

```
temp = int(input("Temperatures: "))  
  
while temp >= 42:  
    if temp < 42:  
        print("Too cold!")  
    temp = int(input("Temperatures: "))  
else:  
    print("Too hot!")  
    temp = int(input("Temperatures: "))  
  
print("Just right!")
```

```
population = 700000000  
year = 0  
  
cust = int(input("Number of initial customers: "))  
while cust < population:  
    cust = cust * 2  
    year += 1  
print("Number of years: " + str(year))
```

Python 3.3 ขั้นพื้นฐาน

ฉบับพิเศษ FREE

ฉบับเต็ม กำลังจะตามมาฉบับ...

เต็มไปด้วยแบบฝึกหัดโปรแกรม
ตัวอย่างโปรแกรม
และตัวอย่างคำตอบ

```
n = int(input("How many sheep: "))  
i = 1  
while i <= n:  
    print(str(i) + " sheep")  
    i += 1
```

```
population = 700000000  
year = 0  
  
cust = int(input("Number of initial customers: "))  
while cust < population:  
    cust = cust * 2  
    year += 1  
print("Number of years: " + str(year))
```

“Python 3.3 ขั้นพื้นฐาน” ฉบับพิเศษ FREE

สงวนลิขสิทธิ์ พ.ศ. 2558 โดย Paopeng Pty Ltd, Australia
ห้ามลอกเลียนแบบไม่ว่าส่วนหนึ่งส่วนใดของหนังสือนอกจากจะ
ได้รับอนุญาตจากผู้เขียน มิฉะนั้นจะถือว่าละเมิดลิขสิทธิ์และมี
ความผิดทางอาญา

หากพบว่าหนังสือมีข้อผิดพลาดหรือไม่ได้มาตรฐาน โปรดติดต่อ
ผู้เขียนทาง

Twitter: @JohnJPP168

Facebook fanpage: อ.จอห์น เผ่าเพ็ง

Dedicate กราบขอบคุณ

กราบขอบคุณ

คุณย่า ทองอินทร์ เผ่าเพ็ง

คุณอา สุกัญญา เผ่าเพ็ง

คุณอา ถวิล บุญทศ (เผ่าเพ็ง)

ที่อบรมปณิธาน ส่งเสียเล่าเรียน เลี้ยงดูมาตั้งแต่เด็ก ถ้าไม่มีอา ไม่มี
ย่า เราก็ไม่มีวันนี้

กราบขอบคุณ

คุณพ่อ ไสภณ เผ่าเพ็ง

คุณแม่ ทรัพย์สิน เวฬุวนารักษ์

บุพการีผู้ให้กำเนิดและ unconditional love ที่ให้ลูกมา

กราบขอบคุณ

ครูบาอาจารย์ทุกคนที่เมืองไทย ที่อบรมสั่งสอนเรามา ถ้าไม่ได้
อาจารย์ที่เมืองไทย เราก็ไม่มีวันนี้

คำนำ

หนังสือ Python 3.3 ขั้นพื้นฐาน ทั้งหมดจะมีเนื้อหา 9 บทนะครับ เราจะทำออกมาก่อนเป็นฉบับพิเศษแค่ 4 บทก่อนเพื่อให้ผู้อ่านได้ลองอ่านดูด้วยราคาที่ไม่แพงมากนัก **ส่วนเล่มนี้ก็ทำแจกฟรีนะครับ สำหรับเนื้อหาบทแรก** และหากผู้อ่านต้องการให้เราดัดแปลงหรือปรับปรุงอะไรก็สามารถติดต่อผู้เขียนได้ทาง Twitter หรือ Facebook fanpage นะครับ

เนื่องด้วยผู้เขียนเองเป็นอาจารย์อยู่ที่ประเทศออสเตรเลีย เนื้อหาและคำศัพท์หลาย ๆ คำอาจจะใช้ภาษาอังกฤษไปเลย ถ้าหากผู้อ่านต้องการให้เรามีการแก้ไขอะไร ก็ติดต่อกลับมาได้นะครับ เราพร้อมน้อมรับคำแนะนำของท่านทุกอย่าง

หนังสือเล่มนี้เหมาะสำหรับทุกท่านที่ต้องการเรียนรู้การเขียนโปรแกรมภาษา Python ซึ่งเป็นภาษาใหม่ กำลังได้รับความนิยมเป็นอย่างดีเลยทีเดียว

หนังสือฉบับเต็มกำลังจะออกมาภายในเร็ว ๆ นี้

ผู้เขียนก็หวังว่าจะได้มีโอกาสออกหนังสือ Python 3.3 สำหรับ
intermediate และ advance ให้ได้อ่านกันด้วยนะครับ

อ.จอห์น เผ่าเพ็ญ

สารบัญ

บทที่ 1: เริ่มต้นง่าย ๆ	6
บทที่ 2: การคำนวณ	35
บทที่ 3: การตัดสินใจ	61
บทที่ 4: ว่าด้วยเรื่องของ String	94
เปรียบเทียบคำตอบ	1 18

บทที่ 5-9 กำลังจะตามมานะครับ ฉบับเต็ม

Twitter: @JohnJPP168

Facebook fanpage: อ.จอห์น เผ่าเพ็ญ

บทที่ 1: เริ่มต้นง่าย ๆ

อะไรคือ Python

Python คือภาษาโปรแกรม computer language หรือ programming language. Python เป็นภาษาใหม่ คนรู้จักกันยังไม่แพร่หลายถ้าเปรียบเทียบกับภาษา C, C++ หรือ Java.

Python สามารถใช้งานได้ทั้งบนระบบ Windows, MacOS, Linux และก็ระบบอื่น ๆ ซึ่งทางหน่วยงานที่รับผิดชอบจะมีการเพิ่มระบบต่าง ๆ เข้ามาเรื่อย ๆ ที่สามารถ run Python ได้

Python สามารถดาวน์โหลดได้ฟรีนะครับที่ <https://www.python.org>

Python ที่ผมจะสอนก็จะเป็นเวอร์ชัน 3.3 นะครับ ซึ่งเวอร์ชันปัจจุบัน ณ เดือน มิถุนายน 2558 คือเวอร์ชัน 3.4 แต่ commands และฟังก์ชันทุกอย่างที่เขียนในเวอร์ชัน 3.3 ก็สามารถใช้งานได้ ในเวอร์ชัน 3.4 เพราะ compiler ที่ผมใช้ก็เป็นเวอร์ชัน 3.4 ซึ่งก็ทำงานได้ดี ไม่มีปัญหาอะไรนะครับ

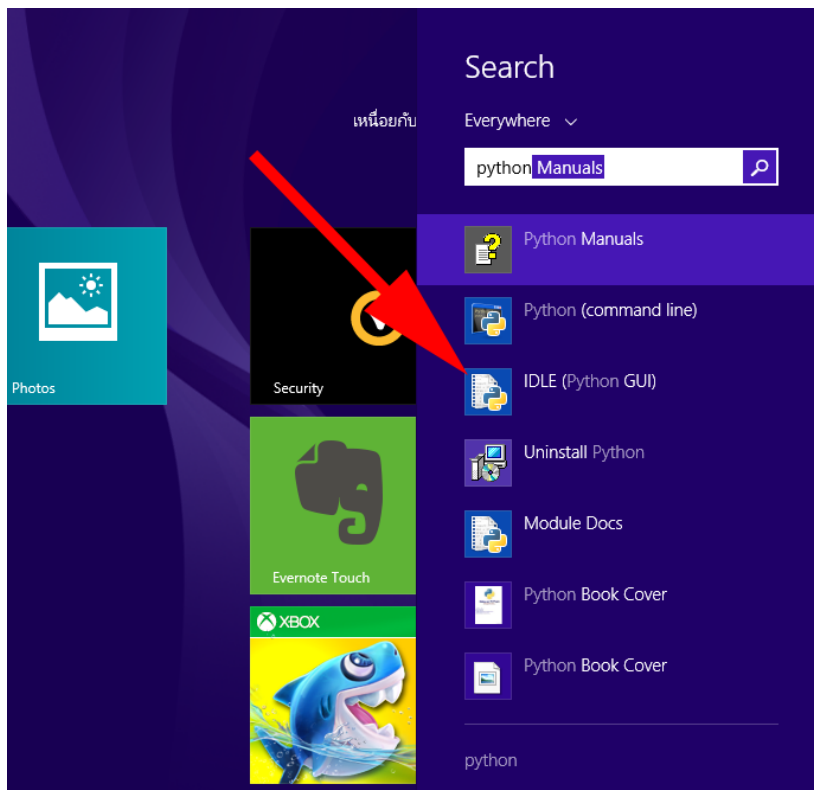
การใช้โปรแกรม Python

อันดับแรกคือ เราเรียกโปรแกรม Python ออกมาทำงาน ผมแนะนำ

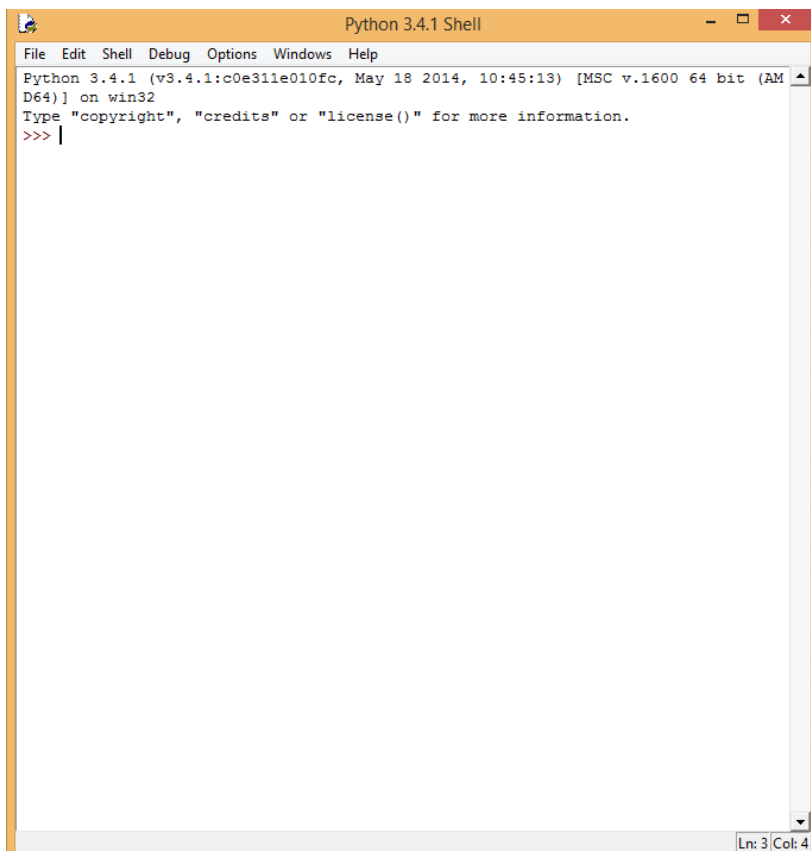
ให้ใช้แบบ IDLE Python นะครับ คือเป็น GUI; Graphical User

Interface, ไม่ใช่แบบ command line เพราะการใช้แบบ GUI จะใช้งานได้

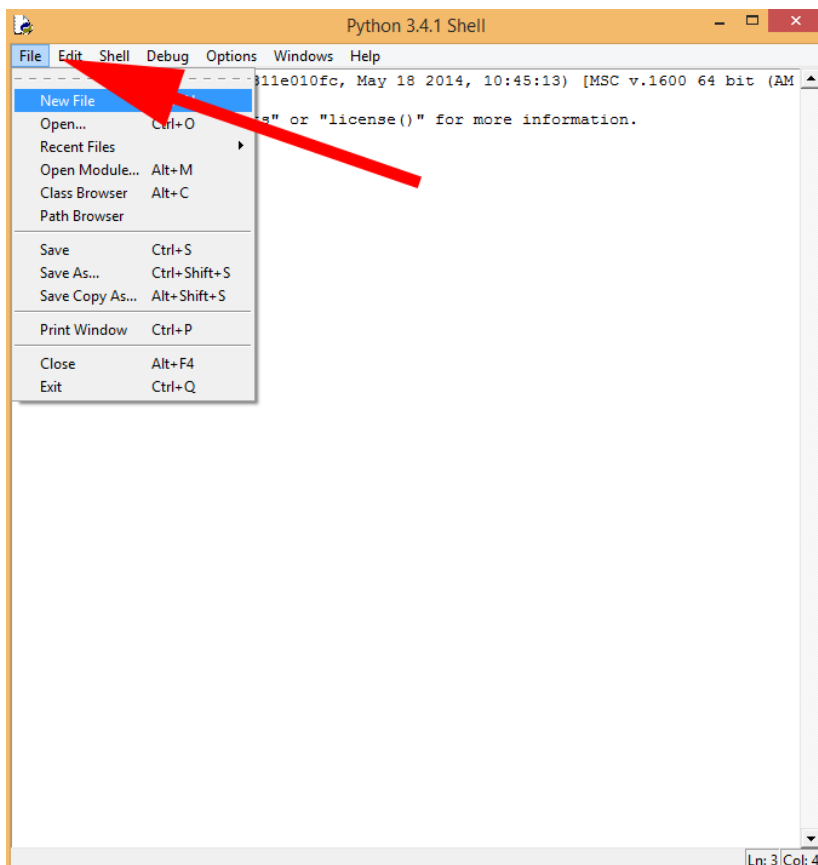
ง่ายมากกว่า



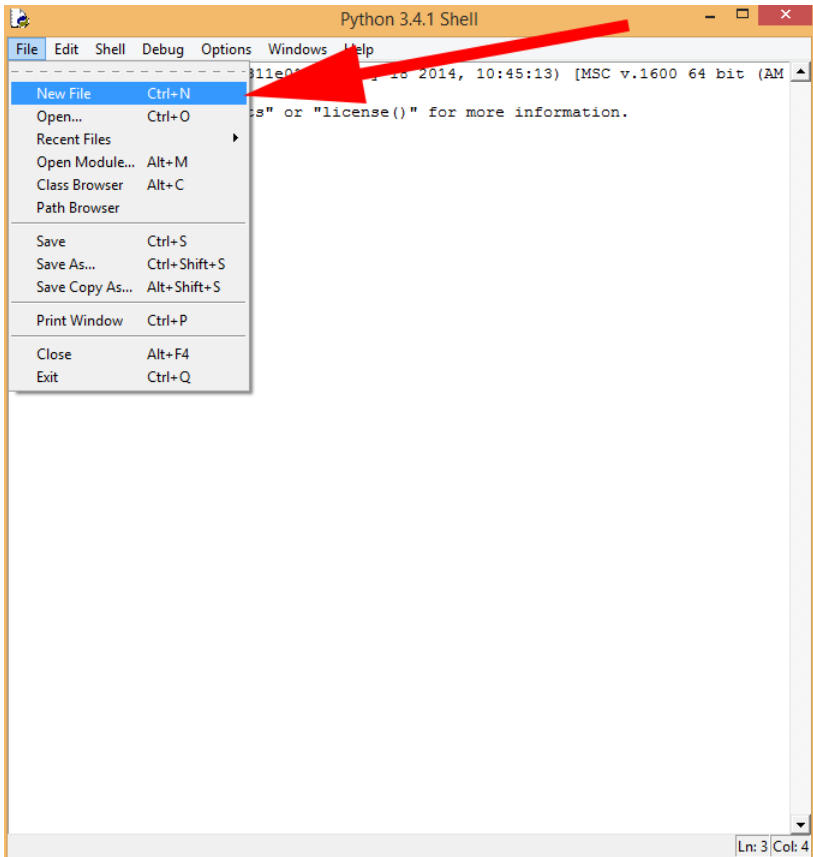
พอเปิดโปรแกรมมา หน้าตาโปรแกรมก็จะเป็นแบบนี้ละครับ ผมใช้เวอร์ชัน 3.4 ในการ run โปรแกรม แต่สไลด์การเขียนโปรแกรมก็ยังเป็น 3.3 นะครับ



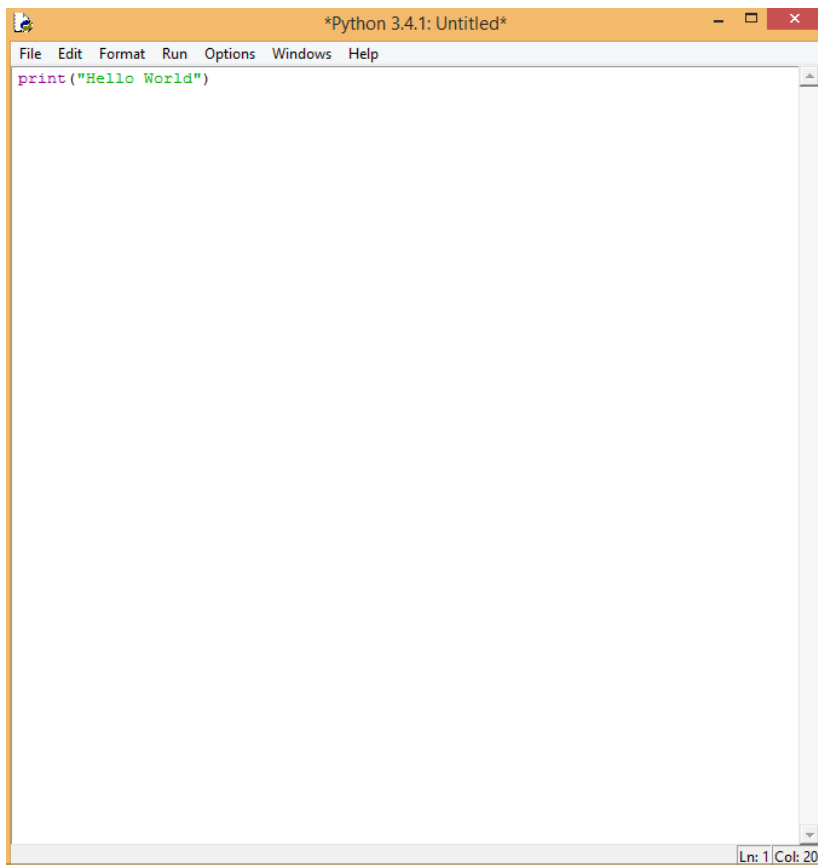
ในการเขียนโปรแกรม ผมก็แนะนำให้เขียนไปใบไฟล์ ไม่ใช่เขียนใน shell เพราะการเขียนในไฟล์ ถ้ามีอะไรผิดพลาด เราก็สามารถกลับมาแก้ไขโค้ดโปรแกรมเราได้ ก็แนะนำให้ทำตามรูปนะครับ กด File



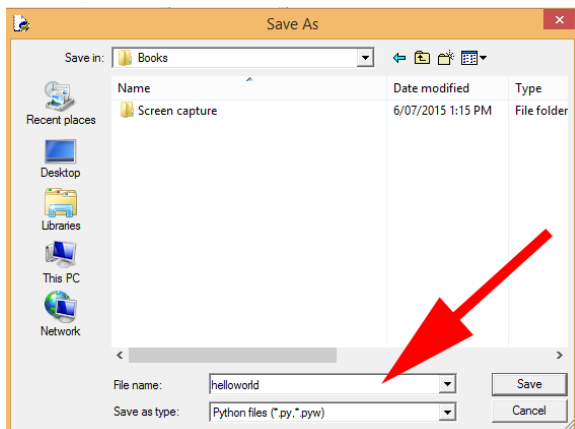
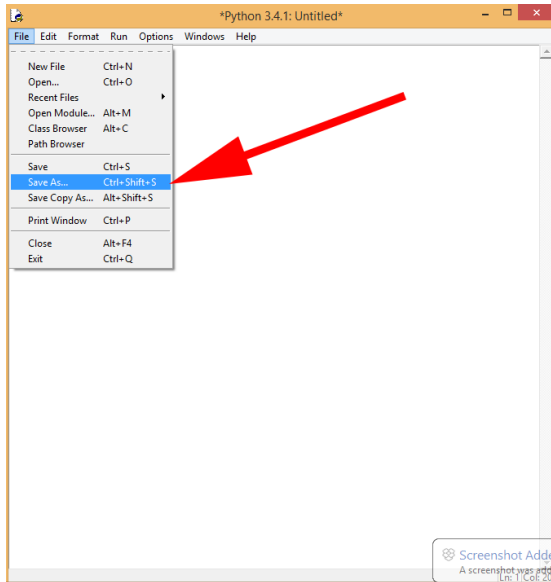
แล้วก็ตามด้วย New File



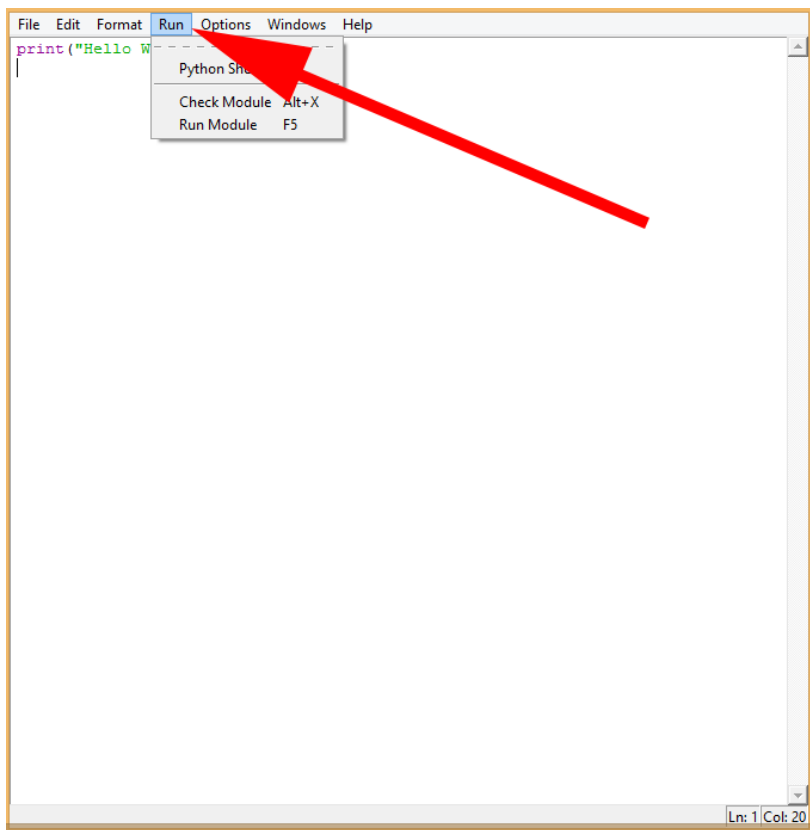
เสร็จแล้วก็เริ่มเขียนโปรแกรมได้เลยนะครับ ผมจะเริ่มเขียน
โปรแกรมแบบง่าย ๆ คือ โปรแกรม Hello World ที่โปรแกรมเมอร์
ส่วนมากรู้จัก

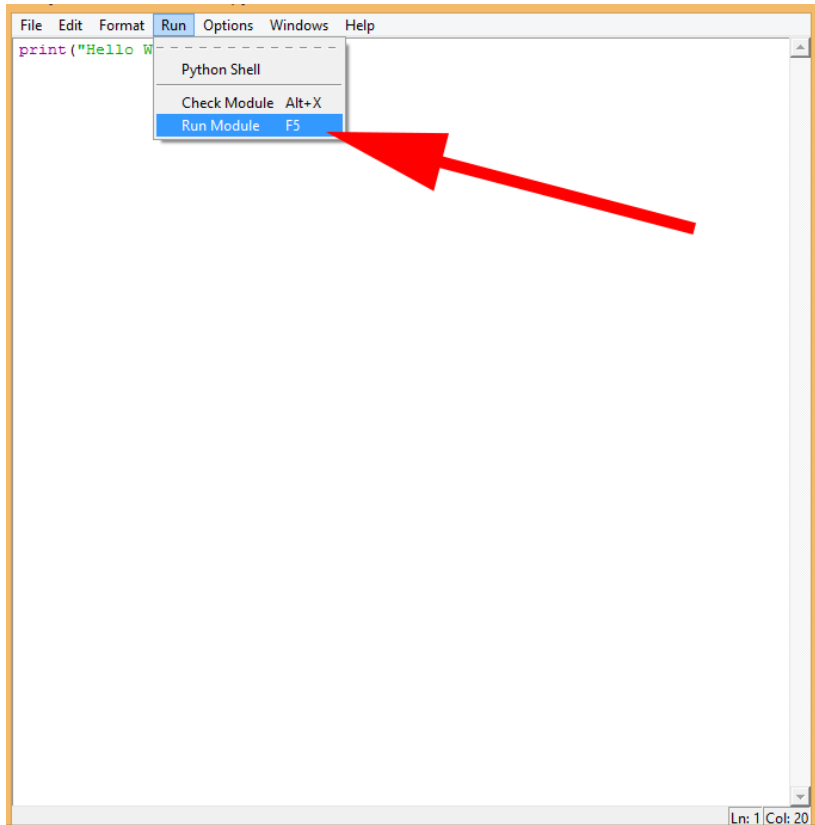


เสร็จแล้วเราก็ save ไฟล์ โดยเราจะ save เป็น Python ไฟล์ (.py) โดยการกด File แล้วก็ตามด้วย Save As แล้วเราก็ใส่ชื่อไฟล์ไปตามที่เราต้องการ ผมใส่ชื่อไฟล์เป็น helloworld.py นะครับ



การ run โปรแกรม เราก็สามารถ run ได้ง่าย ๆ โดยกด Run แล้วก็ตามด้วย Run Module หรือ เราสามารถกด F5 ก็ได้





โปรแกรมก็จะ run โดยการแสดงผลหน้าจอ ออกมาเป็นข้อความ
`Hello World` นะครับ

ผมก็แนะนำให้ลองทำตาม steps ดูนะครับ



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:45:13) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Hello World
>>>
```

หากใครมีปัญหาในการใช้โปรแกรม หรือ run โปรแกรม ติดต่อผมได้
นะครับ ทาง

Twitter: @JohnJPP168

Facebook fanpage: อ.จอห์น เผ่าเพ็ญ

นะครับ

แบบฝึกหัด 1.1: Hello, World!

สังเกตกันหรือเปล่าครับว่า โปรแกรมแรกของทุกภาษาจะเป็นโปรแกรม Hello World ตลอด ไม่ว่าจะเป็น C, C++, Java, COBOL หรือแม้แต่ HTML5. ปกติผมจะให้ลูกศิษย์เขียนโปรแกรมแรกเป็น Hello World ตลอด เพราะเป็นอะไรที่เขียนได้ง่าย ๆ

การเขียน `Hello World` ใน Python ก็เขียนแบบนี้ครับ

```
print('Hello, World!')
```

จะเขียนโปรแกรмыังไง

จะสังเกตว่าตัวหนังสือและสัญลักษณ์จะมีสีที่แตกต่างกันตอนเขียนโปรแกรม

```
print("Hello, World!")
```

การเขียนโปรแกรมคอมพิวเตอร์ก็เหมือนที่เวลาเราพูดหรือเขียน ไม่ว่าจะเป็นภาษาไทย ภาษาอังกฤษ หรือภาษาอะไร เราก็ต้องพูดหรือเขียนไปตามหลักไวยากรณ์ ไม่งั้นคนฟังหรือคนอ่านก็จะฟังหรืออ่าน

ไม่รู้เรื่อง ส่วนไวยากรณ์ทางด้านคอมพิวเตอร์ เราก็เรียกว่า syntax ซึ่งหลาย ๆ คนก็คงเคยได้ยินคุ้น ๆหูหรือเคยเห็นกันมาบ้างแล้ว

Python จะใช้สัญลักษณ์ในการบ่งบอกถึง syntax ต่าง ๆ เพื่อช่วยให้โปรแกรมเมอร์หรือคนเขียนโปรแกรม ได้เขียนและอ่านโปรแกรมได้ง่ายขึ้น

```
print("Hello, World!")
```

สีม่วงของ `print` นั่นคือ function (ฟังก์ชัน)

สีเขียวของ `"Hello, World!"` นั่นคือ string (สตริง), เดี่ยวเราจะได้เรียนเรื่อง string กันต่อไปว่า string คืออะไร

ดังนั้นเวลาเราเขียนโปรแกรม เราก็ต้องสังเกตสีของโปรแกรมด้วย ถ้าเราเห็นสีของโปรแกรมแปลก ๆ ก็แสดงว่าเราคงเขียนโปรแกรมผิดแล้วแหละ

จะเกิดอะไรขึ้นเมื่อเราเขียนโปรแกรมผิด

Python จะมีอินเตอร์เพรเตอร์ interpreter ซึ่งเป็นโปรแกรมที่อ่าน และ runs โปรแกรมของเราที่เขียน ถ้าหากเราเขียนโปรแกรมผิด syntax, interpreter ก็จะมีการแจ้ง error อย่างเช่น `SyntaxError` หรือ `NameError`.

ยกตัวอย่างการเขียนผิด syntax ของ `print()`:

```
write('Hi There')
```

Python ไม่เข้าใจฟังก์ชันของ `write` เพราะ `write` ไม่ใช่ฟังก์ชันของ Python, ดังนั้น interpreter ก็จะมีการแจ้ง error ดังต่อไปนี้

Traceback (most recent call last):

File "program.py", line 1, in <module>

write('Hi There')

NameError: name 'write' is not defined

Interpreter จะมีการแจ้ง error เป็นสีแดง ซึ่งก็จะบอกประเภทของ error และก็จะบอกว่าอยู่บรรทัดไหนด้วย ตัวอย่างของ error ข้างบน ประเภทของ error คือ **NameError** อยู่บรรทัดที่ 1

Syntax error ที่เป็นสีแดงก็จะช่วยให้เราหา error ได้ง่ายขึ้น จะสังเกตว่า **write** ไม่ใช่ฟังก์ชันของ Python ดังนั้น Python จะไม่มีการเปลี่ยนเป็นสีม่วงเหมือนฟังก์ชัน **print()**.

มี Syntax Error จะทำยังไงดี

ถ้าเรามี error ก็ไม่ต้องตกใจนะครับ มันเป็นเรื่องปกติของการเขียนโปรแกรมอยู่แล้ว เพียงแต่เราต้องเรียนรู้ในการแก้โปรแกรมหรือ debug เท่านั้นเอง มันเป็นส่วนหนึ่งของชีวิตโปรแกรมเมอร์

แต่ตัวอย่างที่ผมยกมาเป็นตัวอย่างที่เราสามารถแก้ไขได้ง่าย โดยเปลี่ยนจาก **write()** เป็น **print()** โปรแกรมก็จะทำงานได้ตามปกติ **print("Hello")**

ที่นี่ให้เราลองเขียนดูนะครับ

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

ดูนะครับ จะสังเกตว่าเราลืมปิด string "Hello ด้วยเครื่องหมาย
คำพูด หรือ double quotation. Interpreter คราวนี้ก็จะแจ้งเป็น

SyntaxError ไม่ใช่ NameError

เดี๋ยวมาลองดูอีกตัวอย่างหนึ่งนะครับ

```
print "Hello"
```

ตอนนี้ interpreter ก็จะมีการแจ้ง SyntaxError เพราะเราลืมใส่วงเล็บ

```
("Hello")
```

การแสดงความบนหน้าจ

ข้อความที่อยู่ใน double quotation หรือ single quotation

(เครื่องหมายคำพูด) ภาษาศัพท์ของโปรแกรมเมอร์เราเรียกว่า

string (สตริง), string สามารถเป็นข้อมูลที่เป็นตัวหนังสือยาว ๆ หรือ

เป็นแค่ตัวหนังสือตัวเดียว หรือสัญลักษณ์อะไรก็ได้ที่มาจากแป้น

keyboard ขอเพียงแต่เราใส่ double quotation หรือ single quotation

เข้าไปเท่านั้นแหละ ทุกอย่างที่อยู่ใน quotation หรือเครื่องหมาย

คำพูดนี้ก็จะกลายเป็น string ไปเลยทันที

Python เวอร์ชัน 3.3 และ 3.4 สามารถใช้เครื่องหมายคำพูดได้ทั้ง

แบบ double quotation หรือ single quotation ถ้าเราใช้ double

quotation ในการเปิด string เราก็ต้องใช้ double quotation ในการปิด

string ด้วย โดยนิสัยส่วนตัวของผมแล้ว ผมจะใช้ double quotation

เพราะชินกับการเขียนภาษา C++ และ Java ผมว่า double quotation

จำได้ง่ายกว่านะ ไม่สับสน ดูตัวอย่างดังต่อไปนี้ครับ

```
print("hello")
```

```
hello
```

```
print('hello')
```

```
hello
```

แต่ถ้าเราเปิด string ด้วย double quotation แล้วปิดด้วย single quotation

```
print("hello')
```

โปรแกรมก็จะมี **SyntaxError** อย่างที่เห็น

File "program.py", line 1

```
print("hello')
```

^

SyntaxError: EOL while scanning string literal

Variable (วาริเอเบิล) คืออะไร

หากเราต้องการใช้ค่าหรือ value ในโปรแกรม เราก็สามารถเก็บค่านั้นได้ใน variable หรือ **ตัวแปร** คือเราต้องเก็บค่าข้อมูลไว้ในตัวแปรก่อน เราถึงจะสามารถเรียกค่าข้อมูลออกมาใช้งานได้

Variable ก็เปรียบเหมือนตู้รับจดหมายที่บ้าน เราสามารถเอาข้อมูลหรือ value เก็บไว้ในตู้รับจดหมายได้ แล้วเราก็สามารถเปิดดูข้อมูลในตู้จดหมายนั้นได้เช่นกัน

การเอาข้อมูลใหม่ใส่ไปเข้าไปใน variable มันก็จะเป็นการเอาข้อมูลใหม่ไปแทนที่ข้อมูลเก่า ข้อมูลเก่าก็จะหายไปจาก computer memory หรือหน่วยความจำของคอมพิวเตอร์ คอมพิวเตอร์ก็จะมีตู้รับจดหมายอยู่หลายล้านตู้ในหน่วยความจำ ตัวอย่างเช่นคอมพิวเตอร์ที่มีความจำ 256MB ของ RAM (Random Access Memory) ก็จะมีประมาณ 67,000,000 ตู้รับจดหมาย (67 ล้าน)

Python และ ภาษาคอมพิวเตอร์อื่น ๆ ก็จะมีวิธีการที่จะให้เราจำได้ว่าข้อมูลแต่ละตัวอยู่ตู้รับจดหมายช่องไหน โดยการตั้งชื่อ หรือ

identifier ซึ่งก็จะเป็นง่ายสำหรับเราที่เป็นโปรแกรมเมอร์ เพราะดีกว่าที่จะจำว่า ช่องผู้รับจดหมายหมายเลข 4756860978 มีข้อมูลโน้น ข้อมูลนี้

การตั้งค่าข้อมูลหรือ value ของ variable ก็ทำได้ด้วยการใช้เครื่องหมายเท่ากับ = ซึ่งก็เปรียบเสมือนเอาค่าข้อมูลมาเก็บไว้ในตู้รับจดหมาย ยกตัวอย่างเช่น

```
msg = "Hi there"  
print(msg)
```

โปรแกรมข้างบนก็เป็นการตั้งค่า string ซึ่งค่าของ string ก็คือ "Hi there" และก็เป็นการใช้ค่าแก่ variable ที่ชื่อ msg

ถ้าเรา run โปรแกรม, โปรแกรมก็จะพิมพ์ข้อความออกหน้าจอต้งปรากฏข้างล่าง

Hi there

พิมพ์หน้าจอหลาย ๆ ข้อความ (string)

เราสามารถใช้ฟังก์ชัน `print()` ครั้งเดียวในการพิมพ์ข้อความหลาย ๆ ข้อความบนหน้าจอได้ ไม่จำเป็นต้องใช้ ฟังก์ชัน `print()` หลาย ๆ รอบ เพียงแต่เราต้องใช้ comma (,) ในการแยกข้อความออกจากกันเท่านั้นเอง ยกตัวอย่างเช่น

```
print("John", "Paopeng")
```

John Paopeng

จะสังเกตเห็นว่าข้อความหรือ string จะมีช่องว่าง space เข้ามาگیرระหว่างข้อความด้วย

ในขณะเดียวกันเราก็สามารถใช้ variable ในการเขียนแสดงผลหลาย ๆ ข้อความบนจอได้ ยกตัวอย่างเช่น

```
firstname = 'John'  
lastname = 'Paopeng'  
print(firstname, lastname)
```

John Paopeng

จากตัวอย่างที่ผ่านมา ศัพท์ทางเทคนิคเราเรียกว่า การส่งผ่าน argument (อาร์กิวเมนต์) เข้าไปในฟังก์ชัน ซึ่งเราจะได้เรียนรู้ในหัวข้อต่อไปนะครับ

รับข้อมูลจากผู้ใช้โปรแกรม

เราสามารถเขียน Python โปรแกรมเพื่อรับข้อมูลจากผู้ใช้โปรแกรมได้ โปรแกรมข้างล่างจะถามชื่อผู้ใช้โปรแกรม และก็จะพิมพ์แสดงผลออกทางหน้าจอ

```
name = input("What is your name? ")  
print(name)
```

ทดลองเขียนและก็ run โปรแกรมดูนะครับ หลังจาก run โปรแกรม เราจะเห็นเครื่องหมาย cursor เคอร์เซอร์ หลังข้อความ `What is your name?` รอให้เราใส่ข้อมูลลงไป ก็แนะนำให้ใส่ชื่อเราลงไปแล้วก็กด Enter นะครับ

What is your name? Paopeng

Paopeng

แบบฝึกหัด 1.2: Hello to you, too

เราเรียนรู้การเขียนโปรแกรม Python เบื้องต้นกันมาแล้ว ลองมาทำแบบฝึกหัดดูนะครับ บทเฉลยอยู่ด้านหลังนะครับ แต่ก็แนะนำให้ลองเขียนดูก่อน ก่อนที่จะดูเฉลยหรือเปรียบเทียบคำตอบ

แบบฝึกหัดนี้ ให้เขียนโปรแกรมเพื่อถามชื่อผู้ใช้งานโปรแกรม เสร็จแล้วก็ให้ตอบผู้ใช้งานโปรแกรมกลับด้วยคำว่า Hello, แล้วตามด้วยชื่อผู้ใช้งานโปรแกรมที่เขาพิมพ์ลงมา สังเกตว่ามีคอมมา (,) ด้วยนะครับ

ยกตัวอย่างเช่น ถ้าผมเป็นผู้ใช้งานโปรแกรม หน้าตาโปรแกรมก็จะออกมาแบบนี้

What is your name? Paopeng

Hello, Paopeng

หรือคนอื่นที่ใช้งานโปรแกรม หน้าตาก็น่าจะออกมาอีกแบบหนึ่ง

What is your name? Sukanya

Hello, Sukanya

เสร็จแล้วก็ลองเปรียบเทียบคำตอบที่ท้ายเล่มดูนะครับ

เขียนข้อความมารวมกัน

เราได้เรียนรู้การเขียนข้อความรวมกันโดยการส่งผ่าน argument (อาร์กิวเมนต์) เข้าไปในฟังก์ชันกันมาแล้ว เราจะมาดูอีกวิธีหนึ่งที่ สามารถแสดงผลการเขียนข้อความรวมกันได้ เราใช้การรวม string เข้าด้วยกันด้วยเครื่องหมายบวก + ภาษาทางโปรแกรมมิ่งเรา เรียกว่า string concatenation (สตริง คอนแคทนิเนชัน)

ยกตัวอย่างเช่น ถ้าเราต้องการแสดงผลหน้าจอคำว่า "John" แล้ว ตามด้วยคำว่า "Paopeng", เราก็สามารถเขียนได้ว่า

```
print("John" + "Paopeng")
```

JohnPaopeng

จะสังเกตว่าการเขียนแบบนี้ string concatenation โดยใช้เครื่องหมายบวก + นี้ การแสดงผลหน้าจอก็จะไม่มีการใส่ช่องว่างหรือ space ลงไประหว่างข้อความ ถ้าเราต้องการช่องว่างหรือ space เราก็ต้องใส่เอง เช่น

```
print("John" + " " + "Paopeng")
```

John Paopeng

หรือ

```
print("John " + "Paopeng")
```

John Paopeng

ตัวอย่างที่ 2 จะมีช่องว่างหรือ space หลังจาก John แล้วค่อยปิด double quotation "John "

เขียนข้อความมารวมกันโดยใช้ variables

ในขณะเดียวกัน เราก็สามารถใช้ variable หรือตัวแปร ในการเขียนข้อความมารวมกันแบบ string concatenation ได้ ดังตัวอย่างข้างล่าง

```
firstname = "John"
lastname = "Paopeng"
print(firstname + " " + lastname)
John Paopeng
```

อย่าลืมบวกช่องว่างหรือ space เข้าไปด้วยนะครับ

เราสามารถใส่เครื่องหมายตกใจหรือเครื่องหมายอะไรต่าง ๆ เข้าไป
ในข้อความเราได้ด้วย ตัวอย่างเช่น

```
forecast = "hot!"
print("Today is so " + forecast)
```

ก็ลองเขียนและ run โปรแกรมดูนะครับ อย่าลืมใช้ช่องว่าง หรือ
space นะครับ

การใช้เครื่องหมายคำพูดหรือ quotation

แล้วถ้าเราอยากใช้เครื่องหมาย quotation ในประโยคข้อความล่ะ?

ถ้าเราเอา double quotation เข้าไปอยู่ข้างในของ double quotation

ลองเขียนและ run ดูนะครับ จะรู้ว่าจะ run ไม่ผ่าน

```
name = "Paopeng"  
  
print(name + " is his name!")
```

สาเหตุที่โปรแกรม run ไม่ผ่านก็เพราะว่า โปรแกรมสับสน ไม่รู้จะใช้

double quotation ไหนดี เพราะมี 2 double quotation อยู่ซ้อนกัน

อย่าลืมนะครับว่า string ใน Python สามารถใช้ได้ทั้ง single quotation

และ double quotation ขอเพียงแต่ว่า ถ้าเราเปิด string ด้วย quotation

ไหน ให้เราปิดด้วย quotation นั้น

```
firstname = "Thong-in"  
  
lastname = "Paopeng"  
  
print(firstname + " " + lastname)
```

Thong-in Paopeng


```
firstname = 'Thong-in'
lastname = 'Paopeng'
print(firstname + ' ' + lastname)
Thong-in Paopeng
```

เราสามารถเขียน double quotation ในประโยคของเราได้ โดยใช้
double quotation เข้าไปอยู่ข้างใน single quotation อ่าน ำดูอาจงง
แนะนำให้ดูตัวอย่างเลยนะครับ

```
name = "Paopeng"
print(name + " is his name!")
"Paopeng" is his name!
```

แบบฝึกหัด 1.3: Under the rain ได้สายฝน

เดี๋ยวเรามาฝึกเขียนโปรแกรมแปลก ๆ กันดูนะครับ เดี่ยวเราเขียนโปรแกรมที่ให้ผู้ใช้โปรแกรมพิมพ์คำตอบเป็น verb แล้วเราแสดงผลออกมาเป็นประโยคทางหน้าจอ อย่างเช่น

Enter a verb: walk

One does not simply "walk" under the rain!

อีกตัวอย่างหนึ่ง

Enter a verb: skip

One does not simply "skip" under the rain!

และอีกตัวอย่างสุดท้าย

Enter a verb: swim

One does not simply "swim" under the rain!

ก็ลองฝึกเขียนและ run โปรแกรมดูนะครับ ตัวอย่างคำตอบอยู่ด้านหลัง แต่นั่นก็เป็นตัวอย่างคำตอบของผมนะครับ การเขียนโปรแกรมนั้นมีคำตอบและการเขียนที่หลากหลาย เราสามารถเขียน

ยังงี้ก็ได้ที่สามารถแสดงผลคำตอบโปรแกรมออกมาตามที่เราต้องการ ขอให้ทำตามคำตอบที่เราต้องการ ใช้ได้หมดครับ ผมก็แนะนำให้ใช้ตัวอย่างคำตอบด้านหลังให้เป็นแค่แนวทางเท่านั้น

แบบฝึกหัด 1.4: เสียงสะท้อน

เคยไหมครับที่เราตะโกนเสียงเข้าไปในหุบเขา แล้วเราก็ได้ยินเสียงสะท้อนออกมา เดี่ยวเรามาเขียน Python โปรแกรมเลียนแบบเสียงสะท้อนกันเถอะ แบบฝึกหัดนี้ให้เขียนโปรแกรมที่รับข้อมูลจากผู้ใช้ แล้วพิมพ์ออกมาแสดงผลหน้าจอ เหมือนเสียงสะท้อน 3 ครั้ง โปรแกรมเราควรแสดงผลตามที่เห็นนะครับ

Shout: hello

hello hello hello

โปรแกรมเราต้องแสดงผล เหมือนเสียงสะท้อน 3 ครั้งนะครับ

เสร็จแล้วก็ลองเปรียบเทียบคำตอบที่ท้ายเล่มดูนะครับ

เปรียบเทียบคำตอบ (จากหน้า 118)

ตามปกติของการเขียนโปรแกรมทั่วไป ไม่ว่าจะภาษาไหน C, C++, Java หรือรวมมาถึง Python ด้วย เราสามารถเขียนโปรแกรมได้หลายแบบ ขึ้นอยู่กับการตีโจทย์ และประสบการณ์ของโปรแกรมเมอร์นะครับ

ตัวอย่างคำตอบแบบฝึกหัดที่ผมเตรียมไว้ให้ แนะนำให้ดูแค่เป็นแนวทาง เป็น guideline นะครับ อย่าคิดว่าเป็นวิธีทางเดียวในการเขียนโปรแกรม หรือนี้เป็นวิธีทางเดียวในการหาคำตอบ ถ้าเราได้คำตอบมาอย่างไร ก็ลองเอามาเปรียบเทียบกับดูนะครับ และก็เรียนรู้จากการเปรียบเทียบคำตอบ

แบบฝึกหัด 1.2: Hello to you, too

ตัวอย่างคำตอบ:

```
name = input("What is your name? ")  
  
print("Hello, " + name)
```

แบบฝึกหัด 1.3: Under the rain ได้สายฝน

ตัวอย่างคำตอบ:

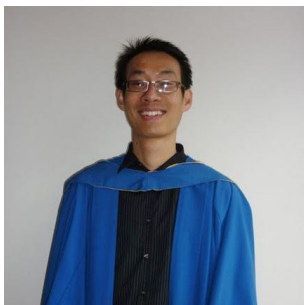
```
verb = input("Enter a verb: ")  
  
print("One does not simply " + verb + " under the rain!")
```

แบบฝึกหัด 1.4: เสียงสะท้อน

ตัวอย่างคำตอบ:

```
word = input("Shout: ")  
  
print(word + " " + word + " " + word)
```

ประวัติคนเขียน



ชื่อ: จอห์น อีระเดช เฝ้าเพ็ง

ประวัติการทำงานย่อ ๆ

โปรแกรมเมอร์: ถูกจองตัวไปทำงานที่ประเทศสิงคโปร์ตั้งแต่เรียนเทอมสุดท้ายที่ [University of Wollongong](http://www.uowollongong.edu.au) พอทำงานอยู่ที่สิงคโปร์ได้ 4 ปีก็กลับมาอยู่ที่ออสเตรเลีย เพราะชอบการใช้ชีวิตอยู่ที่นี่

อาชีพส่วนตัว: พอกลับมาที่ออสเตรเลียก็ทำอาชีพส่วนตัวอยู่ประมาณ 11 ปี และก็เรียนไปด้วยเพิ่มเติม

อาจารย์: เป็นข้าราชการประจำ (fulltime permanent) กับกระทรวงการศึกษาของรัฐ New South Wales (NSW) สอนวิชาคณิตศาสตร์และวิชาคอมพิวเตอร์

ประวัติการศึกษา

ที่ออสเตรเลีย

- Master Degree of Information Systems Security, [Charles Sturt University](#) (กำลังรอสอบตัวสุดท้าย)
- Bachelor Degree of Computer Science, University of Wollongong
- Bachelor Degree of Arts (Japanese), University of Wollongong
- Graduate Diploma in Learning & Teaching, [University of Southern Queensland](#)
- Graduate Certificate in Australian Migration Law & Practice, [The Australian National University](#)
- Graduate Certificate in Management, University of Southern Queensland

ที่เมืองไทย

มหาวิทยาลัยขอนแก่น คณะวิศวกรรมศาสตร์ เกียรตินิยมที่ 30; ปี 1

โรงเรียนชุมแพศึกษา อ.ชุมแพ จ.ขอนแก่น; ม. 2-6

โรงเรียนคอนสารวิทยาคม อ.คอนสาร จ.ชัยภูมิ; ม. 1

โรงเรียนบ้านน้ำพุปางวัว อ.คอนสาร จ.ชัยภูมิ; ป. 5-6

โรงเรียนบ้านสร้างช้างมิตรภาพที่ 143 ต.ไผ่ อ.ทรายมูล จ.ยโสธร; ป.4

โรงเรียนบ้านหนองทองกลาง ต.หนองหิน อ.เมือง จ.ยโสธร; ป. 1-3