

Lecture 2 Getting Started

เริ่มการเขียนโปรแกรม

16 มิถุนายน 2557 13:44

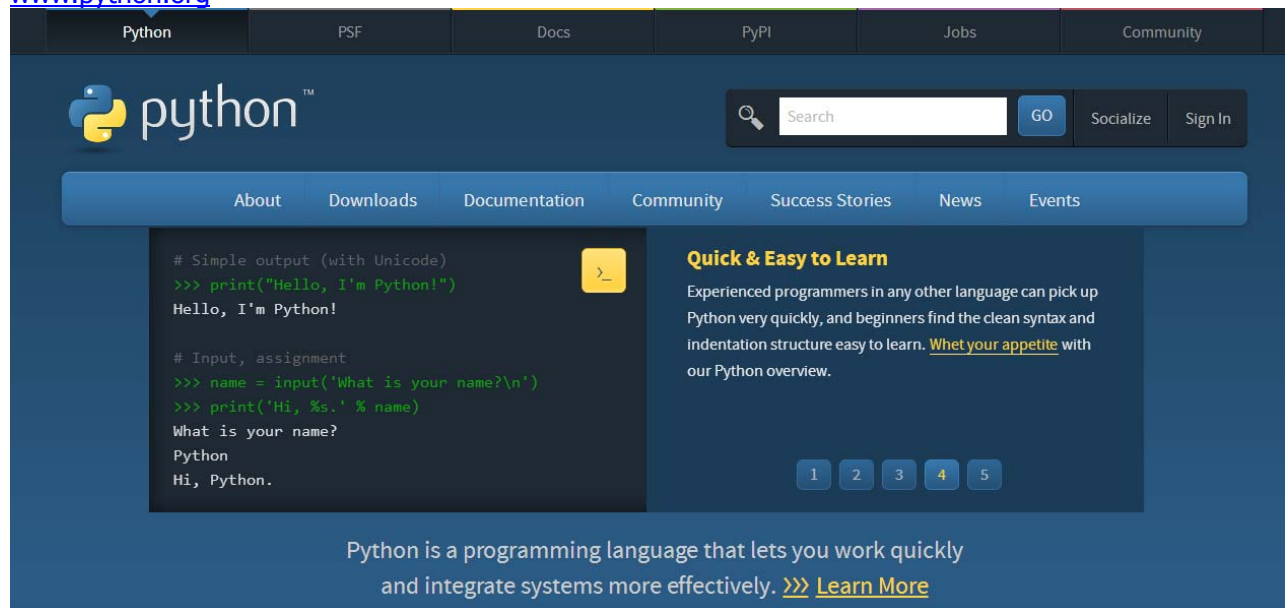
ทุกวันนี้คอมพิวเตอร์ใช้งานง่ายกว่าเดิมเมื่อเปรียบเทียบกับยุคก่อน คุณทราบหรือไม่ว่า ยุคเริ่มต้นของคอมพิวเตอร์ตั้งโต๊ะขาย หรือ คอมพิวเตอร์ส่วนบุคคล หรือ ที่เรียกว่า เครื่องพีซี (PC ย่อมาจาก Personal Computer) คอมพิวเตอร์ส่วนบุคคลมีเพียงแค่ภาษา Basic ที่เป็นภาษาคอมพิวเตอร์หนึ่งแถมมาเพียงเท่านั้น หน้าจอแสดงผลขาวดำเพียงคำว่า "READY" (พร้อมที่ใช้งาน) เท่านั้น ไม่มีหน้าต่างวินโดว (window) ไม่มีโปรแกรมสำเร็จรูป (Software package) ไม่มีโปรแกรมเอกสาร (Word Processing) เว็บเบราว์เซอร์ (Web browser) (ตอนนั้นยังไม่มีแม้แต่อินเทอร์เน็ต) ไม่รูปภาพสีสันใดๆ มีเพียงแค่ข้อความ ถ้าต้องการใช้โปรแกรมใดๆ เพียงแค่คุณเขียนโปรแกรมเอาเองด้วยภาษา Basic

การติดตั้งโปรแกรม (Installing Python)

ดาวน์โหลดโปรแกรม (Download Program)

เราสามารถดูข้อมูลเกี่ยวกับ Python และดาวน์โหลดโปรแกรมได้ที่เว็บไซต์ www.python.org โดยสามารถเข้าไปดาวน์โหลดโปรแกรม Python ได้ที่ www.python.org/download เนื่องจากเรื่องที่เราจะเรียนต่อไปนี้มี GUI และการทำเกมส์ด้วยเครื่องมือที่พัฒนาเพิ่มมาจาก Python คือ pygame สำหรับทำเกมส์โดยเฉพาะและทำงานร่วมกับ python ภาษาโปรแกรมมิ่ง pygame เป็น โมดูล (Module) ชนิดหนึ่ง ซึ่งโมดูลหมายถึง ระบบที่ทำหน้าที่เฉพาะสามารถมาเป็นส่วนประกอบใช้ทำงานเชื่อมต่อกับระบบอื่น ซึ่งมีการดัดแปลงออกแบบและควบคุมภายในระบบของตัวเอง ดังนั้นเพื่อความมั่นใจว่าเวอร์ชัน (version) Python ที่จะติดตั้งถูกต้อง สามารถทำงานร่วมกับ pygame ได้สามารถดาวน์โหลดได้ที่ลิงค์นี้ <http://pygame.info/downloads/>

www.python.org



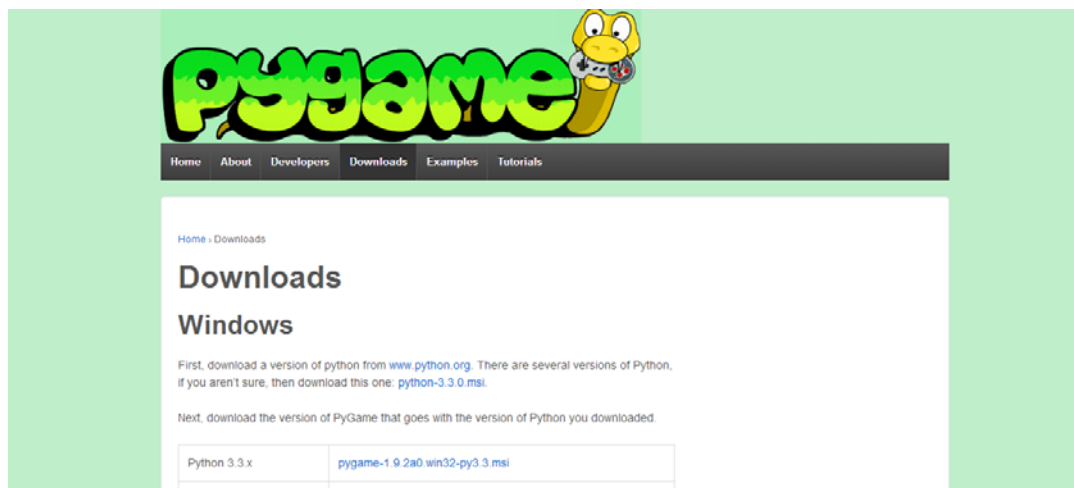
Quick & Easy to Learn

เรียนรู้รวดเร็วและง่าย

Python is a programming language that lets you work quickly and integrate systems more effectively.

การเขียนโปรแกรมด้วย Python ทำให้คุณทำงานเร็วขึ้น และจัดการระบบอย่างมีประสิทธิภาพ

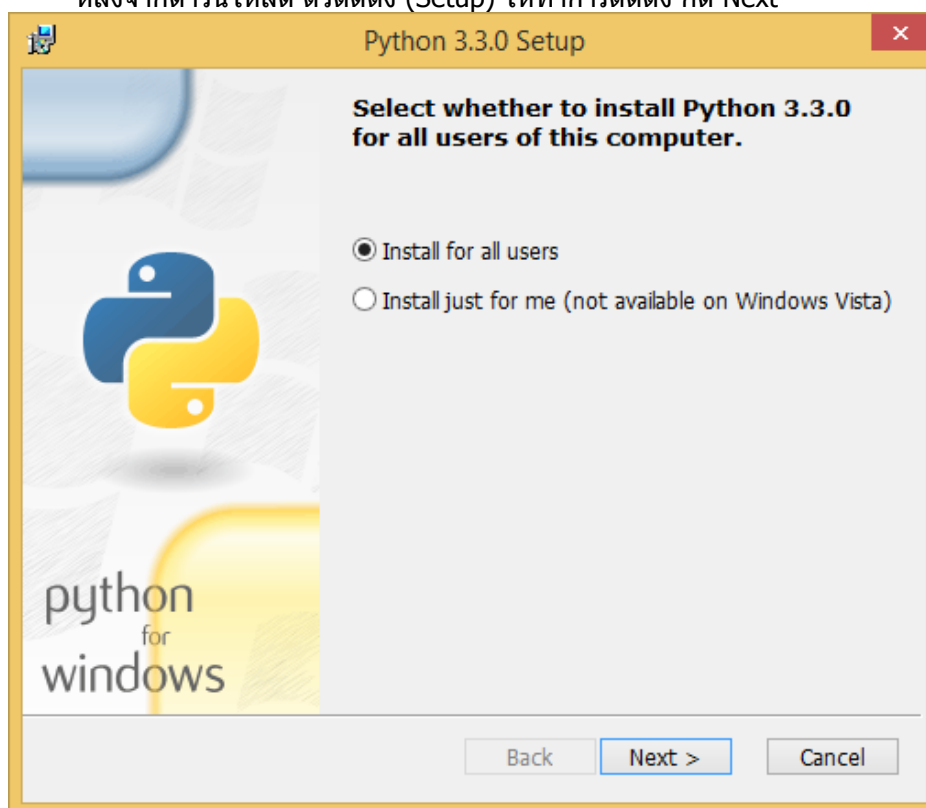
<http://pygame.info/downloads/>



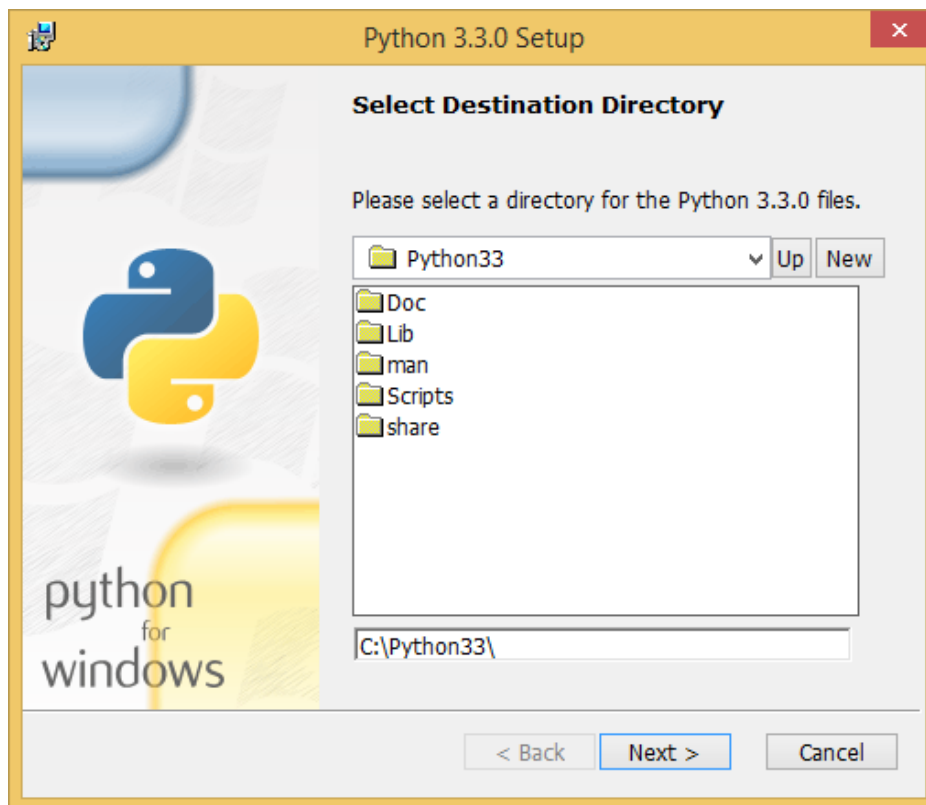
การติดตั้ง Installing

1 Install Python-3.3.0

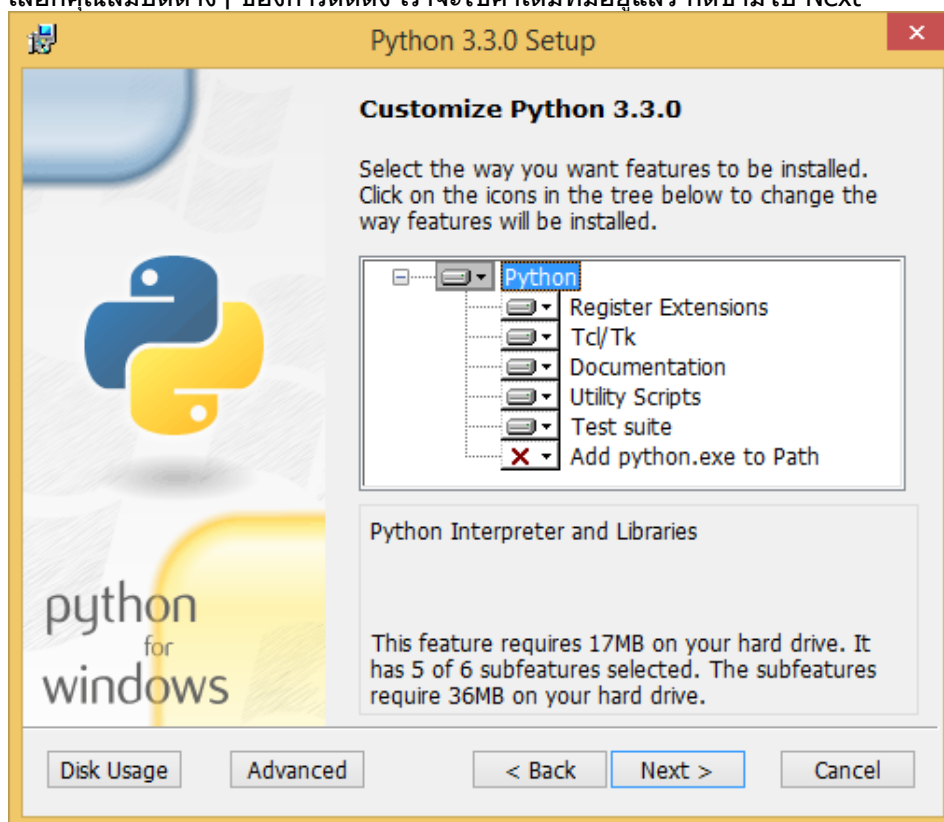
หลังจากดาวน์โหลด ตัวติดตั้ง (Setup) ให้ทำการติดตั้ง กด Next



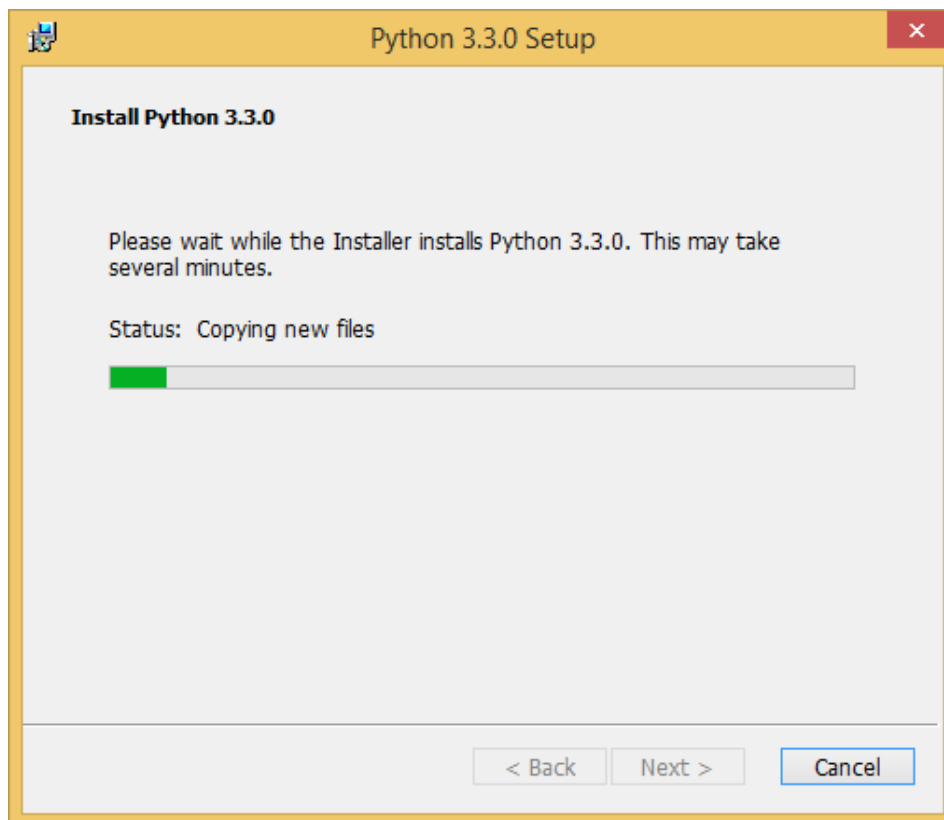
เลือกโฟลเดอร์ที่โปรแกรมจะติดตั้ง ถ้าเลือกเสร็จให้ กดข้ามไป Next



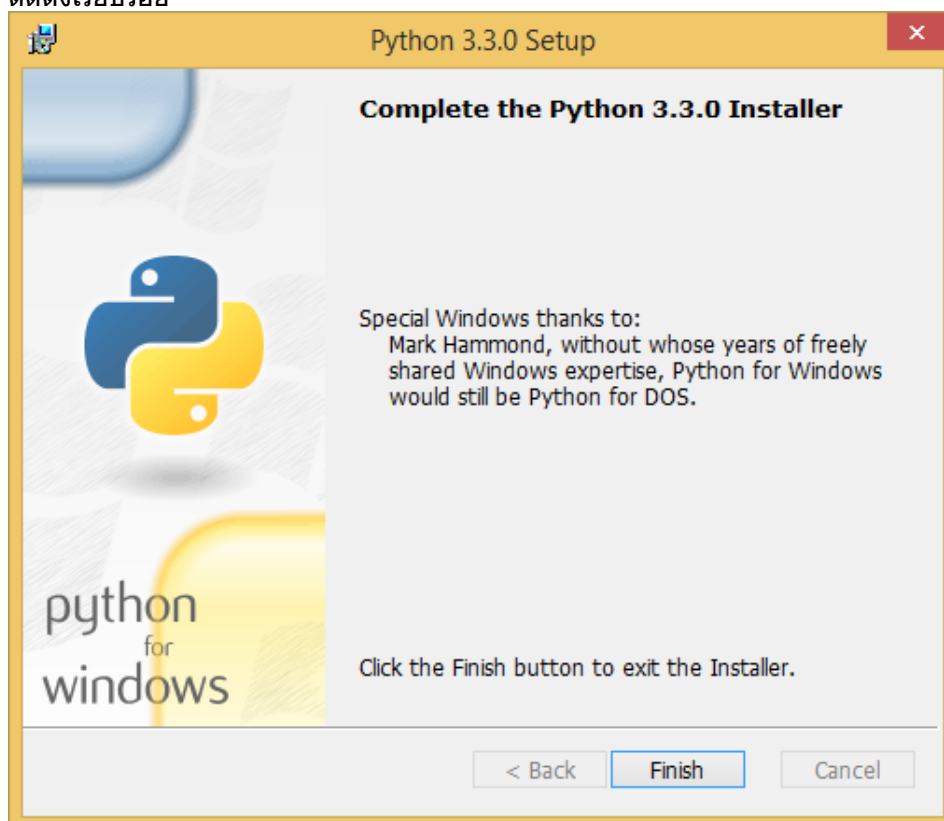
เลือกคุณสมบัติต่างๆ ของการติดตั้ง เราจะใช้ค่าเดิมที่มีอยู่แล้ว กดข้ามไป Next



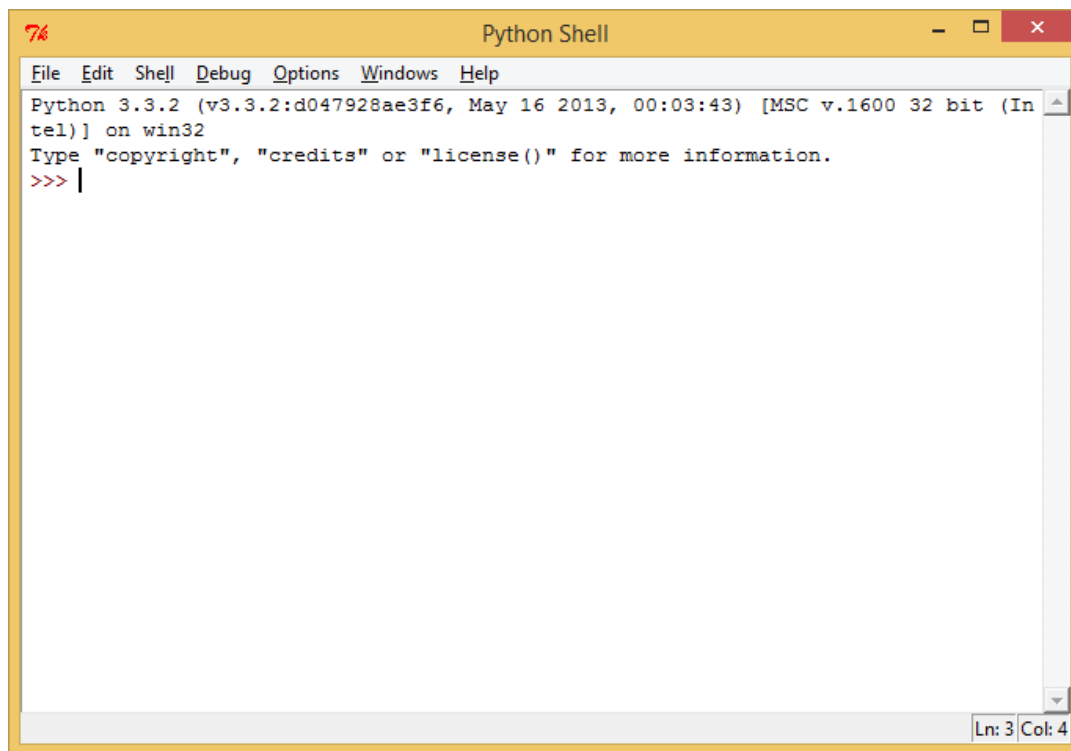
โปรแกรมกำลังติดตั้ง



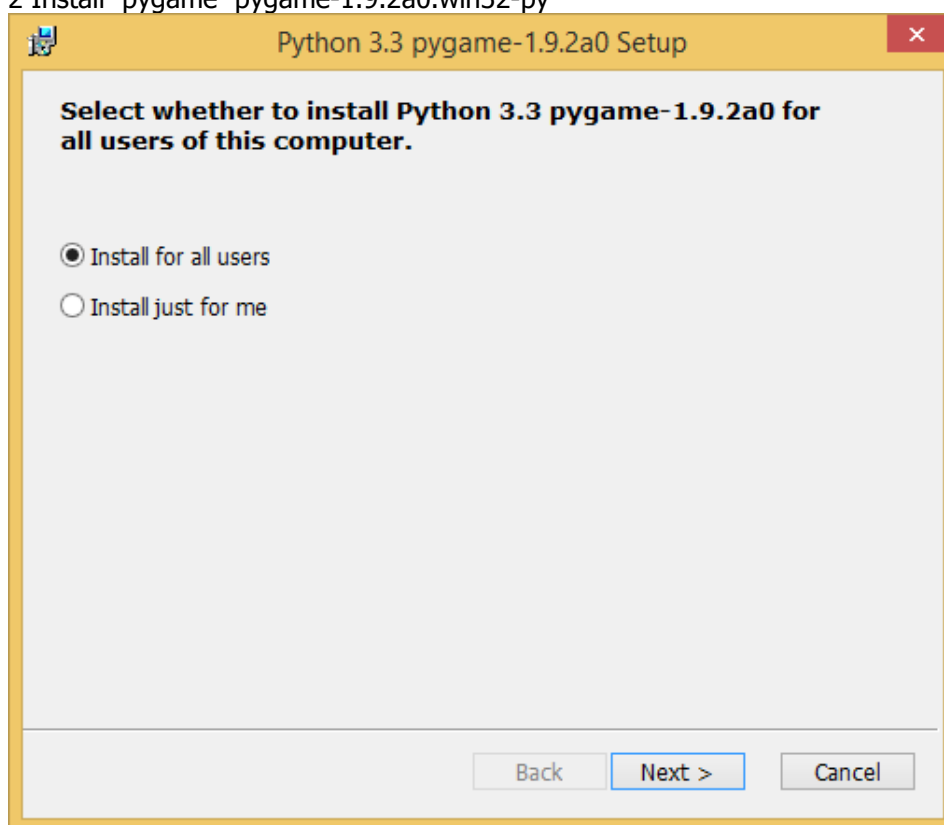
ติดตั้งเรียบร้อยแล้ว



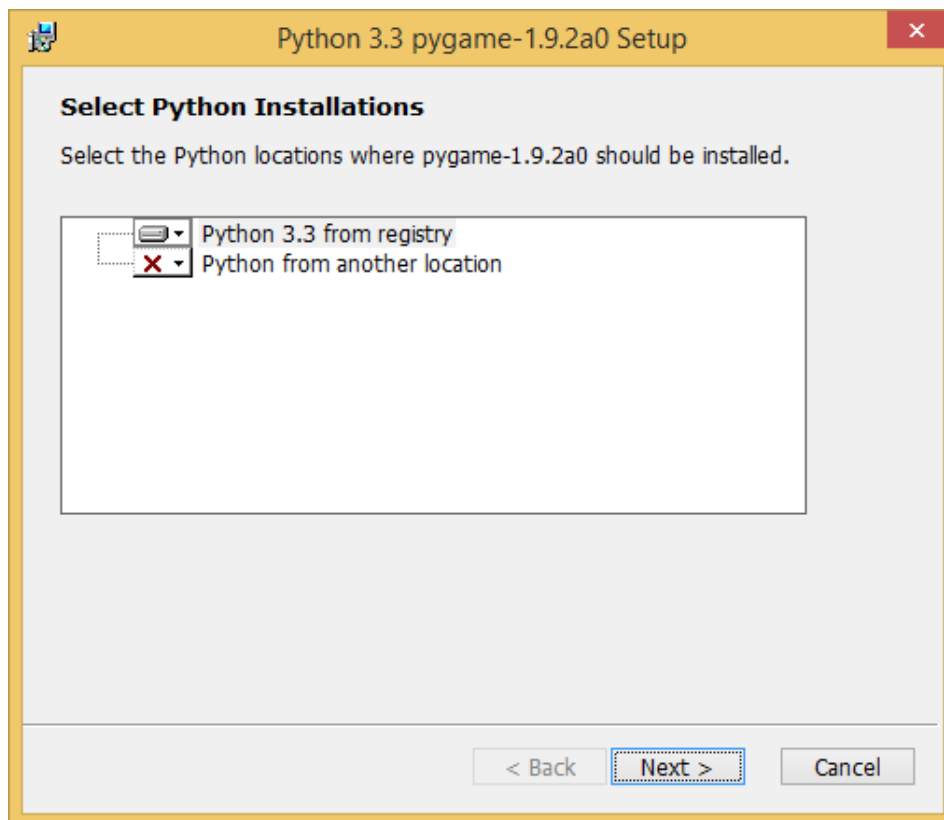
ทดสอบเปิด Python ให้ไปที่ window startup เลือก IDLE (Python GUI) จะปรากฏ หน้าต่างดังข้างล่างนี้



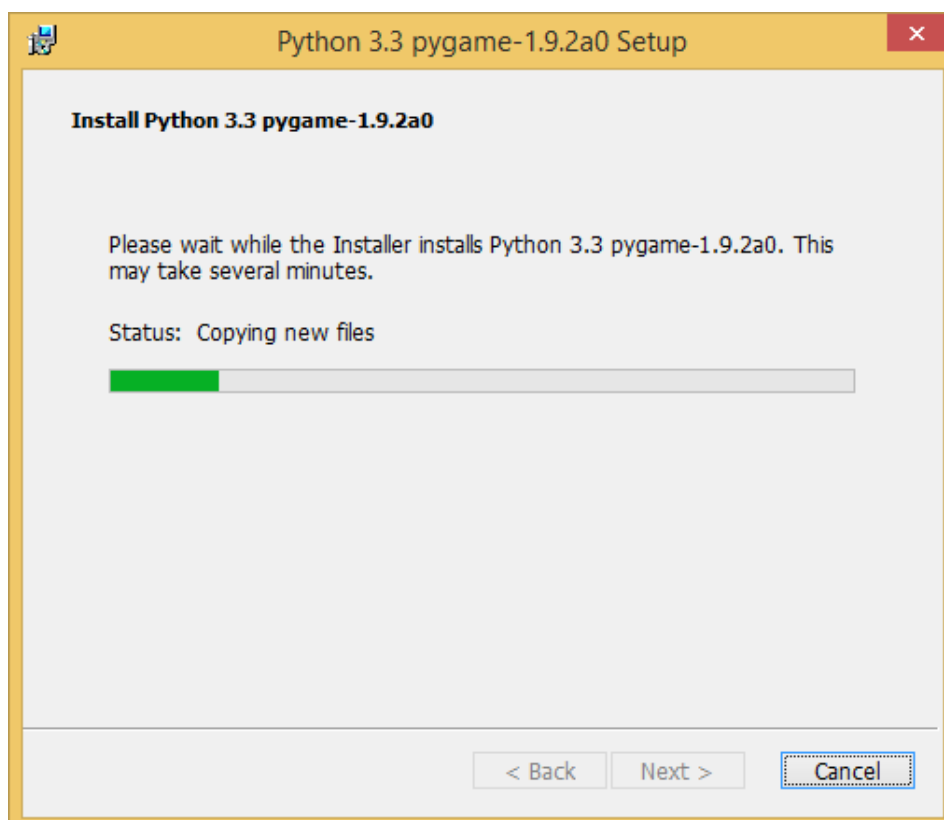
2 Install pygame pygame-1.9.2a0.win32-py



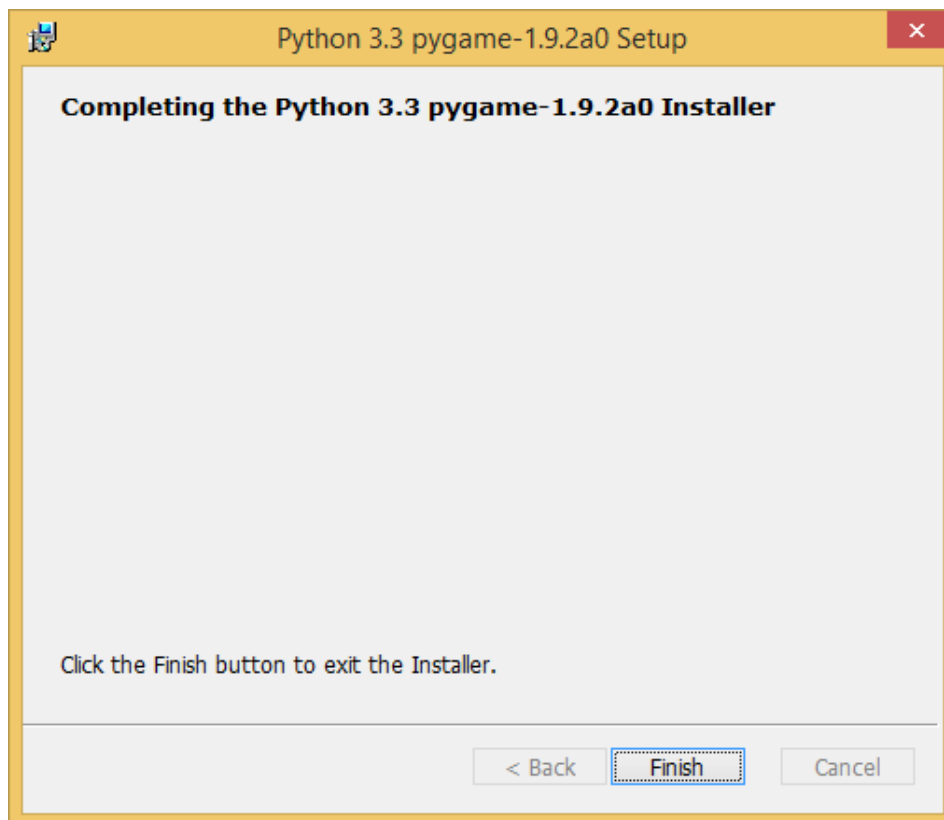
การทำงานร่วมกับ python ที่ติดตั้งก่อนหน้านี้ ใช้ค่าที่มีมาให้อยู่แล้วได้เลย



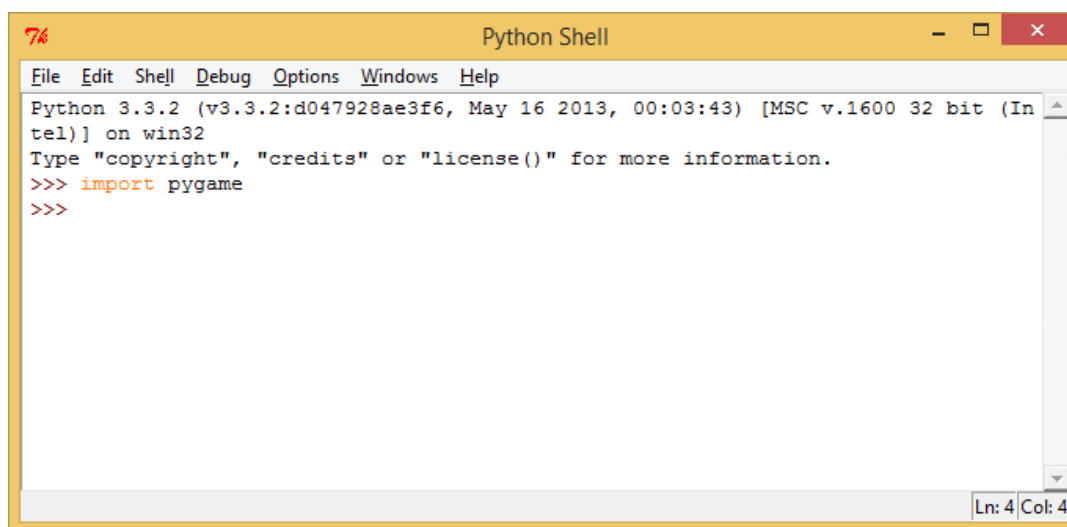
ติดตั้ง pygame



ติดตั้งสำเร็จ



ทดสอบ เรียกใช้โมดูล pygame ในบทต่อไปเราจะได้ทดลองมากกว่านี้



IDLE

IDLE (Python GUI)

IDLE ย่อมาจาก Integrated DeveLopment Environment เครื่องการพัฒนโปรแกรมแบบรวม ที่พัฒนามาจาก Tkinter ที่ใช้เขียน GUI ที่ย่อมาจาก Graphics User Interface ส่วนติดต่อผู้ใช้ที่เป็นกราฟฟิก มีหน้าต่าง มีแถบเมนู (Menu Bar) แถวเลื่อนด้านข้าง (Scrollbars) โปรแกรมที่ไม่ใช่ GUI จะแสดงเป็นหน้าต่างคอมมานด์ไลน์ (command line) ที่มีเพียงข้อความแสดงผล ซึ่งอาจจะเรียกว่า text-mode (โหมดข้อความ) หรือเรียกว่า โปรแกรมคอนโซล (console program)

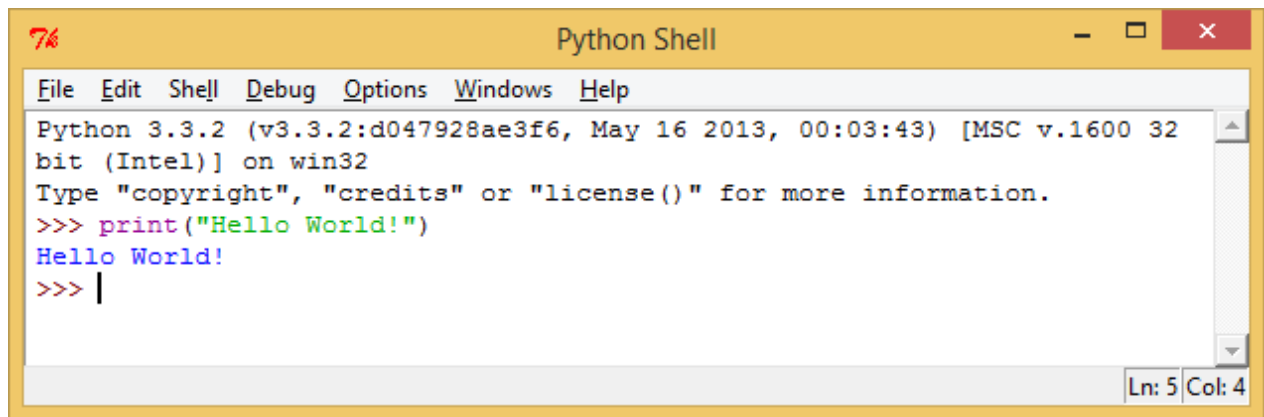
เราจะเห็นบรรทัดสุดท้ายของโปรแกรมมีตัวอักษร >>> เราเรียกว่า prompt เป็นบรรทัดที่คอมพิวเตอร์พร้อมทำงานและรอคำสั่งจากคุณ หน้าต่างทั้งหมดนี้เราอาจเรียกว่า Python Shell

ทดสอบสั่งพิมพ์ข้อความแสดงผล

พิมพ์ข้อความต่อไปนี้ ลงใน Python Shell ต่อจาก >>>

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

เมื่อพิมพ์เสร็จกด Enter
ได้ผลลัพธ์ดังนี้

A screenshot of a Python Shell window titled "Python Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the Python version and build information: "Python 3.3.2 (v3.3.2:d047928ae3f6, May 16 2013, 00:03:43) [MSC v.1600 32 bit (Intel)] on win32". It then prompts the user to type "copyright", "credits", or "license()" for more information. The user has entered the command ">>> print('Hello World!')", and the shell has output "Hello World!". The prompt ">>>|" is visible on the next line. The status bar at the bottom right shows "Ln: 5 Col: 4".

Hello World!

คือผลลัพธ์ที่ออกจากคำสั่งพิมพ์ ที่เราสั่งด้วย คำสั่ง `print()` ซึ่งจะพิมพ์ข้อความที่เราระบุส่งเข้าไป `Hello World!` ซึ่งจะอยู่ภายใต้เครื่องหมาย " " และอยู่ภายในวงเล็บ () อีกของคำสั่งที่

ทำไม IDLE จะต้องมีความหลากหลาย คำสั่ง พิมพ์ และตัวเลข ?

สีของตัวหนังสือบอกถึงชนิดและหน้าที่ของข้อความนั้นต่างกัน อาทิเช่น

`print` เป็นคำสั่ง (instructions)

`Hello World!` เป็นข้อความ (text)

`Hello World!` เป็นผลลัพธ์จากโปรแกรม (output)

`while` เป็นคำสั่งเฉพาะที่เราใช้ในภาษา Python เรียกว่า คีย์เวิร์ด (keyword) เป็นคำที่สงวนไม่ให้ใช้

ถ้าเราสั่งผิดพลาด เราจะสังเกตได้จากสี ตัวอย่างเช่น

```
>>> print("Hello World!")
Hello World!
>>> pront("Hello World!")
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    pront("Hello World!")
NameError: name 'pront' is not defined
```

ถ้าเราใช้คำสั่ง `print`

แต่พิมพ์ผิดเป็น `pront` ผลลัพธ์ที่ออกจากโปรแกรม จะแสดงผลว่าพิมพ์ผิด หรือที่เรียกว่า เออเรอร์ **Error** สังเกตว่าสีของตัวหนังสือจะผิด

Traceback (most recent call last):

ตรวจสอบกลับ สิ่งสุดท้ายที่เรียกใช้

File "<pyshell#1>", line 1, in <module>

ไฟล์ที่ผิดพลาดคือ pyshell ที่ 1 บรรทัดที่ 1 ในโมดูล

`pront("Hello World!")`

NameError: name 'pront' is not defined

ชื่อผิดพลาด ชื่อ `pront` ไม่เคยประกาศมาก่อน

รูปแบบของการผิดพลาดคือ NameError มีความหมายว่าชื่อผิดพลาด

การโต้ตอบกับ Python (Interacting with Python)

การโต้ตอบ (Interaction) ในที่นี้ คือ เมื่อเราพิมพ์คำสั่ง (command or instruction) ไป Python ทำการคำนวณ (Execution) ให้ทันที

ทดสอบบวกเลขและลบเลข

`10 + 5`

`10 - 5`


```
>>> 10 + 5
15
>>> 10 - 5
5
>>>
```

ทดสอบการคูณ

สัญลักษณ์การคูณในคณิตศาสตร์ คือ x สำหรับนักคอมพิวเตอร์แล้วเราใช้ดอกจัน (star) * เราอาจสังเกตจากคีย์บอร์ด (keyboard) ของเราก็ใช้ดอกจันสำหรับการคูณเช่นกัน

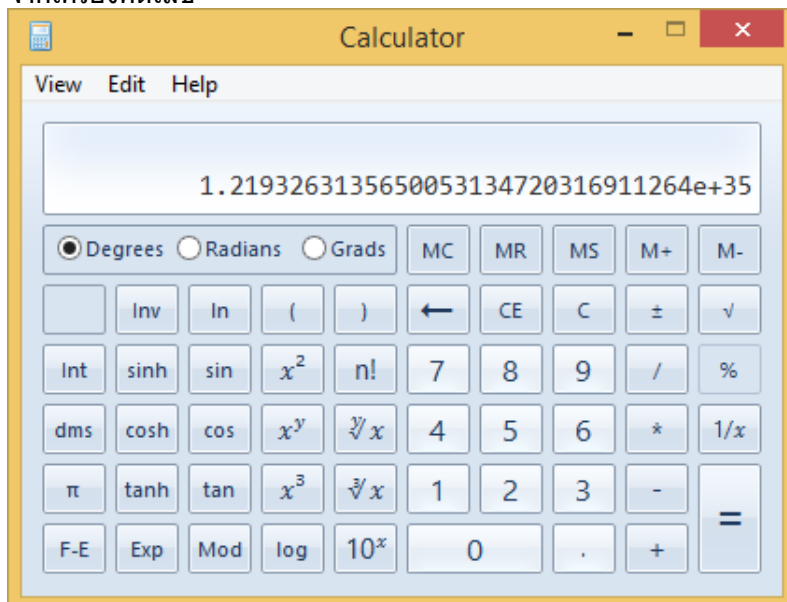
```
>>> 10 * 5
50
>>>
```

ถึงแม้เครื่องคิดเลขในมือถือหรือคอมพิวเตอร์จะทำได้เหมือนกัน แต่สำหรับการคำนวณตัวเลขเยอะๆ จะเป็นอย่างไร

จาก Python

```
>>> 123456789123456789 * 987654321987654321
121932631356500531347203169112635269
>>>
```

จากเครื่องคิดเลข



จะเห็นได้ว่าการหาค่าด้วยโปรแกรม Python และเครื่องคิดเลขความละเอียดของค่าที่ได้รับต่างกัน เนื่องจากคอมพิวเตอร์มีความสามารถในการคำนวณที่มากกว่า

ทดสอบการหาร

สัญลักษณ์ที่ใช้กับการหารคือ สแลช /

```
>>> 6 / 2
3.0
```

โดยรายละเอียดในการเขียนโปรแกรมคำนวณทางคณิตศาสตร์จะมีมากกว่าในบทต่อไป ทั้งการสร้างตัวแปร การสร้างสมการ การใช้ฟังก์ชัน นอกจากคำนวณทางคณิตศาสตร์แล้วยังทำได้มากกว่านี้ อาทิเช่น การใช้เครื่องหมายกับข้อความ อาทิเช่น

```
>>> "cat" + "dog"
'catdog'
```

"cat" และ "dog" มีชนิดเป็นข้อความ เราสามารถใช้เครื่องหมายบวกเพื่อรวมเป็นข้อความเดียวกัน ซึ่งจะกล่าวรายละเอียดภายหลังอีกที นอกจากนี้ใช้ เครื่องหมายคูณข้อความกับตัวเลขได้

```
>>> "hello " * 20
'hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello '

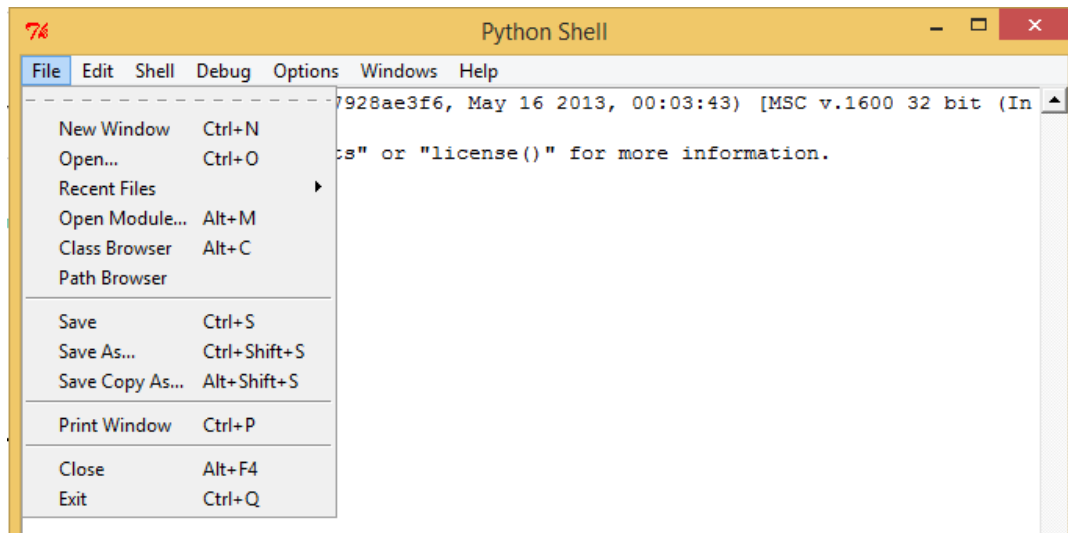
```

ผลที่ได้มีข้อความ "hello" 20 คำ

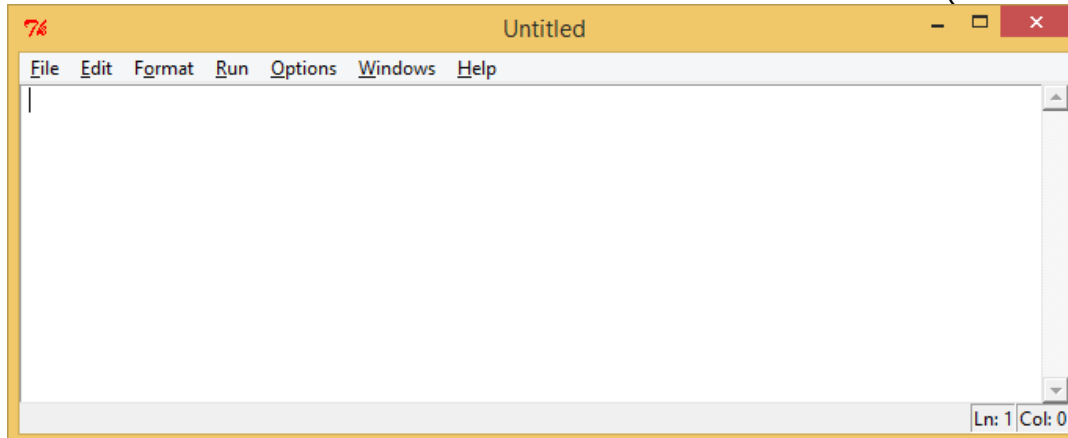
เริ่มการเขียนโปรแกรม (Time to program)

เราได้ทดสอบคำสั่งลง DILE แล้วดูผลลัพธ์ออกมา ต่อไปนี้จะเป็นการเขียนโปรแกรมแรกของเรา โปรแกรมคือกลุ่มของคำสั่งเพื่อให้คอมพิวเตอร์ทำงานที่ต้องการ ดังนั้นวิธีการเขียนโปรแกรม คือ การเขียนคำสั่งหลายๆ คำสั่งเก็บไว้บนไฟล์ แล้วรัน (run) เพื่อดูผลลัพธ์ที่เราสั่ง การเขียนโปรแกรมภาษา Python จะต้องสร้างไฟล์ที่มีนามสกุล .py ซึ่งเราสามารถสร้างได้โดยตรงจาก DILE หรืออาจจะเขียนบนเครื่องมือตัวอื่น อาทิเช่น Notepad หรือ เท็กซ์อีดิเตอร์ TextEditor

การสร้างไฟล์โปรแกรมบน DILE ให้เราไปที่ เมนูหลัก (Main Menu) กดปุ่ม File (ไฟล์) จะได้รับด้านล่างนี้



แล้วกดที่ New Window จะได้หน้าต่างเปล่าขึ้นมาอีกหนึ่งอัน ที่มีหัวเรื่องเป็น Untitled (ไม่มีชื่อเรื่อง) ดังรูป

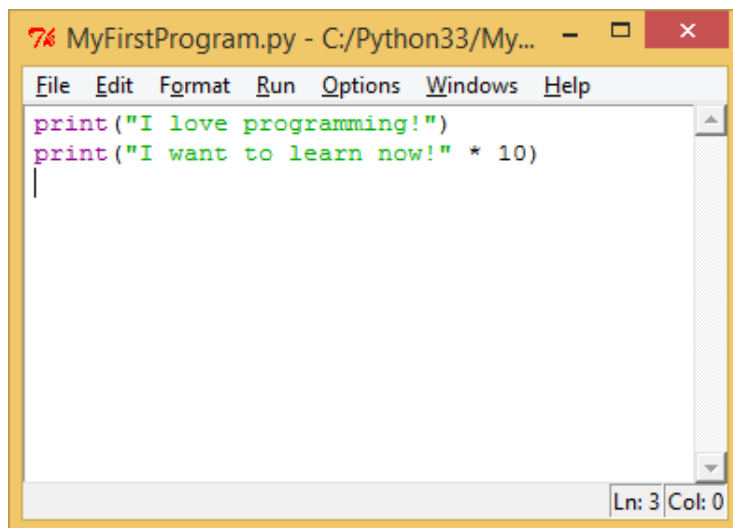


จากขั้นตอนการสร้างไฟล์โปรแกรมใหม่ เราขอแทนด้วยสัญลักษณ์ลำดับการทำงานให้อยู่ในรูปแบบง่ายๆ ดังนี้ **File > New Window** กดปุ่ม File ที่เมนู และกดปุ่ม New Window ที่อยู่ภายใต้เมนู File ตามลำดับ (เราใช้สัญลักษณ์ > หมายถึงภายใต้)

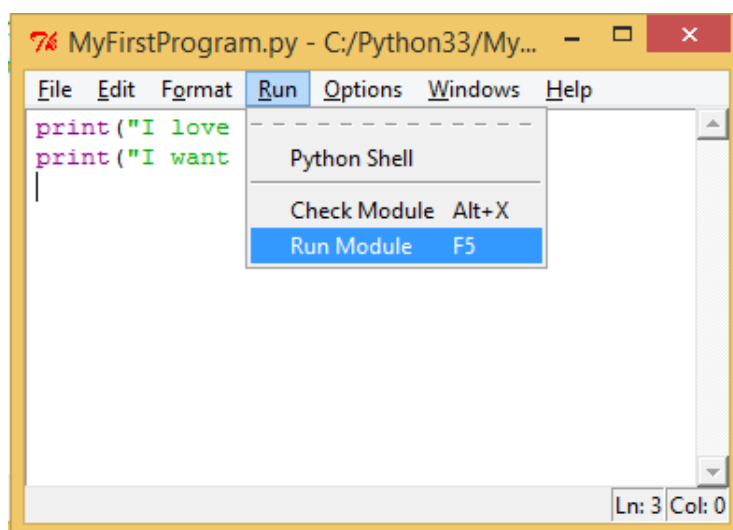
เริ่มเขียนโปรแกรมแรกของเรา เพื่อแสดงผลข้อความ ลงในไฟล์ที่เราเพิ่งสร้างขึ้นใหม่ โดยเขียนโค้ดด้านล่างนี้

```
print("I love programming!")
print("I want to learn now!" * 10)
```

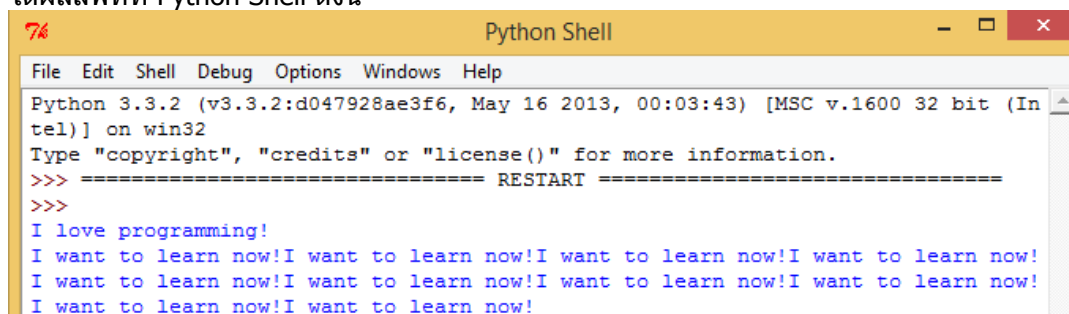
ทำการบันทึก (Save) ตามลำดับนี้ **File > Save** เลือกโฟลเดอร์ที่เราต้องการจะบันทึก แล้วตั้งชื่อไฟล์ของเราว่า MyfirstProgram (โปรแกรมแรกของฉัน) จะได้ดังรูปนี้



รันดูผลลัพธ์ด้วยวิธีการนี้ **Run > Run Module**



ได้ผลลัพธ์ที่ Python Shell ดังนี้

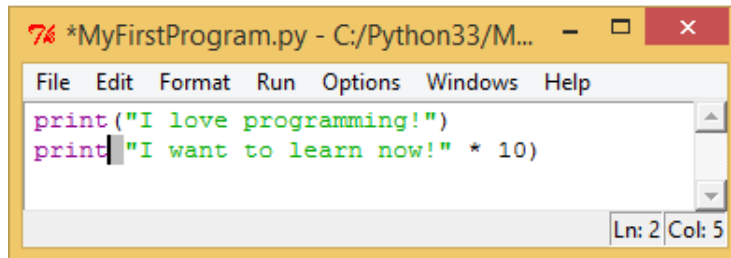


การเขียนโปรแกรมมีความผิดพลาดขึ้น (Error)

Syntax - ซินแท็ก หมายถึง ข้อบังคับการเขียนโปรแกรม ขึ้นอยู่กับแต่ละภาษาคอมพิวเตอร์นั้น

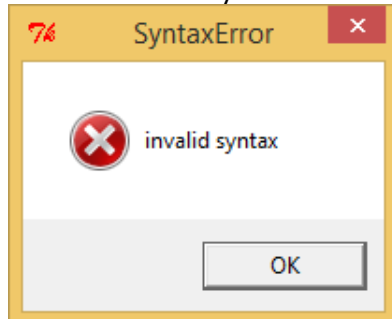
Error มีอยู่สองรูปแบบ คือ

1. ความผิดพลาดจากเขียนคำสั่งผิดรูปแบบ (Syntax Error) ตัวอย่างเช่น โปรแกรมที่เราเขียนบรรทัดที่สอง ลืมวงเล็บเปิด ของคำสั่ง print



```
File Edit Format Run Options Windows Help
print('I love programming!')
print('I want to learn now!' * 10)
Ln: 2 Col: 5
```

จะมีการแจ้งเตือน Syntax ผิดพลาด ดังนี้

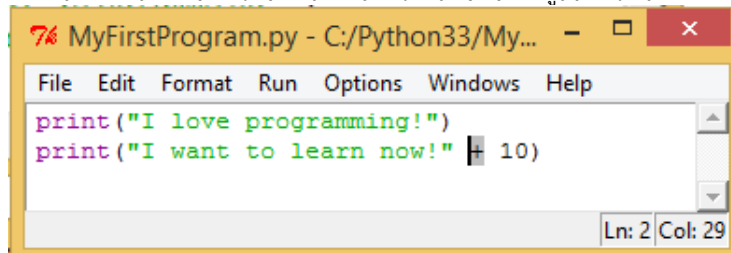


Invalid syntax (การเขียนโปรแกรมผิดพลาด)

เมื่อเรากด OK เคอร์เซอร์ (Cursor) จะบรรทัดที่ผิดพลาด หรือเราสามารถทดสอบโดยตรงกับ Python Shell จะเกิด Error ชัดเจน

```
>>> print 'I want to learn now!' * 10)
SyntaxError: invalid syntax
```

2. ความผิดพลาดระหว่างรันโปรแกรม (Runtime Error) Syntax ถูกต้องแต่เมื่อคอมพิวเตอร์รันโปรแกรมแล้วจะผิดพลาด ตัวอย่างเช่น เมื่อเราเปลี่ยนเครื่องหมายคูณเป็นบวก



```
File Edit Format Run Options Windows Help
print('I love programming!')
print('I want to learn now!' + 10)
Ln: 2 Col: 29
```

ครั้งนี้การรันจะไม่มีแจ้งเตือน SyntaxError แต่จะมี Error แสดงผลที่ Python Shell ดังนี้

```
>>>
I love programming!
Traceback (most recent call last):
  File "C:/Python33/MyFirstProgram.py", line 2, in <module>
    print('I want to learn now!' + 10)
TypeError: Can't convert 'int' object to str implicitly
```

นี้หมายถึง Runtime error

ทำไม "I want to learn now!" + 10 จึง Error ขณะที่
"I want to learn now!" * 10 จึงได้

ทดสอบเพิ่มเติม

```
>>> 2 + 3
5
>>> "2" + "3"
'23'
>>> "2" * 3
'222'
>>> 2 * 3
6
>>> "2" + 3
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    "2" + 3
TypeError: Can't convert 'int' object to str implicitly
```

ข้อมูลที่เราทดสอบมีสองชนิด

- ตัวเลข (Number) 2 และ 3
- ข้อความ (Text) "2" และ "3"

การบวกเราจำเป็นต้องบวกสิ่งของที่มีชนิดเดียว ข้อความกับข้อความ ตัวเลขกับตัวเลข เราไม่สามารถรวมสองสิ่งที่แตกต่างกันได้ Python ไม่อนุญาต (ถ้ารวมกัน ได้ไม่รู้จะรวมกันอย่างไร) ดังนั้น

```
"2" + 3
```

เราจะเห็นข้อความ Error บรรทัดสุดท้ายว่า

TypeError: Can't convert 'int' object to str implicitly

TypeError = Type Error (ชนิดผิดพลาด)

Can't convert 'int' object to str implicitly ไม่สามารถเปลี่ยนข้อมูล (object วัตถุ) 'int' (จำนวนเต็ม) เป็นข้อความ (str เป็นข้อมูลชนิดหนึ่ง ย่อมาจาก สตริง - string ภาษาทางโปรแกรมมิ่งหมายถึง ชนิดข้อความ)

```
>>> "2 + 3
SyntaxError: EOL while scanning string literal
```

ส่วนนี้ Error เพราะอะไร

ตัวอย่างโปรแกรมเกมส์แรกของเรา (Our first game Program)

เกมส์ทายเลข โดยคอมพิวเตอร์จะสุ่มคำตอบจากเลข 1-99 ไว้ก่อน จากนั้นให้เราทายเลขนั้น ให้ถูกต้อง โดยโปรแกรมจะช่วยให้เราว่า เลขที่ทายมีค่ามากหรือน้อยกว่าเลขคำตอบ จำนวนครั้งมีจำกัด

เกมส์มีขั้นตอนดังนี้

- The secret number is randomly picked by the program.
โปรแกรมสุ่มตัวเลขคำตอบเก็บไว้
- The user inputs guesses
ผู้ใช้ทายตัวเลข
- The program checks each guess against the secret number: is it higher or lower?
โปรแกรมตรวจสอบตัวเลขที่ผู้ใช้ทายและคำตอบ ว่าถูกหรือไม่ ถ้าไม่ มีค่ามากหรือน้อยกว่า
- The user keeps typing until he guesses the number or runs out of turns.
ผู้ใช้ทายเลขจนครบจำนวนครั้งหรือทายจนถูกต้อง

สามารถนำไปเขียนโปรแกรม ได้ดังต่อไปนี้

```

import random
secret = random.randint(1, 99)
guess = 0
tries = 0
print("AHOY! I'm the Dread Pirate ROberts, and I have a secret!")
print("It is a number from 1 to 99. I'll give you 7 tries")
while guess != secret and tries < 7:
    guess = int(input("What's yer guess? "))
    if guess < secret:
        print("Too low, ye scurvy dog!")
    elif guess > secret:
        print("Too high, landlubber!")

    tries = tries + 1

if guess == secret:
    print("Avast! ye got it! Found my secret, ye did!")
else:
    print("No more guesses! Better luck next time, matey!")
    print("The secret number was", secret)

```

เรายังไม่จำเป็นต้องรู้ความหมาย โปรแกรมนี้ต้องการทดสอบการพิมพ์ให้ถูกต้อง ถ้าพิมพ์ผิดพลาดโปรแกรมจะเตือน
 ยิงใจ แล้วเราจะแก้ไขอย่างไร ส่วนความหมายคำอื่นๆ เราจะเรียนในบทหลัง

แต่เราจำเป็นต้องเขียนให้ถูกต้อง ทั้งการกดแถบหนึ่งระดับบางบรรทัด เช่นตัวอย่างโค้ด

```

while guess != secret and tries < 7:
    guess = int(input("What's yer guess? "))

```

แต่ถ้าเราใส่โคลอล (:) อย่างถูกต้องเมื่อ Enter จะแถบ (tab) ให้อัตโนมัติ

ผลรันของโปรแกรมเป็นดังนี้

```

>>>
AHOY! I'm the Dread Pirate ROberts, and I have a secret!
It is a number from 1 to 99. I'll give you 6 tries
What's yer guess? 50
Too high, landlubber!
What's yer guess? 25
Too low, ye scurvy dog!
What's yer guess? 37
Too high, landlubber!
What's yer guess? 31
Too high, landlubber!
What's yer guess? 28
Too high, landlubber!
What's yer guess? 27
Avast! ye got it! Found my secret, ye did!

```

ทบทวน

- การติดตั้ง Python และ pygame
- การใช้งาน IDLE , Python Shell
- การโต้ตอบกับ Python Shell
- การโปรแกรมเบื้องต้น
- การคำนวณด้วยคณิตศาสตร์เบื้องต้น
- การเขียนโปรแกรมเป็นไฟล์ด้วย IDLE
- การรันโปรแกรม
- ความผิดพลาด Error ของการเขียนโปรแกรม
- การเขียนโปรแกรมเกมส์

คำถาม

- เราเปิด IDLE อย่างไร
- คำสั่ง `print` ทำอะไร

- เครื่องหมายคุณลักษณะอย่างไร

จงเขียนโปรแกรมต่อไปนี้

1. จงหาคำนวณจำนวนนาฬิกาในหนึ่งสัปดาห์
2. จงเขียนโปรแกรมแนะนำตัวเองต่อไปนี้ ให้พิมพ์ชื่อ วันเกิด และสีที่ชอบ ดังผลลัพธ์ด้านล่างนี้
`My name is Sarayut Gonwirat.`
`I was born November 9, 1984.`
`My favorite color is blue`