

คู่มือเรียน เขียนโปรแกรม

python

(ภาคปฏิบัติ)

มีตัวอย่าง
โปรเจกต์สำหรับ
การต่อยอด

```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodeName()
    label=symbol.sym_name.get_label(ast)
    print ' %s [%s]' % (label, '%s' % ast)
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '=' % ast[1]
        else:
            print ''
    else:
        print '['
        children = []
        for n, child in enumerate(ast[1:]):
            children.append(dotwrite(child))
        print ' %s -> {' % nodename,
        for name in children:
            print '%s' % name,
        print '];'
        children = []
        for n, child in enumerate(ast[1:]):
            children.append(dotwrite(child))
        print ' %s -> {' % nodename,
        for name in children:
            print '%s' % name,
        print '];'
        children = []
        for n, child in enumerate(ast[1:]):
            children.append(dotwrite(child))
        print ' %s -> {' % nodename,
        for name in children:
            print '%s' % name,
        print '];'
```

จุดเริ่มต้นสำหรับผู้สนใจการเขียนโปรแกรมด้วยภาษา Python ที่แม้แต่บริษัทยักษ์ใหญ่อย่าง Google ก็นำมาใช้งาน

- เรียนรู้ตั้งแต่คำสั่งพื้นฐานไปจนถึงการเขียนโปรแกรมเพื่อเชื่อมต่อกับระบบฐานข้อมูล
- สามารถนำไปใช้งานได้หลากหลาย แม้แต่ใช้เขียนโปรแกรมควบคุมหุ่นยนต์ก็ทำได้

มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

ห้องสมุดพระนครเหนือ



501031609

อธิบายโปรแกรมอย่างละเอียด เข้าใจทุกขั้นตอนในตัวอย่าง

ผู้ประพันธ์: หล่อเลิศสุนทร และ จิตะพันธ์ หล่อเลิศสุนทร

คำนำ

แม้ว่าการเลือกภาษาสำหรับการเขียนโปรแกรมสั่งงานคอมพิวเตอร์เป็นเรื่องไม่่ง่ายนัก เพราะมีภาษาอยู่มากมาย แต่ละภาษาก็มีจุดดี-จุดด้อยต่างกันออกไป เหมาะกับการใช้งานในลักษณะต่างกัน แต่สำหรับสภาพแวดล้อมทางเทคโนโลยีในปัจจุบัน ซึ่งมีแพลตฟอร์มเทคโนโลยีหลากหลาย การเลือกใช้ภาษา Python ที่ถูกพัฒนาขึ้นโดยไม่ยึดติดกับแพลตฟอร์ม คือสามารถรันโปรแกรมที่เขียนขึ้นได้ทั้งบนคอมพิวเตอร์ทั่วไป และเครื่องเซิร์ฟเวอร์ที่รันระบบปฏิบัติการ Unix, Linux, Windows (ทั้งรุ่นเก่า เช่น Windows Me ไปจนถึง Windows 7), FreeBSD, Mac OS X หรือแม้แต่เครื่อง Palm และโทรศัพท์มือถือที่ใช้ระบบปฏิบัติการข้างต้น หรือระบบปฏิบัติการ Symbian ของ Nokia ก็ตาม ถือเป็นทางเลือกหนึ่งที่น่าสนใจ ทำให้เขียนโปรแกรมครั้งเดียว แต่สามารถนำไปใช้บนระบบปฏิบัติการต่างๆ ได้ไม่ต้องเสียเวลาในการเขียนโปรแกรมสำหรับแต่ละระบบปฏิบัติการ อีกทั้งยังเป็นซอฟต์แวร์แบบ Open Source ที่แจกจ่ายให้ใช้งานฟรีอีกด้วย

ภาษา Python ง่ายต่อการเรียนรู้ มีโครงสร้างของภาษาไม่ซับซ้อน เข้าใจง่าย รูปแบบของคำสั่งง่ายเหมือนภาษา BASIC การกำหนดชนิดของตัวแปรและใช้งานทำได้ด้วยวิธีง่ายๆ การกำหนดเงื่อนไข หรือสั่งให้วนรอบทำงานซ้ำๆ ก็มีคำสั่งไม่มากจนเกินไป ตัวภาษามีความยืดหยุ่นสูง และมีคำสั่งในการจัดการกับข้อความ และไฟล์ข้อความ (Text File) ได้เป็นอย่างดี อีกทั้งมีการรวมเอาส่วนดีของภาษาต่างๆ เข้ามาไว้ด้วยกันในตัว

ด้วยข้อดีหลายๆ ประการของภาษา Python ทำให้มีผู้นิยมใช้มากขึ้นเรื่อยๆ แม้บริษัทยักษ์ใหญ่ในโลกอินเทอร์เน็ตอย่าง Google ก็นำมาใช้งาน หรือแม้แต่การเขียนโปรแกรมควบคุมหุ่นยนต์ก็สามารถใช้ Python เขียนได้ เหล่านี้เป็นเครื่องรับประกันได้ถึงความสามารถของภาษา Python ได้เป็นอย่างดี หากสนใจจะดูตัวอย่างเพิ่มเติมว่า มีคนเอาภาษา Python ไปพัฒนาอะไรบาง ลองดูได้ที่ <http://www.python.org/about/success> ครับ

ความน่าสนใจของภาษา Python จึงเป็นแรงบันดาลใจให้หนังสือเล่มนี้ถูกเขียนขึ้น เพื่อเป็นจุดเริ่มต้นให้ผู้สนใจการเขียนโปรแกรมด้วยภาษา Python ได้ศึกษา ตั้งแต่คำสั่งพื้นฐาน ไปจนถึงการเขียนโปรแกรมเพื่อเชื่อมต่อกับระบบฐานข้อมูล โดยขอแนะนำว่า การเรียนเขียนโปรแกรมเป็นการเรียนที่ต้องอิงการปฏิบัติ ผู้อ่านควรทดลองพิมพ์คำสั่งเพื่อให้เห็นการทำงานและสร้างความเข้าใจไปพร้อมกัน ซึ่งการอธิบายโปรแกรมที่ใช้ในหนังสือเล่มนี้ ได้เน้นอธิบายการทำงานเป็นบรรทัดๆ และเห็นถึงผลลัพธ์ที่ได้จากการทำงาน การพิมพ์โปรแกรมพร้อมกับการอ่านคำอธิบายไปด้วยกันจะทำให้เรียนรู้ได้เร็วขึ้น

โซทิพันธุ์ หล่อเลิศสุนทร
จิวะพันธุ์ หล่อเลิศสุนทร

สารบัญ

บทที่ 1	ประโยชน์จากการใช้งาน Python	9
	ขั้นตอนการดาวน์โหลดและนำมาติดตั้งสำหรับ Microsoft Windows และ Linux	11
	ขั้นตอนการทดสอบหลังทำการติดตั้งสำเร็จ	14
บทที่ 2	โครงสร้างและลักษณะการเขียนโปรแกรม	15
	คำสั่ง import	19
	คำสั่ง from	20
	รูปแบบการเขียนด้วยวิธีที่ 1	21
	รูปแบบการเขียนด้วยวิธีที่ 2	21
	รูปแบบการเขียนด้วยวิธีที่ 3 : แบบ Object Oriented	22
	รูปแบบการเขียนด้วยวิธีที่ 4 : แบบ Object Oriented พร้อมกลุ่มคำสั่งเรียกใช้งาน Object	23
	รูปแบบการเขียนด้วยวิธีที่ 5 : แบบ Object Oriented ที่เกิดจากวิธีที่ 3 และ 4	
	มาประยุกต์ใช้งานร่วมกัน	24
	การกำหนดคำสั่งกลุ่มย่อยให้กับคำสั่งหลัก (Indenting)	26
	การทำความเข้าใจการใช้เครื่องหมาย ' หรือ " สำหรับข้อความหรือตัวแปรชนิดข้อความ	27
	การใช้เครื่องหมาย (), [], { } ร่วมกับตัวแปรหรือฟังก์ชัน	28
	ประเภทการใช้เครื่องหมายร่วมกับตัวแปร	28
	ประเภทการใช้เครื่องหมายร่วมกับไพเรซีเจอร์หรือฟังก์ชัน	29
	ประเภทการใช้เครื่องหมายร่วมกับคำสั่งหรือฟังก์ชันภายใน	30
	ประเภทการใช้เครื่องหมายที่ทำหน้าที่เสมือนเป็นฟังก์ชันภายในแบบอัตโนมัติ	30
	การใช้เครื่องหมายคอมเมนต์ (Comment)	31
	Python กับการใช้ตัวอักษรตัวเล็กหรือใหญ่ (Case Sensitive)	31
	มาทดลองเขียนโปรแกรมด้วยภาษา Python	31
	ทดลองเขียนคำสั่งด้วย Python Shell	31
	ทดลองเขียนคำสั่งด้วยรูปแบบไฟล์โปรแกรม	37
บทที่ 3	รู้จักกับการใช้ตัวแปรของ Python	39
	การเลือกใช้งานตัวแปรสำหรับการเขียนโปรแกรม	40
	รูปแบบการสร้างและใช้งานตัวแปร	40
	ชนิดของตัวแปรที่มีใช้งานใน Python	41
	ทำความเข้าใจกับชนิดตัวแปรพื้นฐานของ Python	42
	ตัวแปรชนิดตัวเลขจำนวนเต็ม (Integer)	43
	ตัวแปรชนิดบูลีน (Boolean)	45

ตัวแปรชนิดตัวเลขที่มีทศนิยม	46
ตัวแปรชนิดตัวเลขจำนวนเชิงซ้อน (Complex Number)	48
ตัวแปรชนิดตัวเลขทศนิยม Decimal	48
ตัวแปรชนิดข้อความ (String)	49
เทคนิคการ Slicing และ Striding String สำหรับตัวแปรชนิดข้อความ	66
การรีเซตค่าในตัวแปร	73
การใช้ตัวแปรประเภท Sequence	73
เทคนิคการใช้งานตัวแปรประเภท Sequence ในลักษณะ Nested	76
รูปแบบการเขียนตัวแปรประเภท Sequence	78
ตัวแปรชนิด Tuple	80
ตัวแปรชนิด List	81
ตัวแปรชนิด Dictionary	84
การใช้โอเปอเรเตอร์ (Operator) กับตัวแปร	89
การใช้โอเปอเรเตอร์กับตัวแปรชนิดตัวเลข (Numeric And Iterable Operator)	90
การใช้โอเปอเรเตอร์กับตัวเลขจำนวนเต็มในระดับบิต (Integer Bitwise Operator)	94
เซตของข้อมูลกับการใช้โอเปอเรเตอร์ (Set Operator)	96
การกำหนดขอบเขตการใช้งานตัวแปรในโมดูล	101

บทที่ 4 การตรวจสอบเงื่อนไข **109**

ผลลัพธ์เป็นไปตามเงื่อนไขหรือค่าที่เป็นจริง (True)	110
ผลลัพธ์ไม่ตรงกับเงื่อนไขที่ตรวจสอบหรือได้ผลเป็นเท็จ (False)	110
การใช้คำสั่ง if ตรวจสอบเงื่อนไขและทำงานกลุ่มคำสั่งย่อย เมื่อการตรวจสอบเงื่อนไขเป็นจริง (True) เท่านั้น	119
การใช้คำสั่ง if ร่วมกับ elif สำหรับตรวจสอบเงื่อนไขและทำงานกลุ่มคำสั่งย่อย เมื่อผลลัพธ์การตรวจสอบเป็นจริง (True) เท่านั้น	120
การใช้คำสั่ง if ที่มีลักษณะการตรวจสอบเงื่อนไขแบบซ้อนกัน (Nested)	123
การใช้คำสั่ง if ร่วมกับคำสั่ง else	127
การกำหนดเงื่อนไขมากกว่าหนึ่งเงื่อนไขให้กับคำสั่ง if	130
รูปแบบการเขียนคำสั่ง if ร่วมกับ and operator	130
รูปแบบการเขียนคำสั่ง if ร่วมกับ or operator	131
รูปแบบการเขียนคำสั่ง if ร่วมกับ and และ or operator	132

บทที่ 5 คำสั่งการวนทำซ้ำ **137**

ตัวอย่างที่ 1 คำสั่งรับข้อมูลแบบลูกทุ่งลูกทุ่ง	137
ตัวอย่างที่ 2 คำสั่งรับข้อมูลโดยใช้คำสั่งการวนทำซ้ำ for	139
ตัวอย่างที่ 3 การใช้คำสั่ง while สำหรับกลุ่มคำสั่งย่อยในการวนทำซ้ำ และสิ้นสุดด้วยการตรวจสอบเงื่อนไข	140

ตัวอย่างที่ 4 การใช้คำสั่ง while ที่มีมากกว่า 1 เงื่อนไขสำหรับการวนซ้ำ	141
ตัวอย่างที่ 5 การใช้คำสั่ง while กับการกำหนดเงื่อนไขในลักษณะ Infinity Repetition	142
ตัวอย่างที่ 6 การใช้คำสั่ง for ร่วมกับ break และ continue	147
ตัวอย่างที่ 7 การใช้คำสั่ง while ร่วมกับ break และ continue	148

บทที่ 6 การสร้างไพโรซิเจอร์หรือฟังก์ชันของ Python **151**

รูปแบบการสร้างฟังก์ชันของภาษา Python และวิธีเขียนโปรแกรมเพื่อเรียกใช้งาน	153
ฟังก์ชันที่ใช้งานแบบ Global	155
ฟังก์ชันที่ใช้งานแบบ Local	157
Internal Method หรือ Private Method	160
Interface Method	161
การกำหนดขอบเขตของตัวแปรภายในของฟังก์ชัน	164
ลักษณะการใช้คีย์เวิร์ดอาร์กิวเมนต์และพารามิเตอร์ร่วมกับฟังก์ชัน (Argument and Parameter Function Keyword)	168
อาร์กิวเมนต์แบบตำแหน่งและจำนวนใช้งานที่แน่นอน (Positional Argument)	169
อาร์กิวเมนต์แบบขยายจำนวนตามค่าข้อมูล (Sequence Unpacking Operator)	172
ความแตกต่างระหว่างฟังก์ชันที่ใช้และไม่ใช้คำสั่ง return	176

บทที่ 7 การจัดการกับปัญหาเมื่อเกิดข้อผิดพลาดระหว่างการพัฒนาโปรแกรม (Error And Exceptions) **179**

ข้อผิดพลาดที่เกิดจากการใช้คำสั่งผิดรูปแบบที่กำหนดไว้ (Syntax Error)	179
ข้อผิดพลาดที่เกิดระหว่างการทำงานของคำสั่ง (Runtime Error)	180
ข้อผิดพลาดที่เกิดจากการประมวลผลของกลุ่มคำสั่ง แต่ได้ผลลัพธ์ที่ไม่ตรงตาม ความต้องการ (Logic Error)	181
คำสั่ง try และ except กับการเกิด Error	181
การใช้คำสั่ง else ร่วมกับคำสั่ง try และ except	192
การใช้คำสั่ง finally ร่วมกับคำสั่ง try และ except	195
สร้างให้เกิด Error ด้วยคำสั่ง raise exception	199

บทที่ 8 Python กับการเชื่อมต่อฐานข้อมูล **201**

ประโยชน์ที่ได้จากการใช้ฐานข้อมูล	202
จะใช้ฐานข้อมูลต้องรู้จักอะไรบ้าง	205
แนวทางการต่อยอดทางข้อมูลในเชิงลึกกับฐานข้อมูล	210
มาเลือกใช้ฐานข้อมูลให้เหมาะกับงานที่กำลังพัฒนาโปรแกรม	212
กลุ่มฐานข้อมูลประเภท Flat File Database	213
การเลือกใช้ Database-Manager (DBM) ร่วมกับไฟล์ข้อมูล	219
กลุ่มฐานข้อมูลประเภท File Server-Base Database	224

การอ้างอิงโมดูลของ SQLite มาใช้งานในโมดูล Python	226
วิธีการสร้างไฟล์ฐานข้อมูลด้วยกลุ่มคำสั่ง DB-API ของ Python	226
การใช้ Cursor Object กับการอ่านค่าข้อมูลด้วยคำสั่ง Select	228
การใช้ Cursor Object กับการเพิ่มข้อมูลใหม่ในตารางด้วยคำสั่ง Insert	232
การใช้ Cursor Object กับการแก้ไขข้อมูลที่มีอยู่แล้วในตารางด้วยคำสั่ง Update	234
การใช้ Cursor Object กับการลบเรคอร์ดข้อมูลจากตารางด้วยคำสั่ง Delete	237
การสร้างตาราง (Table) และความสัมพันธ์ของตาราง (Relationship)	238
สร้างตารางด้วยกลุ่มคำสั่ง DDL (Data Definition Language)	239
สร้างตารางด้วยโปรแกรมเครื่องมือสำหรับไฟล์ฐานข้อมูลประเภท SQLite	246
การเพิ่มเรคอร์ดลงในตารางด้วยคำสั่ง insert ของ SQL Statement	247
การอ่านค่าข้อมูลจากตารางโดยใช้คำสั่ง Select	254
การแก้ไขค่าข้อมูลเดิมในตารางด้วยคำสั่ง Update	259
การลบค่าข้อมูลในตารางข้อมูลด้วยคำสั่ง Delete	262
การปิดการเชื่อมต่อฐานข้อมูล	267
กลุ่มฐานข้อมูลประเภท Database Server	267
การดาวน์โหลด MySQL Server พร้อมชุด Workbench ของ MySQL	
และโมดูลพิเศษสำหรับเชื่อมต่อกับ Python	269
การติดตั้ง MySQL Server Database และชุดบริหารการจัดการฐานข้อมูล (MySQL Workbench)	269
การติดตั้งโมดูลพิเศษสำหรับการเชื่อมโยงฐานข้อมูลกับ Python	274
การสร้างฐานข้อมูลและตารางสำหรับการทดสอบโปรแกรมด้วย MySQL Workbench	275
การพัฒนาโปรแกรมสำหรับเชื่อมต่อฐานข้อมูล MySQL ด้วย DB-API ของ Python	293
ปัญหาและวิธีการแก้ไขในระหว่างการพัฒนาโปรแกรม	325
บทที่ 9 พื้นฐานการใช้โมดูล Tkinter สำหรับ Graphical User Interface (GUI)	329
ทดสอบสร้างและปรับแต่งค่าใช้งานของจอภาพประเภท GUI ด้วยโมดูล Tkinter	335
ตัวอย่างที่ 1 แสดงการสร้างจอภาพจากโมดูล Tkinter	335
ตัวอย่างที่ 2 แสดงการสร้างจอภาพและปรับแต่งค่า Attribute ที่ชื่อ title และ geometry	335
ตัวอย่างที่ 3 แสดงการสร้างจอภาพและอ่านค่าความกว้างและสูงจาก method เพื่อปรับ	
การแสดงผลให้ตรงกับหน้าจอที่ใช้งานในปัจจุบัน	336
ตัวอย่างที่ 4 แสดงการสร้างจอภาพพร้อมการเรียกฟังก์ชันใช้งานด้วยการ bind	337
ทดสอบสร้างและปรับแต่งค่าใช้งานของกลุ่ม widget ภายใต้ GUI ของโมดูล Tkinter	337
ตัวอย่างที่ 1 แสดงการสร้าง Label widget บนจอภาพ	341
ตัวอย่างที่ 2 แสดงการสร้าง Label widget พร้อมการปรับแต่ง Option	342
ตัวอย่างที่ 3 แสดงการสร้าง Label widget ที่ใช้ร่วมกับรูปภาพจากไฟล์เพื่อแสดงบนจอภาพ	343

บทที่ 1

ประโยชน์จากการใช้งาน Python

การเลือกภาษาสำหรับการเขียนโปรแกรม แคนึกชื่อไม่ว่าใครก็ตามที่เป็นมือใหม่หรือโปรแกรมเมอร์ที่กำลังมองหาภาษาใหม่ๆ มาฝึกหัดซ้อมมือเพื่อนำไปใช้งาน หลายคนก็มักจะมีแต่เครื่องหมายต่างๆ ในสมอง เช่น เครื่องหมายคำถาม สำหรับการตัดสินใจเลือกภาษาเพื่อนำมาศึกษาและใช้งาน เมื่อเลือกภาษามาแล้ว ก็ไม่รู้ว่าจะอนาคตจะใช้งานได้จริงมั๊ย และอายุของภาษาจะได้รับความนิยม หรือจะยุติเมื่อใด ต่อมาความยากง่ายในการเรียนรู้สอดคล้องกับระยะเวลาที่ต้องอ่าน ต้องฝึกความสามารถของการเขียนโปรแกรมโดยทั่วไป หรือต้องใช้เทคนิคการเขียนโปรแกรมที่มีความซับซ้อนด้วยเงื่อนไขของลักษณะงานที่ถูกกำหนดขึ้น ทั้งหมดนี้ผู้พัฒนาโปรแกรมจะไปค้นคว้าหาคำตอบ เพื่อนำมาสร้างหรือใช้ในการเขียนโปรแกรมจากที่ไหน ผู้เขียนหมายถึงตัวภาษาที่เราเลือกเป็นที่นิยมแพร่หลายมากมั๊ย แหล่งค้นคว้าจากตำรับตำรา หรือทางด้านโลกอินเทอร์เน็ตมีให้ค้นคว้า ถาม-ตอบ แอบบอกเทคนิคต่างๆ หรือแหล่งดาวน์โหลดตัวอย่าง Source Code ซึ่งในตัวอย่างจะแนะแนวการเขียน และแสดงผลลัพธ์ที่เกิดจากการทำงานของโปรแกรม ทำให้ผู้เรียนเกิดความเข้าใจ และพิจารณาการนำตัวอย่างโปรแกรมไปประยุกต์ให้ตรงกับงานที่ผู้เรียนกำลังต้องการนำไปใช้งาน ต่อมาก็เครื่องหมายตกใจ ! ที่เกิดขึ้นในสมอง จะเริ่มที่ภาษาไหน เริ่มต้นอย่างไร อ่านและฝึกเท่าไรจึงจะเขียนเป็น จะต้องอ่านหนังสือเล่มไหน อ่านอีกกี่เล่ม !!! คิดแล้วคงเหนื่อยใจพอสมควร ทั้งนี้ยังไม่รวมการเขียนโปรแกรมตามเทคโนโลยีชนิดต่างๆ เช่น การสร้างโปรแกรมให้ทำงานแบบแอปพลิเคชัน สำหรับติดตั้งใช้งานบนเครื่องคอมพิวเตอร์นั้นๆ หรือสร้างเป็นแบบเว็บแอปพลิเคชัน ที่เหมาะต่อการติดตามหรือใช้งานผ่านอินเทอร์เน็ต ซึ่งสามารถใช้งานที่ไหนก็ได้ ขอให้มีอินเทอร์เน็ตเข้าถึงก็พอใช้ได้ หรือจะฝังเข้าไปกับเครื่องโทรศัพท์ประเภทโมบายโฟน ที่มีหลากหลายยี่ห้อแต่ละยี่ห้อ ก็มีระบบปฏิบัติการไม่

ซ้ำกันอีก ที่เล่าสู่กันฟัง ก็น่าสนใจสำหรับการเริ่มศึกษาการเขียนโปรแกรมด้วยภาษาต่างๆ เพื่อจะได้นำไปพัฒนาการสร้างงานตามความต้องการและเหมาะสมต่อการใช้งาน และสอดคล้องกับเทคโนโลยีที่นำมาใช้ในปัจจุบัน

ด้วยปัจจุบันเรามีระบบปฏิบัติการที่หลากหลายสำหรับใช้งานร่วมกับเทคโนโลยีของฮาร์ดแวร์ เช่น ระบบปฏิบัติการ Mac OS ของ Macintosh ระบบปฏิบัติการ Windows ระบบปฏิบัติการ Linux บนเครื่องคอมพิวเตอร์ทั่วไป หรือเครื่องคอมพิวเตอร์ประเภทเซิร์ฟเวอร์หลัก แม้กระทั่งเครื่องโทรศัพท์เคลื่อนที่บางยี่ห้อ ที่ทำงานภายใต้ระบบปฏิบัติการดังกล่าวก็มีมากขึ้นตามลำดับ หรือแม้แต่ระบบปฏิบัติการโดยเฉพาะโทรศัพท์มือถือของ Nokia ที่ใช้ระบบปฏิบัติการ Symbian หรือระบบปฏิบัติการ Android ที่เริ่มใช้ในมือถือซัมซุง เป็นต้น

จะเห็นได้ว่าเทคโนโลยีของระบบปฏิบัติการได้รับการปรับปรุงและพัฒนาให้มีความเหมาะสมกับเทคโนโลยีของฮาร์ดแวร์ในปัจจุบัน หากเราต้องการพัฒนาโปรแกรมสำหรับใช้ร่วมกับระบบปฏิบัติการเหล่านี้ ต้องใช้ความรู้ในการเขียนโปรแกรมอย่างน้อยก็ 2 ถึง 3 ภาษา เป็นอย่างต่ำแน่ๆ Python นับเป็นภาษาหนึ่งที่น่าสนใจ ต่อการนำมาศึกษาการเขียนโปรแกรม เพื่อพัฒนาตามต้องการภายใต้ระบบปฏิบัติการที่หลากหลาย ด้วยที่ภาษา Python เป็น Open Source แบบฟรีไลเซนส์ และด้วยโครงสร้างของภาษา Python ยังเปิดโอกาสให้ผู้ใช้งานพัฒนาไลบรารีใหม่เพิ่มเติม ด้วยภาษาของ Python, C++, Java, ตระกูล Dot Net และอีกหลายๆ ภาษา

ก่อนอื่นขอแนะนำผู้สร้างภาษา Python ให้กับท่านผู้อ่านได้รู้จักชื่อเสียงเรียงนาม เขาชื่อ Guido Van Rossum บุคคลผู้เฝ้ามองงานทางเขียนโปรแกรมให้กับพวกเรา ปีที่ Python เปิดตัวคือ พ.ศ. 2534 (1991) เราควรยกย่องบุคคลเหล่านี้ครับ เขาสร้างเพื่อที่จะนำมาใช้งาน แล้วยังเปิดโอกาสให้ผู้อื่นได้ใช้ภาษา Python แบบฟรีไลเซนส์ (Free License) อีกทั้งได้ทุนสนับสนุนการเงินจากบริษัทยักษ์ใหญ่ เช่น Google, eBay เป็นต้น ให้กับสมาคมกลุ่ม Python เพื่อพัฒนาภาษา Python ในเวอร์ชันใหม่ ปัจจุบัน Guido Van Rossum ได้เข้าร่วมทำงานให้กับบริษัท Google โดยทาง Google ก็ได้ประกาศสนับสนุน Python ให้เป็นภาษาสำหรับการพัฒนาร่วมกับระบบปฏิบัติการ Android ของ Google นั้น แสดงให้เห็นถึงการตอบสนองของตลาด Python ที่มีโอกาสเติบโตในตลาดอุตสาหกรรม เช่น อุตสาหกรรมบันเทิง โดยเฉพาะซอฟต์แวร์เกี่ยวกับคอมพิวเตอร์กราฟิก, เสียง, Visual Effect สำหรับภาพยนตร์ก็ได้นำภาษา Python มาช่วยในการพัฒนาซอฟต์แวร์ประเภทนี้ร่วมกับภาษา C หรือ C++, อุตสาหกรรมเกี่ยวกับ Enterprise Resource Planning (ERP), อุตสาหกรรมเกี่ยวกับ Customer Relationship Management (CRM) อุตสาหกรรมเกมส์, อุตสาหกรรมมือถือ เช่น โนเกีย ก็สามารถนำ Python พัฒนาโปรแกรมอุตสาหกรรมการบิน เช่น ซอฟต์แวร์เกี่ยวกับ Plainview, อุตสาหกรรมซอฟต์แวร์ประเภท Content Management System เป็นต้น

ลักษณะโดยรวมของภาษา Python ถูกสร้างขึ้นมาเพื่อให้เรียนรู้ได้ไม่ยาก รูปแบบของคำสั่งก็คล้ายๆ

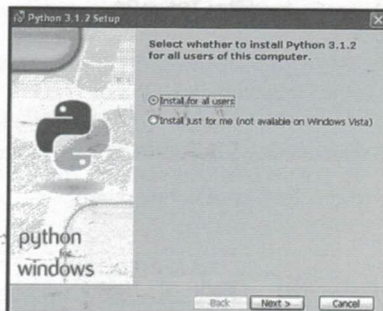
กับภาษาเบสิก ชนิดของตัวแปรก็ทำการกำหนดและสร้างมาใช้งานได้ด้วยวิธีง่ายๆ การกำหนด เงื่อนไข หรือการวนลูปก็มีคำสั่งไม่มากจนเกินไป แต่สามารถเขียนใช้งานได้เท่ากับภาษาอื่นๆ โดยพื้นฐานการเขียน โปรแกรม Python จะทำงานด้วยตัวแปลภาษาแบบอินเทอร์พรีเตอร์ (Interpreter) ผลลัพธ์จากการ ทดสอบคำสั่งโดยพื้นฐานจะเป็นลักษณะโหมดข้อความ (Text Mode) แต่ถ้าหากต้องการใช้เป็น Graphic User Interface (GUI) หรือแบบมีจอภาพสวย มีช่องสำหรับกรอกข้อมูล, มีปุ่มให้กดทำการจัดเก็บข้อมูล พิมพ์ออกรายงาน ก็สามารถใช้คำสั่ง หรือเครื่องมือเสริมของ Python มาทำการจัดสร้างได้เช่นกัน ข้อเด่น ของภาษา Python หรือกลุ่มคำสั่งที่ Python สร้างเสร็จไว้ใช้งานแล้ว เราสามารถนำไปใช้ งานต่างระบบ ปฏิบัติการได้ด้วย

ภาษา Python สามารถที่จะเขียนให้เป็นโปรแกรมเชิงวัตถุ (Object Oriented Programming) ได้ หรือเปล่า? คำถามนี้น่าสนใจต่อผู้ที่กำลังศึกษาภาษา Python ถ้าภาษานี้สามารถรองรับการเขียนโปรแกรม แบบ OOP ได้ ก็น่าจะเป็นแรงจูงใจให้กับผู้ที่สนใจที่จะนำ Python ไปพัฒนางานต่างๆ ตามต้องการ คำ ตอบก็คือ ภาษา Python รองรับการพัฒนาโปรแกรมแบบ OOP หรือเขียนโปรแกรมแบบโครงสร้างปกติ ทั่วไปก็ได้ ส่วนเทคนิคการเขียนโปรแกรม OOP จะครอบคลุมตามหลักการของ OOP ชนิด 100% หรือ เปล่า? ก็ให้ตรวจสอบขีดความสามารถในการเขียนของ Python แต่ละเวอร์ชัน

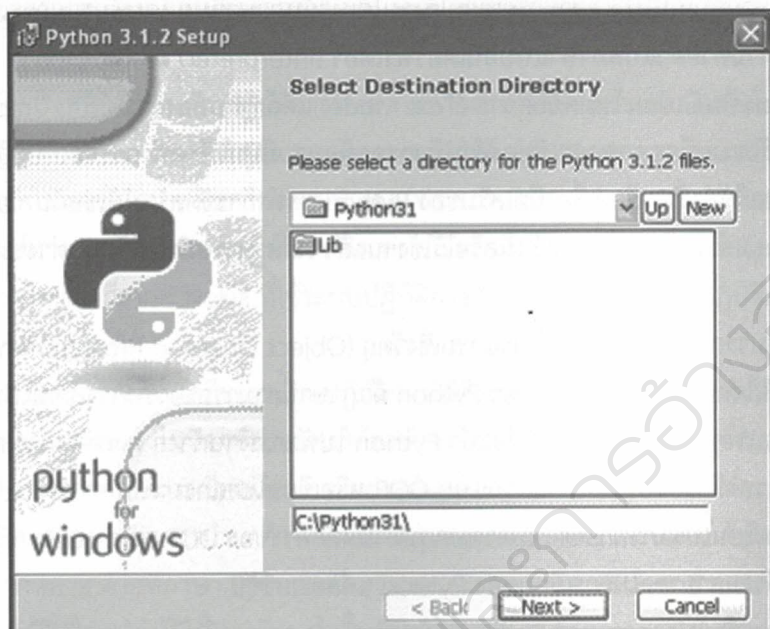
ก้าวแรกก่อนที่เราจะเป็นโปรแกรมเมอร์ Python ก็ต้องติดตั้ง Python เพื่อที่จะนำมาหัดเขียนให้ สำเร็จ ภาษา Python ที่จะนำมาติดตั้ง ให้ดาวน์โหลดที่เว็บไซต์ <http://www.python.org> พอเข้าไปใน เมนูดาวน์โหลดแล้ว ก็เลือกให้ตรงกับระบบปฏิบัติการที่เราจะนำไปใช้งาน เช่น Python สำหรับ Microsoft Windows, Python สำหรับ Linux หรือสำหรับเครื่อง Mac OS เป็นต้น อีกเรื่องที่ต้องดูให้ดี คือ Python สำหรับ 32 บิต หรือ 64 บิต ให้ตรวจสอบจาก CPU ของเครื่องเราว่าเป็นชนิดไหน ในขณะที่ผู้เขียนกำลัง เขียนหนังสือเล่มนี้ Python ที่เปิดให้ดาวน์โหลดเป็นเวอร์ชัน 3.1.2 ถ้าหากใครใช้ Python ที่ต่ำกว่าเวอร์ชัน นี้ ก็จะมีบางคำสั่งที่อาจแตกต่างกันบ้าง แต่โดยรวมก็ยังใช้งานด้วยกันได้

หลังจากดาวน์โหลดไฟล์มาจาก www.python.org แล้ว ให้รันไฟล์สำหรับติดตั้ง ซึ่งจะมีขั้นตอนติด ตั้งง่ายๆ ดังนี้

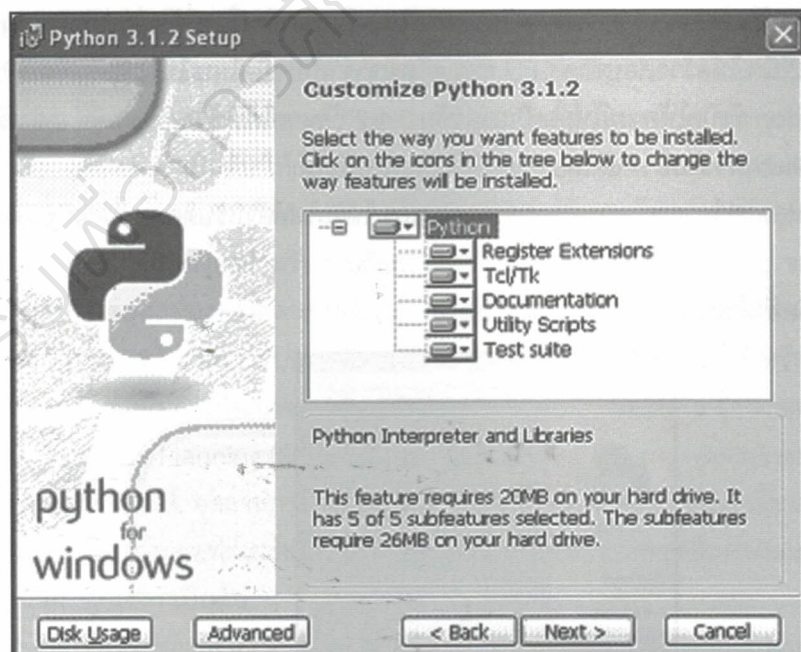
1. เลือกว่าจะติดตั้งสำหรับใช้งานได้ทุกคนหรือไม่



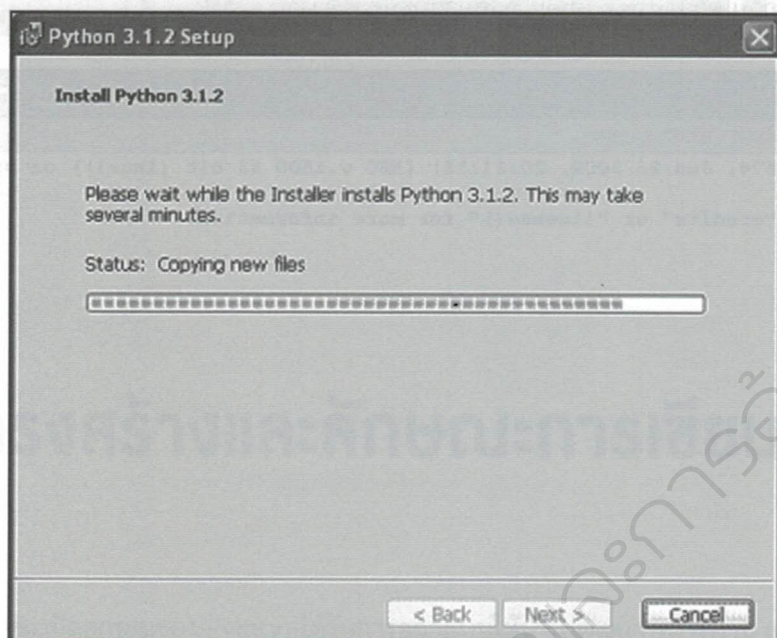
2. เลือกไดเรกทอรีที่จะติดตั้งโปรแกรม



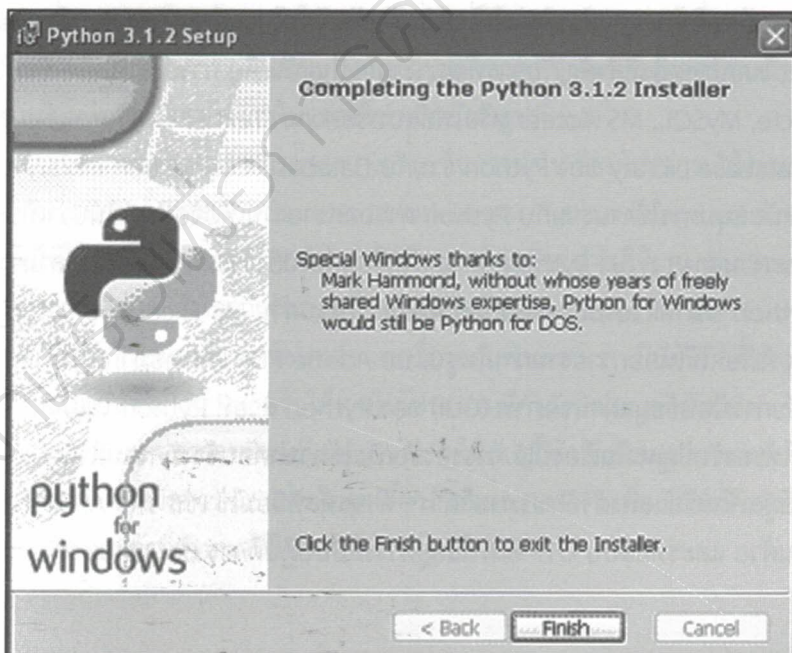
3. เลือกติดตั้งองค์ประกอบต่างๆ



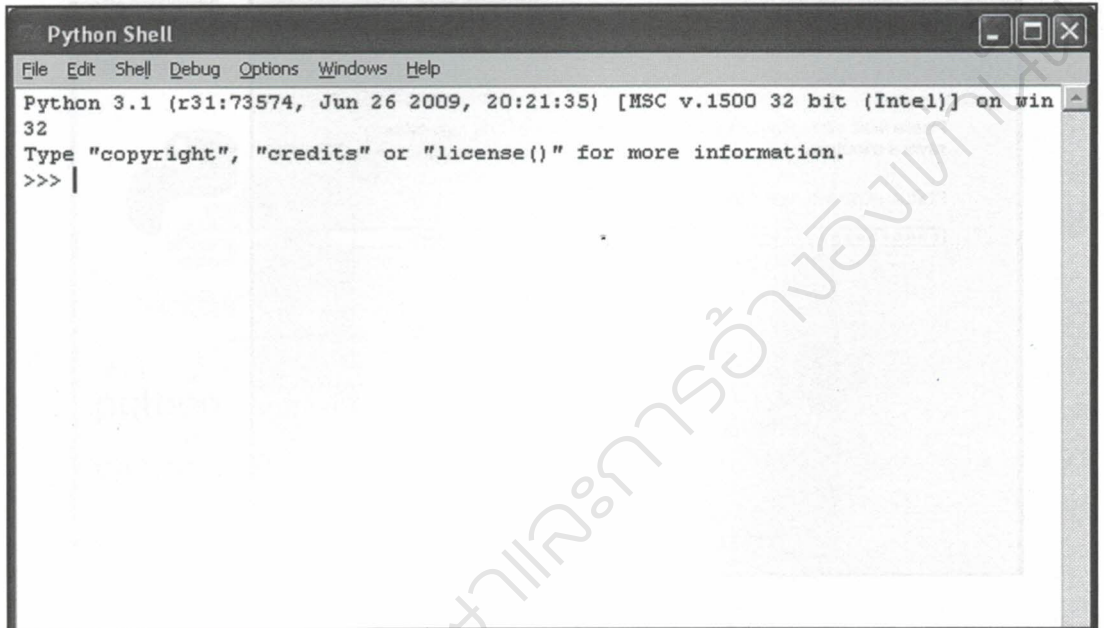
4. โปรแกรมกำลังติดตั้งและแสดงสถานะให้ทราบ



5. หน้าจอแจ้งว่าทำการติดตั้งเสร็จแล้ว



หลังจากติดตั้งเสร็จ ให้ทดลองเรียกใช้ IDLE (Python GUI) จากเมนู Start > All Programs > Python 3.1 ซึ่งจะปรากฏวินโดว์ Python Shell พร้อมสำหรับการทำงาน ดังรูป



เก็บตกความสามารถของภาษา Python ก่อนที่จะเริ่มต้นหัดเขียนโปรแกรม Python เพื่อจะนำไปพัฒนางานกันจริงๆ จังๆ ในอนาคตสิ่งที่สำคัญกับการพัฒนางานในปัจจุบัน คือ การนำข้อมูลไปเชื่อมโยงกับฐานข้อมูล เช่น Oracle, MySQL, MS Access หรือไฟล์แบบข้อความ (Text File) หรือฐานข้อมูลยี่ห้ออื่นๆ ด้วย ผ่าน SQL Database Library ของ Python ร่วมกับ Database Connection Library ของฐานข้อมูลแต่ละยี่ห้อที่สนับสนุนการใช้งานร่วมกับ Python การออกรายงานเพื่อจัดพิมพ์ก็ถือว่าเป็นเรื่องหนึ่งที่สำคัญในการพัฒนางานต่างๆ ภาษา Python ก็สามารถเชื่อมต่อไปยังโปรแกรมที่มีความสามารถออกรายงานได้ เช่น Python บน Microsoft Windows ก็ออกรายงานผ่าน MS Excel หรือ Python บน Linux หรือ Mac OS X ก็เรียกใช้ให้ออกรายงานผ่านในรูปแบบ Adobe PDF เรื่องต่อมาก็เป็นส่วนของการรองรับภาษาไทย สำหรับการป้อนข้อมูลผ่านจอภาพ (GUI) ของ Python ด้วยที่ Python ในแต่ละเวอร์ชันได้รับการปรับปรุง เพื่อให้รองรับปัญหาในเรื่องนี้ได้ การรองรับการใช้ภาษาคอนข้างมีความสำคัญ ดังนั้น Python ก็กับการป้อนข้อมูลเข้าหรือแสดงตัวอักษรภาษาต่างๆ ด้วยรหัสที่นิยมใช้ เช่น รหัส ASCII ซึ่งเป็นมาตรฐานของการใช้งานด้วย และรหัสแบบ UTF-8 กรณีอยู่ต่างระบบปฏิบัติการ เช่น Linux หรือ Mac OS X เป็นต้น

บทที่ 2

โครงสร้างและลักษณะการเขียนโปรแกรม

สถาปัตยกรรมของ Python ได้รับการออกแบบให้นำไปพัฒนารองรับงาน แบ่งได้เป็น 2 ประเภท ได้แก่ การพัฒนางานประเภทแอปพลิเคชัน (Application) และเว็บแอปพลิเคชัน (Web Application) ซึ่งวิธีการเขียนโปรแกรม การใช้คำสั่ง รูปแบบการใช้งาน เทคนิคการเขียนทั้งหลายไม่ได้แตกต่างกันมาก โดยหากนำภาษา Python ไปสร้างงานเว็บแอปพลิเคชัน ก็ต้องเพิ่มเติมความรู้เกี่ยวกับภาษา HTML, CSS, Java Script และการจัดการดูแลโปรแกรมบนเว็บเซิร์ฟเวอร์ จึงจะทำให้การเขียนเว็บโปรแกรมเป็นไปได้ อย่างเหมาะสมต่อการนำไปใช้งานในแต่ละโปรเจกต์ (ดังรูปที่ 2.1)


ขอบเขตและขีดความสามารถของ Python ในการนำไปเขียนโปรแกรมใช้งานเชิงลึกร่วมกับไลบรารีมาตรฐาน หรือบางครั้งก็เรียกว่า Standard Module ของ Python (Standard Module เป็นที่อยู่ของกลุ่มคำสั่งที่มีใช้งานใน Python โดยแบ่งตามชนิดของกลุ่มที่จะนำไปใช้งาน เช่น Standard Module เกี่ยวกับ OS ก็จะเป็นกลุ่มคำสั่งในการจัดการเกี่ยวกับ การคัดลอก (Copy), การตัด (Cut) การสร้างและคำสั่งอื่นๆ สำหรับไฟล์, โพลเดอร์ในฮาร์ดไดรฟ์ของเครื่องคอมพิวเตอร์เรา เป็นต้น) ก็ขึ้นอยู่กับงานที่พัฒนาขึ้นมา ต้องการให้กลุ่มคำสั่งอะไรบ้างสำหรับใช้งานในโปรแกรมที่กำลังพัฒนา ยกตัวอย่าง เราต้องการอ่านข้อมูลจากไฟล์ในไดรฟ์ C แล้วทำการจัดส่งอีเมล (E-mail) ให้กับลูกค้า ซึ่งในโปรแกรมของ Python จำเป็นต้องเพิ่มโมดูลเกี่ยวกับการจัดการ OS และโมดูลเกี่ยวกับอีเมล (SMTP) เพื่อเรียกใช้คำสั่งให้ครบถ้วนตามกระบวนการทำงานดังที่กำหนดไว้ในตอนต้น (ดังรูปที่ 2.2)

สามารถยืมและติดตามหนังสือใหม่ได้ที่ ระบบห้องสมุดอัตโนมัติ WALAI AutoLib

<http://lib.rmutp.ac.th/catalog/BibItem.aspx?BibID=b00103306>



คู่มือเรียน เขียนโปรแกรม Python (ภาคปฏิบัติ) / วิชิตพันธ์ หล่อเลิศสุนทร และ ฐิตะพันธ์ หล่อเลิศสุนทร.

Author	วิชิตพันธ์ หล่อเลิศสุนทร
Published	กรุงเทพฯ : คอร์ฟังก์ชั่น, 2562
Detail	360 หน้า : ภาพประกอบ ; 23 ซม
Subject	ไพธอน (ภาษาคอมพิวเตอร์)(+) การเขียนโปรแกรม (คอมพิวเตอร์)(+)
Added Author	ฐิตะพันธ์ หล่อเลิศสุนทร
ISBN	9786167502878
ประเภทแหล่งที่มา	 Book

 Multi view  View map

"สำหรับเพื่อการศึกษาระดับปริญญาตรีและการอ้างอิงเท่านั้น"