x = bytes(5)</code>	bytes
<code>x = bytearray(5)</code>	bytearray
<code>x = memoryview(bytes(5))</code>	memoryview

ได้ค่าอะไร ?

Python Operation

- Python Casting การคอนเวิร์ตค่า
- Python Operators
- Python User Input
- Python If ... Else
- Python For Loops
- อาร์เรย์ (Array)
 - Python Collections (Arrays)
 - List
 - Tuple
 - Set
 - Dictionary
- Python Coding Example

Python Casting

- ปัญหาการนำ string กับตัวเลขมาบวกกัน โดยที่ไม่แปลงก่อน

- เช่น

```
x = "hello"
```

```
y = 5
```

```
print(x+y)
```

- จะสังเกตเห็นว่ามันบวกกันไม่ได้จะ Error เพราะฉะนั้นต้องทำให้ค่าที่นำมาบวกกันเป็นประเภทเดียวกันก่อน จะเอื้อมความบวกกันเพื่อต่อข้อความ หรือตัวเลขบวกกัน

- ในเคสด้านบนจะทำได้แค่แปลงเป็นข้อความแล้วบวกกัน

```
x = "hello"
```

```
y = str(5)
```

```
print(x+y)
```

- int() แปลงเป็นจำนวนเต็ม, float() แปลงเป็นทศนิยม, str() แปลงเป็นข้อความ

Python Operators

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

Python Operators (๓๑)

x=6

y=2

z=3

print(x+y)

print(x-z)

print(y*z)

print(x/z)

print(z%y)

print(z**y)

print(x//y)

Python Comparison Operators

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

x=5

y=3

print(x==y)

print(x!=y)

print(x>y)

print(x<=y)

Python Logical Operators

Operator	Description	Example
and	Returns True if both statements are true	<code>x < 5 and x < 10</code>
or	Returns True if one of the statements is true	<code>x < 5 or x < 4</code>
not	Reverse the result, returns False if the result is true	<code>not(x < 5 and x < 10)</code>

```
x = 5
```

```
print(x > 3 and x < 10)
```

```
print(x > 3 or x < 4)
```

```
print(not(x > 3 and x < 10))
```

Python Membership Operators

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

```
x = ["apple", "banana"]
```

```
print("banana" in x)
```

```
print("pineapple" not in x)
```

Python User Input การรับค่า input

- Python 3.6 uses the input() method.
- Python 2.7 uses the raw_input() method.

```
username = raw_input("Enter username:")
```

```
print("Username is: " + username)
```

```
C: > python > test.py > ...
1  username = input("Enter username:")
2  print("Username is: " + username)
3
4
5
6
7
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
PS C:\ionic\app_sidemenu> & C:/Users/SOKU_ASUS/AppData/Local/Programs/
Enter username:ssss
Username is: ssss
PS C:\ionic\app_sidemenu> 
```


Python If ... Else

```
a = 200
```

```
b = 33
```

```
if b > a:
```

```
    print("b is greater than a")
```

```
elif a == b:
```

```
    print("a and b are equal")
```

```
else:
```

```
    print("a is greater than b")
```

จะทำจาก IF ถ้าไม่ตรงเงื่อนไขจะมาทำ elif จะมีกี่ครั้งก็ได้ สุดท้ายจะจบที่ else คือ นอกเหนือทั้งหมดจะออกมาอยู่ที่ else

Python If ... Else (๓๑)

- Short Hand If

```
if a > b: print("a is greater than b")
```

- Short Hand If ... Else

```
a = 2
```

```
b = 330
```

```
print("A") if a > b else print("B")
```

LAB 1

- สร้างฟังก์ชัน IF ELSE คำนวณ เกรดนักศึกษา โดยให้กรอกตัวเลขโดยระบุ input score : แล้วเช็คดังนี้

80 ขึ้นไป	A
75-79	B+
70-74	B
65-69	C+
60-64	C
55-59	D+
50-54	D
49 ลงไป	F

ตัวอย่าง OUT PUT

```
PS C:\ionic\app_sidemenu> & C:/Users/SOKU_ASUS/AppData/Local/Programs/Python/Python39/python.exe c:/python/test.py
input score:45
grade is: F
PS C:\ionic\app_sidemenu> & C:/Users/SOKU_ASUS/AppData/Local/Programs/Python/Python39/python.exe c:/python/test.py
input score:70
grade is: B
PS C:\ionic\app_sidemenu> 
```

Python For Loops เป็นการเขียนโคดวนลูป

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```

```
for x in range(10):  
    if x == 6:  
        print(x)  
    else :  
        print("-")
```

โครงสร้างข้อมูลแบบแถวลำดับหรืออาร์เรย์ (Array)

ความหมายของแถวลำดับ

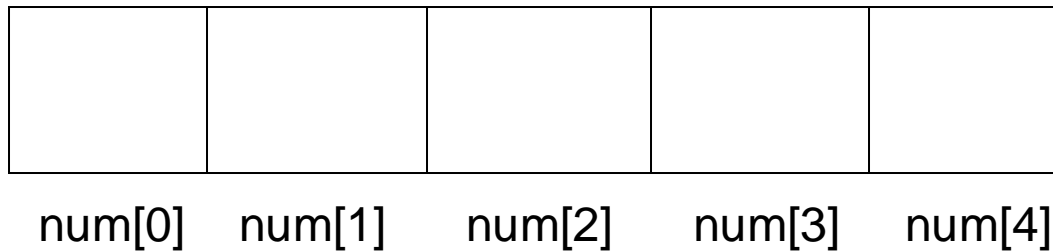
- เป็นโครงสร้างข้อมูลที่มีการจองพื้นที่หน่วยความจำ (Memory) เป็นชุด ๆ แต่ละชุดประกอบด้วยจำนวนช่องข้อมูลหลายช่อง พื้นที่แต่ละช่องข้อมูลจะเก็บข้อมูลชนิดเดียวกัน และอยู่ในตำแหน่งที่ต่อเนื่องกันไปตามลำดับ
- การเข้าถึงข้อมูล (Access) ใด ๆ ในโครงสร้างอาร์เรย์ สามารถกระทำได้โดยการระบุหมายเลขกำกับช่องข้อมูล ที่เรียกว่า ตัวดัชนี (Index) หรือบางครั้งเรียกว่า ตัวห้อย หรือซบสคริปต์ (Sub Script) เป็นตัวอ้างอิงตำแหน่งสมาชิกบนแถวลำดับ เช่น A(3) หมายถึง สมาชิกอาร์เรย์ A ลำดับที่ 3

ชนิดของอาร์เรย์

- อาร์เรย์ 1 มิติ
- อาร์เรย์ 2 มิติ
- อาร์เรย์หลายมิติ (มากกว่า 2 มิติ)

อาร์เรย์หนึ่งมิติ (One Dimension Array)

- มีการจัดเก็บข้อมูลในลักษณะต่อเนื่องกันเป็นแถว ซึ่งจะนำเสนอในมุมมองแบบแนวตั้ง และแนวนอนก็ได้
- สัญลักษณ์ที่ใช้คือ `array_name[size]`
- เช่น `num[5]` หมายถึง `num` เป็นอาร์เรย์ 1 มิติขนาด 5 (มีสมาชิก 5 ตัว หรือ 5 ช่อง)



-
- ในภาษาคอมพิวเตอร์ทั่วไป ค่าดัชนี จะเริ่มต้นจาก 0 เช่น ภาษา C , C++ java และวงเล็บที่อยู่หลังตัวแปรอาร์เรย์ จะใช้เครื่องหมาย [] หรือ () ขึ้นอยู่กับ ภาษานั้น เช่น c , c++ , java ใช้เครื่องหมาย [] visual basic ใช้เครื่องหมาย ()
 - ตัวอย่างการประกาศอาร์เรย์ 1 มิติในภาษา C , C++

```
int num[20];
```

```
char grade[10];
```

-
- ตัวอย่าง การประกาศอาร์เรย์ 1 มิติ ในภาษา Visual Basic

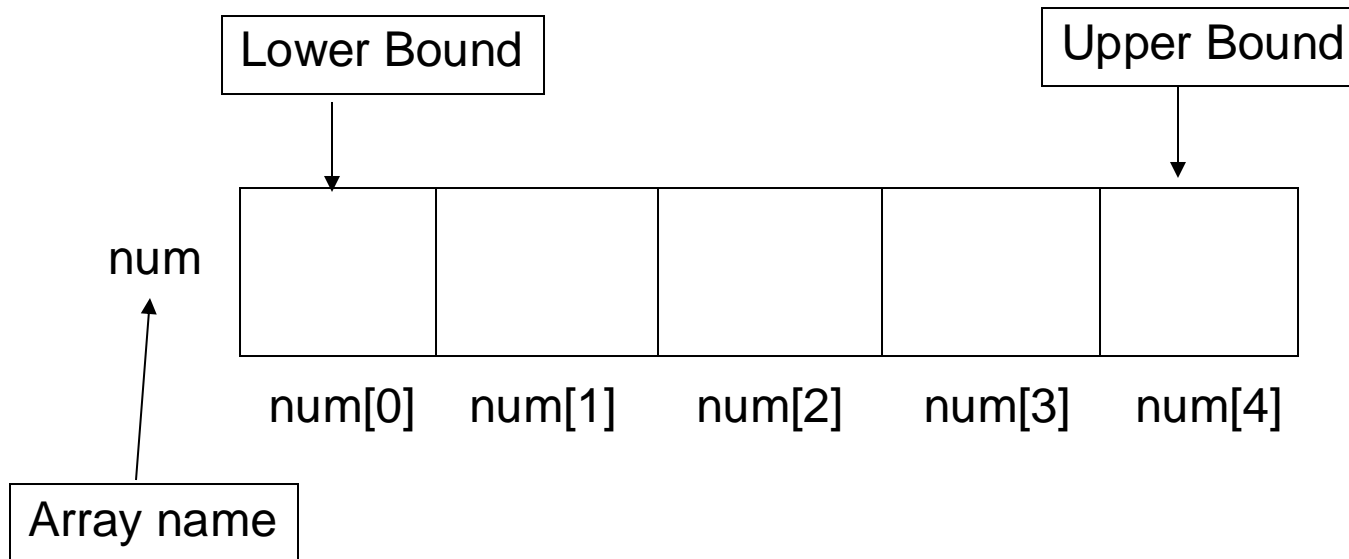
```
dim num(5) as integer
```

- ตัวอย่าง การประกาศอาร์เรย์ 1 มิติ ในภาษา Python

```
a = [1,2,3,4,5,6]
```

ขอบเขตของอาร์เรย์ 1 มิติ

- เลขดัชนีในอาร์เรย์ประกอบด้วยช่วงขอบเขตของค่าซึ่งประกอบด้วยขอบล่างสุด (Lower Bound) และขอบเขตบนสุด (Upper Bound)



ดังนั้น **num(5)** สามารถเขียนเป็น **num[1:5]**

การคำนวณหาจำนวนสมาชิกของอาร์เรย์หนึ่งมิติ

- รูปแบบ `ArrayName [L:U]`

โดย `ArrayName` คือ ชื่อของอาร์เรย์

`L` คือ ขอบเขตล่างสุด (Lower Bound)

`U` คือ ขอบเขตบนสุด (Upper Bound)

- จำนวนสมาชิก = $U-L+1$
- เช่น `num[1:5]` มีสมาชิก เท่ากับ $5-1+1 = 5$ ตัว

การคำนวณหาตำแหน่ง(Address)ในหน่วยความจำของอาร์เรย์หนึ่งมิติ

- หาได้จากสูตร

$$\text{Loc}(A[i]) = B + w * (i - L)$$

โดยที่

i : ตำแหน่งที่ต้องการ

B : ตำแหน่งเริ่มต้นในหน่วยความจำ

L : ตำแหน่งเริ่มต้น

w : ขนาดความกว้างของชนิดข้อมูล

ตัวอย่าง การหาตำแหน่งในหน่วยความจำ ของอาร์เรย์หนึ่งมิติ

กำหนด อาร์เรย์ $\text{Data}[1:5]$ เก็บข้อมูลชนิดตัวอักษร ซึ่งใช้เนื้อที่ในการเก็บข้อมูล 2 bytes ต่อชุด โดยมีตำแหน่งเริ่มต้นในหน่วยความจำอยู่ที่ 1000
จงหาตำแหน่งที่ใช้เก็บข้อมูลของ $\text{Data}[2]$

แทนค่าดังนี้

$i = 2, B = 1000, L = 1, w = 2$

$$\begin{aligned} \text{จะได้ } \text{Loc}(\text{Data}[2]) &= 1000 + 2 * (2 - 1) \\ &= 1000 + 2 * 1 \\ &= 1000 + 2 \\ &= 1002 \end{aligned}$$

Address	Data
1000	
1001	
1002	
1003	
⋮	
1009	

Labels on the right side of the table: Data[1] (next to 1000), Data[2] (next to 1002), Data[3] (next to 1003), Data[4] (next to ⋮), Data[5] (next to 1009)

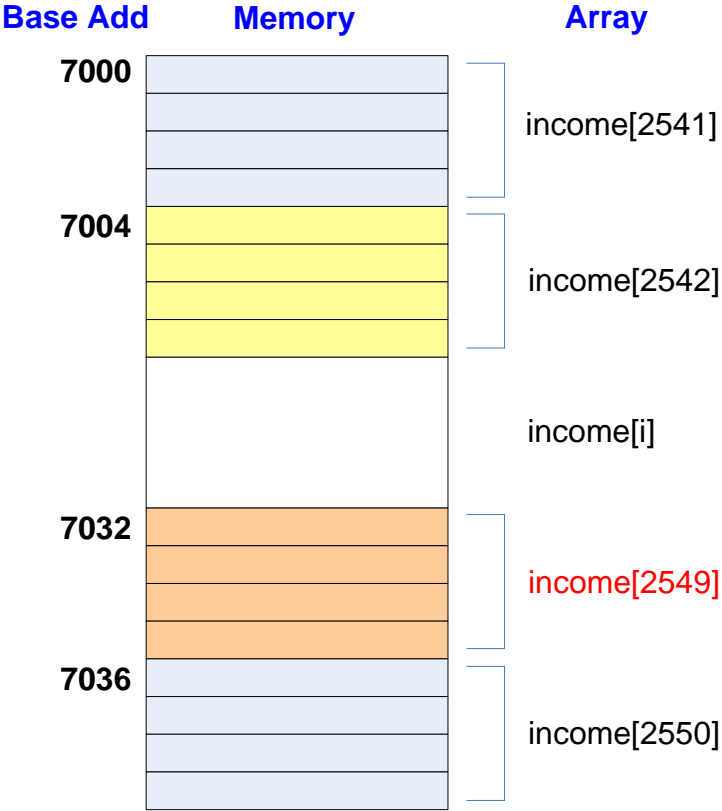
ตัวอย่าง การหาตำแหน่งในหน่วยความจำ ของอาร์เรย์หนึ่งมิติ (ต่อ)

ต้องการประกาศตัวแปรอาร์เรย์ income เพื่อจัดเก็บยอดรายได้ของ ปี พ.ศ. 2541-2550 ซึ่งเป็นไปตามรูปแบบ income[2541:2550] ถ้าข้อมูลที่จัดเก็บกินเนื้อที่ 4 ไบต์ต่อสมาชิก 1 ตัว และสมาชิกตัวแรกจัดเก็บในตำแหน่งที่ 7000 จงคำนวณหาแอดเดรสที่จัดเก็บยอดรายได้ของปี 2549

แทนค่าดังนี้

$$i = 2549, B = 7000, L = 2541, w = 4$$

$$\begin{aligned} \text{จะได้ } \text{LOC}(\text{income}[2549]) &= 7000 + 4 * (2549 - 2541) \\ &= 7000 + 4 * 8 \\ &= 7000 + 32 \\ &= 7032 \end{aligned}$$



อาร์เรย์สองมิติ (Two Dimension Array)

- โครงสร้างข้อมูลที่มีการจัดเก็บข้อมูลแบบตารางสองทาง ข้อมูลมีการจัดเรียงกันตามแถวแถว (Row) และ แถวหลัก (Column) การอ้างถึงข้อมูลต้องระบุตำแหน่งแถวและตำแหน่งหลักที่ข้อมูลนั้นอยู่

	Col 1	Col 2	Col 3	Col 4
Row 1	Data	Data	Data	Data
Row 2	Data	Data	Data	Data
Row 3	Data	Data	Data	Data
Row 4	Data	Data	Data	Data

รูปแบบทั่วไปของโครงสร้างข้อมูลอาร์เรย์ 2 มิติ

ArrayName[L₁ : U₁ , L₂ : U₂]

เมื่อ ArrayName คือ ชื่อของโครงสร้างข้อมูลอาร์เรย์

L₁ คือ ค่าขอบเขตล่างสุด (Lower Bound) ของแถว

U₁ คือ ค่าขอบเขตสูงสุด (Upper Bound) ของแถว

L₂ คือ ค่าขอบเขตล่างสุด (Lower Bound) ของคอลัมน์

U₂ คือ ค่าขอบเขตสูงสุด (Upper Bound) ของคอลัมน์

อาร์เรย์สองมิติ

- เช่น $K[4,3]$ หรือ $K[0:3,0:2]$

		Columns		
		0	1	2
Rows	0	10	5	3
	1	2	1	9
	2	11	6	7
	3	0	4	3

การคำนวณหาจำนวนสมาชิกของอาร์เรย์สองมิติ

- จำนวนสมาชิก = $(U_1 - L_1 + 1) * (U_2 - L_2 + 1)$

โดย U_1 = ขอบเขตบนสุด ของแถว

L_1 = ขอบเขตล่างสุด ของแถว

U_2 = ขอบเขตบนสุด ของคอลัมน์

L_2 = ขอบเขตล่างสุด ของคอลัมน์

เช่น $A[1:2,1:3]$ มีจำนวนสมาชิก $= (2-1+1)*(3-1+1)$
 $= 2*3$
 $= 6$

A[1:2,1:3]

	1	2	3
1			
2			

มีจำนวนสมาชิกเท่ากับ **6** ช่อง

การประกาศอาร์เรย์ 2 มิติในภาษาคอมไพเลอร์

- ภาษา C , C++ ประกาศโดยใช้รูปแบบดังนี้

รูปแบบ `data_type array_name[row_size][column_size]`

เช่น

```
int num[2][3];
```

```
char name[5][20];
```

การจัดเก็บอาร์เรย์สองมิติในหน่วยความจำ

ทำได้ 2 แบบ

1. การจัดเก็บด้วย **การเรียงแถว** เป็นหลัก (Row Major Order)
2. การจัดเก็บด้วย **การเรียงคอลัมน์** เป็นหลัก (Column Major Order)

กรณีใช้กับภาษาฟอร์แทรน เนื่องจากภาษาฟอร์แทรนเป็นภาษาที่ใช้กับคอมพิวเตอร์บางรุ่นในยุคแรกๆ และเครื่องเหล่านั้นมีโครงสร้างแอดเดรสที่เหมาะสมกับการจัดเก็บข้อมูลแบบเรียงคอลัมน์เป็นหลัก

สูตรการคำนวณหาตำแหน่งที่ใช้เก็บข้อมูลในอาร์เรย์สองมิติ

1. แบบการเรียงแถวเป็นหลัก

สูตร

$$\text{LOC}(K[i,j]) = B + w[C(i-L_1) + (j-L_2)]$$

โดยที่

$\text{LOC}(K[i,j])$ = ตำแหน่งแอดเดรสที่เก็บ $K[i,j]$ ในหน่วยความจำ

B = แอดเดรสเริ่มต้น (Base Address)

w = จำนวนช่องของหน่วยความจำที่จัดเก็บข้อมูลต่อหนึ่งสมาชิก

i = ตำแหน่งของแถวในอาร์เรย์

j = ตำแหน่งของคอลัมน์ในอาร์เรย์

L_1 = ค่าขอบเขตล่างสุด (lower Bound) ของคอลัมน์

L_2 = ค่าขอบเขตล่างสุด (lower Bound) ของคอลัมน์

C = จำนวนคอลัมน์ของแถวลำดับ

ตัวอย่าง

ต้องการทราบตำแหน่งแอดเดรสที่เก็บข้อมูลอาร์เรย์ **K** แถวที่ 2 คอลัมน์ 1 ($K[2,1]$) กำหนดให้ $B = 500$ $W = 4$ ไบต์

		Columns		
		0	1	2
Rows	0	$K[0,0]$	$K[0,1]$	$K[0,2]$
	1	$K[1,0]$	$K[1,1]$	$K[1,2]$
	2	$K[2,0]$	$K[2,1]$	$K[2,2]$
	3	$K[3,0]$	$K[3,1]$	$K[3,2]$

$K[0:3, 0:2]$

การคำนวณ

สูตร

$$\text{LOC}(K[i, j]) = B + w[C(i-L1) + (j-L2)]$$

$$\begin{aligned}\text{LOC}(K[2, 1]) &= 500 + 4[3(2-0) + (1-0)] \\ &= 500 + 4[6+1] \\ &= 500 + 28 \\ &= 528\end{aligned}$$

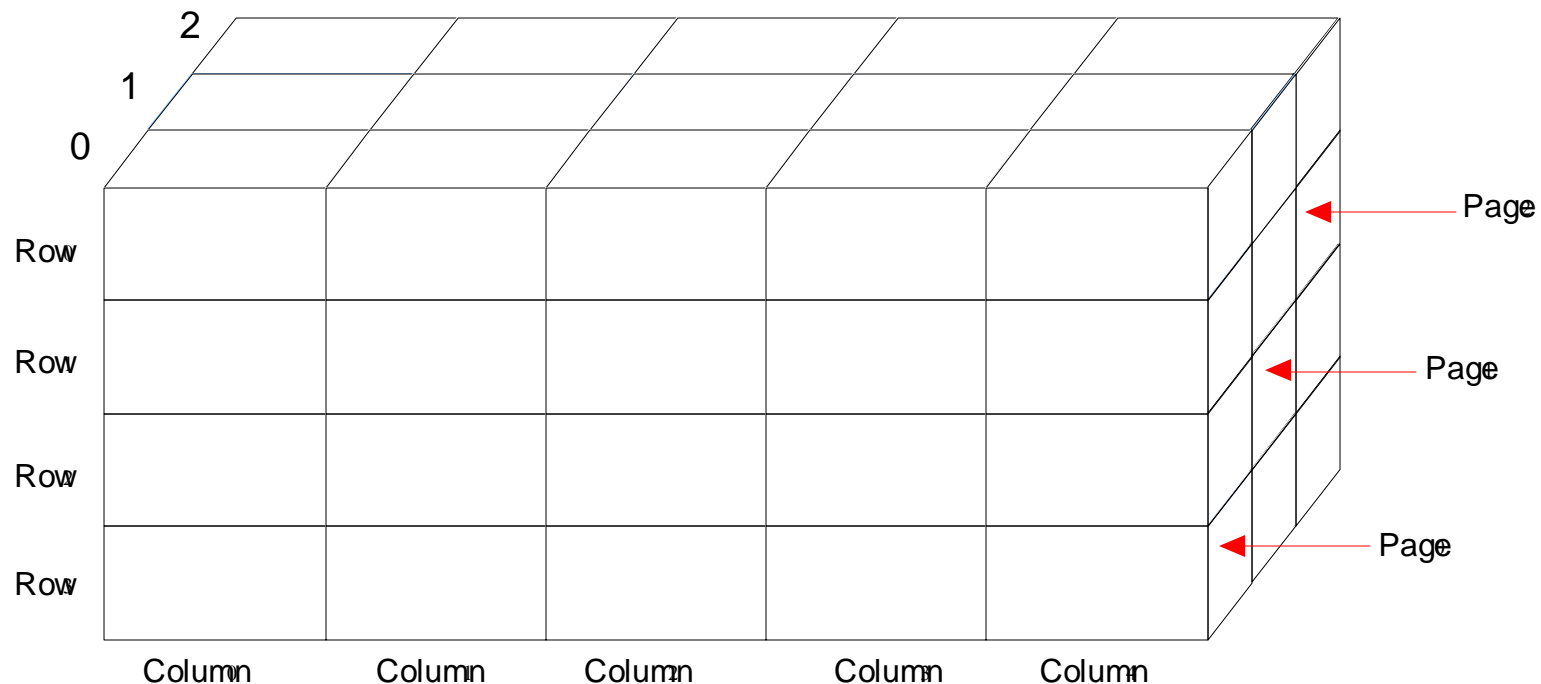
ดังนั้นอาร์เรย์ K แถวที่ 2 คอลัมน์ 1 จะจัดเก็บอยู่ในตำแหน่ง

แอดเดรสที่ 528

Base Add	Memory	Array	
500	7	K[0][0]	Row 0
504	8	K[0][1]	
508	1	K[0][2]	
512	2	K[1][0]	Row 1
516	-3	K[1][1]	
520	5	K[1][2]	
524	-2	K[2][0]	Row 2
528	4	K[2][1]	
532	9	K[2][2]	
536	0	K[3][0]	Row 3
540	6	K[3][1]	
544	3	K[3][2]	

อาร์เรย์สามมิติ (Three Dimension Array)

- อาร์เรย์สามมิติคือการนำเอาอาร์เรย์สองมิติมาเรียงซ้อนกันหลายๆชั้น (page) ดังนั้นจึงทำให้อาร์เรย์สามมิติ จะมีลักษณะเป็นแถวและคอลัมน์แล้วก็จะมีวล็กเพิ่มขึ้นมา



รูปแบบทั่วไปของโครงสร้างข้อมูลอาร์เรย์ 3 มิติ

ArrayName[L₁ : U₁ , L₂ : U₂ , L₃ : U₃]

เมื่อ ArrayName คือ ชื่อของโครงสร้างข้อมูลอาร์เรย์

L₁ คือ ค่าขอบเขตล่างสุด (Lower Bound) ของแถว

U₁ คือ ค่าขอบเขตสูงสุด (Upper Bound) ของแถว

L₂ คือ ค่าขอบเขตล่างสุด (Lower Bound) ของคอลัมน์

U₂ คือ ค่าขอบเขตสูงสุด (Upper Bound) ของคอลัมน์

L₃ คือ ค่าขอบเขตล่างสุด (Lower Bound) ของความลึก

U₃ คือ ค่าขอบเขตสูงสุด (Upper Bound) ของความลึก

การหาจำนวนสมาชิกของอาร์เรย์ 3 มิติ

- หาจากสูตร

$$\text{จำนวนสมาชิก} = (U_1 - L_1 + 1) * (U_2 - L_2 + 1) * (U_3 - L_3 + 1)$$

- เช่น จำนวนสมาชิกของอาร์เรย์ $A(2,3,4)$ หรือ $A(0:1,0:2,0:3)$

$$\text{จำนวนสมาชิก} = (1-0+1)*(2-0+1)*(3-0+1)$$

$$= 2*3*4$$

$$= 24$$

อกรเลขย 1 มิติ

- ใน Python จะมีรูปแบบดังนี้ [1,2,3,4,5,6]

```
a = [1,2,3,4,5,6]
```

```
print(a[2])
```

```
print(a[2:5])
```

- หรือเก็บเป็นตัวหนังสือ

```
cars = ["BMW", "Benz", "Lamborghini"]
```

```
print(cars[1])
```

```
print(cars)
```

```
print(cars[1:3])
```

จำนวนค่าใน Array

- การทราบ Length of an Array สามารถค้นหาได้จากคำสั่ง

len()

```
cars = ["BMW", "Benz", "Lamborghini"]
```

```
x = len(cars)
```

```
print(x)
```


การเปลี่ยนค่าใดค่าหนึ่งใน Array

- `a = [1,2,3,4,5,6]`
- `print(a[2])`
- `a[2]=5`
- `print(a[2])`

ความต้องการแสดงค่าใน Array ทีละตัว

- ใช้ Loop For

```
cars = ["Ford", "Volvo", "BMW"]
```

```
for x in cars:
```

```
    print(x)
```

การเพิ่มค่าใน Array

ในภาษา Python จะใช้ `append()`

```
cars = ["Ford", "Volvo", "BMW"]
```

```
cars.append("Honda")
```

```
print(cars)
```

การลบค่าใน Array

- ในภาษา Python จะใช้ `pop()`

```
cars = ["Ford", "Volvo", "BMW"]
```

```
cars.pop(1)
```

```
print(cars)
```

- หรือใช้ `remove()`

```
cars = ["Ford", "Volvo", "BMW"]
```

```
cars.remove("Volvo")
```

```
print(cars)
```

Python มีวิธีการจัดการกับค่าใน Array ได้หลายคำสั่ง

- `clear()` ลบค่าต่างๆ ใน Array

```
fruits = ['apple', 'banana', 'cherry', 'orange']
```

```
fruits.clear()
```

- `copy()` ทำการตัดลอกค่า

```
x = fruits.copy()
```

- `count()` นับจำนวนค่าที่กำหนด

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

```
print(a.count(2))
```

Python มีวิธีการจัดการกับค่าใน Array ได้หลายคำสั่ง (ต่อ)

- extend() นำ Array List อื่นมาต่อ Array ตัวเดิม เช่น

```
a=[1,2]
```

```
b=[3,4]
```

```
a.extend(b)
```

```
print(a)
```

- index() ค้นหาตำแหน่งใน Array

```
print(a.index(2))
```

Python มีวิธีการจัดการกับค่าใน Array ได้หลายคำสั่ง (ต่อ)

- insert() การเพิ่มค่าใน Array คล้ายๆ append แต่เพิ่มโดยระบุตำแหน่งได้ (append เพิ่มต่อท้าย)

```
fruits = ['apple', 'banana', 'cherry']
```

```
fruits.insert(1, "orange")
```

- reverse() การเรียง array จากท้ายมาหน้า จากหน้าไปท้าย

```
a=[1,2,3,4]
```

```
a.reverse()
```

```
print(a)
```

Python มีวิธีการกับค่าใน Array ได้หลายคำสั่ง (ต่อ)

- sort() เรียงลำดับใน Array

Syntax : `list.sort(reverse=True|False, key=myFunc)`

```
cars = ['Ford', 'BMW', 'Volvo']
```

```
cars.sort()
```

```
print(cars)
```

```
cars.sort(reverse=True)
```

```
print(cars)
```


LAB 1

```
def myFunc(e):  
    return len(e)  
  
cars = ['Ford', 'Mitsubishi', 'BMW', 'VW']  
  
cars.sort(key=myFunc)  
  
print(cars)
```

อาร์เรย์ 2 มิติ

อาร์เรย์ 2 มิติมีการจัดเรียงข้อมูลในแถว / คอลัมน์ สามารถเปรียบเทียบกับเมทริกซ์ในคณิตศาสตร์

ใน Python อาร์เรย์ 2 มิติ จะมีรูปแบบเป็น `[[1, 2], [3, 4]]`

```
b = [[1, 2], [3, 4]]
```

```
print(b[0][0])
```

หากต้องการแสดงทั้งแถวและคอลัมน์ใช้คำสั่ง

```
print(b[0:][0:])
```

หากต้องการเฉพาะแถวที่ 1 ใช้คำสั่ง

```
print(b[0][0:])
```

อาร์เรย์ 3 มิติ

มีลักษณะการใช้งานเหมือนกับที่ผ่านมา แต่มีอาร์เรย์เพิ่มอีกมิติ ตัวอย่างเช่น

```
[[[1, 2, 3], [4, 5, 6], [7, 8, 9]],
```

```
[[10, 11, 12], [13, 14, 15], [16, 17, 18]],
```

```
[[19, 20, 21], [22, 23, 24], [25, 26, 27]]]
```

```
d = [[[1, 2, 3], [4, 5, 6], [7, 8, 9]],[[10, 11, 12], [13, 14, 15], [16, 17, 18]],[[19,  
20, 21], [22, 23, 24], [25, 26, 27]]]
```

```
print(d[0][1])
```

```
print(d[0][1][2])
```

Python Lists

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
print(thislist[2:5])
```

Python Tuples

- A tuple is a collection which is ordered and **unchangeable**. In Python tuples are written with round brackets.

```
thistuple = ("apple", "banana", "cherry")
```

```
print(thistuple[1])
```

```
print(thistuple[-1])
```

จะ add เพิ่มไม่ได้

Python Sets

- A set is a collection which is **unordered and unindexed**. In Python, sets are written with curly brackets

- ค่าจะไม่ถูกเรียงจะ random เวลาสั่งใหม่แสดงค่า

```
thisset = {"apple", "banana", "cherry"}
```

```
print(thisset)
```

- Add an item to a set, using the add() method:

```
thisset.add("orange")
```

```
print(thisset)
```

Python Sets (๓๑)

- Add multiple items to a set, using the update() method:

```
thisset = {"apple", "banana", "cherry"}  
thisset.update(["orange", "mango", "grapes"])  
print(thisset)
```

- Remove "banana" by using the remove() method:

```
thisset.remove("banana")
```

- Remove "banana" by using the discard() method:

```
thisset.discard("banana")
```

Python Sets (๓๑)

- Remove the last item by using the pop() method:

```
thisset = {"apple", "banana", "cherry"}
```

```
x = thisset.pop()
```

```
print(x) #removed item
```

```
print(thisset) #the set after removal
```

- The clear() method empties the set:

```
thisset.clear()
```


Python Dictionaries

- A dictionary is a collection which is **unordered, changeable and indexed**. In Python dictionaries are written with curly brackets, and they have keys and values.

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
x = thisdict.get("model")  
print(x)
```

- C

thisdict["year"] = 2018

- Adding an item

thisdict["color"] = "red"

Python Dictionaries (ต่อ)

- The `pop()` method removes the item with the specified key name:

`thisdict.pop("model")` หรือ `del thisdict["model"]`