ภาษาไพทอน (Python language) และการเขียนโปรแกรมเบื้องต้น

1. แนะนำเบื้องต้นกับภาษาไพทอน (Introduction)
2. ตัวแปร (Variable)
3. การกำหนดค่าให้ตัวแปร (Assignment statement)
4. ชนิดข้อมูล (Data type)
5. ตัวดำเนินการ (Operator)
6. คำสั่งรับและแสดงผล (Input and output statement)
7. ปฏิบัติการ

1. Introduction

1.1 ประวัติความเป็นมา

1.2 คุณลักษณะของภาษาไพทอน

1.3 First program

1.1 ประวัติความเป็นมา

- **ในปี ค.ศ. 1989 ภาษาไพทอนถูกพัฒนาโดย** Guido van Rossum **นักวิจัย แห่งสถาบัน วิจัยแห่งชาติทางด้านคณิตศาสตร์และวิทยาการคอมพิวเตอร์ เมืองอัมสเตอร์ดัม ประเทศเนเธอร์แลนด์**

- **ภาษาไพทอนเป็นภาษาที่นำลักษณะที่ดีของภาษาที่มีอยู่เดิม (**ABC, Modula-3, C, C++, Algol-68, SmallTalk and Unix shell and other scripting languages) **และเพิ่มคุณลักษณะที่ดีเช่น คลาสและอื่น ๆ รวมถึงมี** interface **ให้เขียนโปรแกรมได้สะดวก**

1.2 คุณลักษณะของภาษาไพทอน

ภาษาไพทอนเป็นภาษาระดับสูง มีคุณลักษณะ ดังนี้

1.เป็นภาษาที่จัดอยู่ในกลุ่ม Interpreter

คือแปลแล้วทำงานทีละคำสั่ง มีการประมวลผลทันที (process at runtime)

2.มีลักษณะ interactive คือ เราสามารถพิมพ์คำสั่ง ทำงานในลักษณะ interact คือโต้ตอบได้

3.เป็นภาษาที่ได้รับความนิยม เรียนรู้ได้ง่าย เหมาะกับผู้เริ่มต้นเขียนโปรแกรม

1.3 first program

```
1  # first program
2  # you can do it
3  print(“Welcome to Python!”)   #  print line of text
```

อธิบาย

บรรทัด 1 comment ภาษาไพทอน สัญลักษณ์ # คือ comment

บรรทัด 2 comment comment ได้ทีละ 1 บรรทัด
จะใช้เครื่องหมาย # นำหน้าข้อความที่ต้องการ comment ไปจนจบบรรทัดนั้น

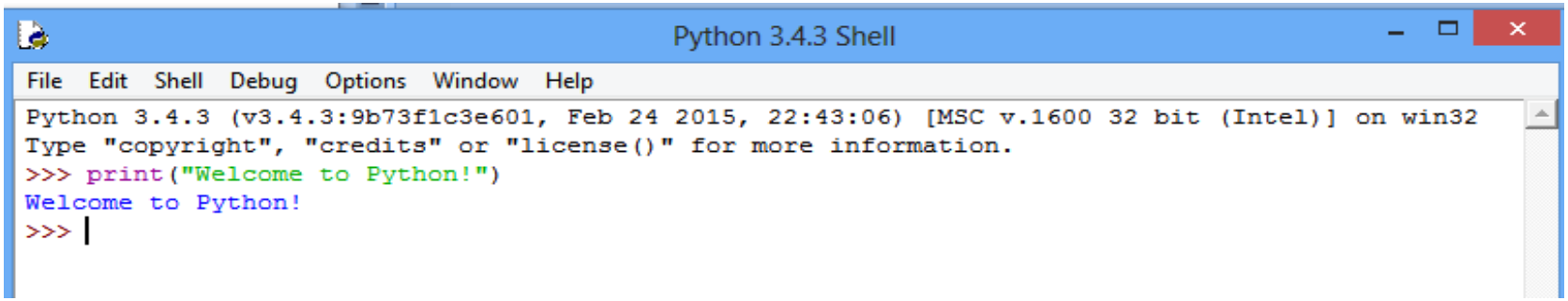
บรรทัด 3 คำสั่ง print () คือฟังก์ชันเพื่อแสดงผลข้อความที่จอภาพ ข้อความอยู่ใน
เครื่องหมายคำพูด ” ”

ผลลัพธ์ที่ได้จากโปรแกรมนี้คือข้อความ Welcome to Python! ปรากฏที่จอภาพ

การทำงานของภาษา python

ภาษาไพทอน execute ได้ 2 mode คือ

(1) INTERACTIVE MODE PROGRAMMING: เป็น mode ที่เราพิมพ์คำสั่ง ภาษาไพทอนจะแปลและทำงานทันที เช่น



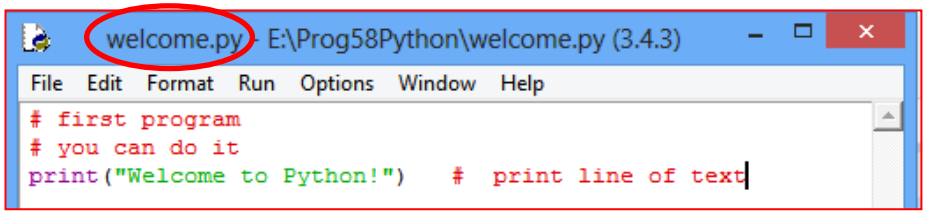
(2) SCRIPT MODE PROGRAMMING:

พิมพ์คำสั่งหรือโปรแกรมที่ editor หรือ IDLE ของไพทอน (write a simple Python program in a script) จากนั้น save file และกำหนด file type เป็น .py

เมื่อรันโปรแกรมจะได้ผลลัพธ์ เช่น

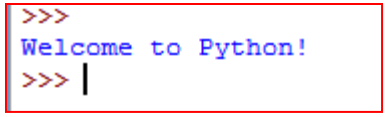
source code in a welcome.py file

print "Welcome to Python!"



This will produce the following result:

Welcome to Python!



2. ตัวแปร (variable)

ตัวแปรจะจองเนื้อที่ในหน่วยความจำ ขนาดหรือเนื้อที่ที่ใช้ ขึ้นกับชนิดข้อมูล

ข้อกำหนดในการตั้งชื่อตัวแปร

1. อักษรตัวแรกต้องเป็นตัวอักษร (A-Z หรือ a-z) หรือ เครื่องหมาย _ (Underscore) เท่านั้น
2. อักษรตัวอื่นๆ ต้องเป็นตัวอักษร (A-Z หรือ a-z) หรือ ตัวเลข 0-9 หรือเครื่องหมาย _ เท่านั้น
3. ห้ามตั้งชื่อตัวแปรซ้ำกับคำสงวน (Reserved Word)
4. ห้ามมีช่องว่างภายในชื่อ
5. ตัวอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ถือว่าต่างกัน

   เช่น NUM , Num และ num เป็นตัวแปรคนละตัวกัน

Reserved words

These reserved words may not be used as constant or variable or any other identifier names. All the Python keywords contain lowercase letters only.

and	**exec**	**not**
assert	**finally**	**or**
break	**for**	**pass**
class	**from**	**print**
continue	**global**	**raise**
def	**if**	**return**
del	**import**	**try**
elif	**in**	**while**
else	**is**	**with**
except	**lambda**	**yield**

ตัวอย่างการตั้งชื่อตัวแปร

123MyID	**ผิด ขึ้นต้นด้วยตัวเลข**
_ThinkBig	**ถูกต้อง**
mylife@CMU	**ผิด ห้ามใช้เครื่องหมาย @**
Pin number	**ผิด ห้ามเว้นวรรค**
Dorm_number	**ถูกต้อง**

3. การกำหนดค่าให้ตัวแปร (assignment statement)

ตัวแปรในภาษาไพทอนเราไม่ต้องประกาศชนิดข้อมูลให้ตัวแปร ก่อนทำงานการประกาศจะเกิดขึ้นโดยอัตโนมัติเมื่อเรากำหนดค่าให้ตัวแปร โดยใช้เครื่องหมาย =

เช่น

```
score = 75
gpa = 2.85
```

หมายความว่า ตัวแปร score มีเป็นจำนวนเต็ม มีค่า 75

ตัวแปร gpa มีเป็นจำนวนจริง มีค่า 2.85

ตัวอย่างการกำหนดค่า

นอกจากนี้ เราสามารถกำหนดค่าในลักษณะ multiple assignment ได้ ดัง

ตัวอย่างที่ 1

```
a = b = c = 1
```

คำอธิบาย

ตัวแปร a b c เป็นจำนวนเต็มมีค่า 1

ตัวอย่างที่ 2

```
a, b, c = 1, 2, "john"
```

คำอธิบาย

ตัวแปร a เก็บข้อมูลจำนวนเต็มที่มีค่าเท่ากับ 1

bเก็บข้อมูลจำนวนเต็มที่มีค่าเท่ากับ 2

c เก็บข้อมูลที่เป็นข้อความ john

4.ชนิดข้อมูล (Data type)

ชนิดข้อมูล สำหรับการเขียนโปรแกรมเบื้องต้นที่ควรรู้จักมีดังนี้

(1) int	**จำนวนเต็ม**
(2) float	**จำนวนจริงหรือทศนิยม**
(3) boolean	**บูลีน : True, False**
(4) string	**สตริงหรือข้อความ**
(5) list	**ลิสต์**

Data type : int float

(1) int เลขจำนวนเต็ม เช่น 2 , 50 , 1009

(2) float เลขจำนวนจริงหรือ ทศนิยม เช่น 15.20 ,-21.9

Data type : (3) บูลีน

บูลีน (Boolean) คือค่า จริง , เท็จ ในภาษาไพทอน ใช้คำว่า True False ในการเปรียบเทียบ จะได้ผลลัพธ์เป็นบูลีน เช่น

การเปรียบเทียบ 4 > 1 เป็นจริง ผลที่ได้คือบูลีน True

6 > 7 เป็นเท็จ ผลคือได้คือบูลีน False

เมื่อเขียนโปรแกรม เราจะใช้ผลการเปรียบเทียบ เพื่อตรวจสอบเงื่อนไขในการทำงาน

Data type : (4) string

สตริงหรือข้อความคือตัวอักษรที่เรียงต่อกันในเครื่องหมายคำพูด ภาษาไพทอนใช้ได้ทั้ง

'	(single)
"	(double)
''' or """	(triple)

ตัวอย่างสตริง

```
text1 = 'word'
text2 = "This is a sentence."
```

นอกจากนี้ยังสามารถใช้ triple (''' or """) เพื่อกำหนดข้อความที่มีหลายบรรทัดได้ เช่น

```
text3 = """This is a paragraph. It is
made up of multiple lines and sentences."""
```

นอกจากนี้ในการทำงานกับสตริงเราสามารถใช้ ตัวดำเนินการ + * (มีตัวอย่าง)

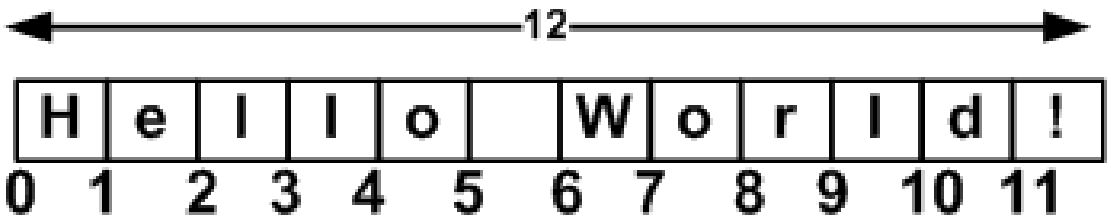
- เครื่องหมาย + นำสตริงหรือข้อความมาต่อกัน
- เครื่องหมาย * ทำซ้ำ

Data type : (4) string

- ซับสตริง (substring) หรือข้อความย่อยในสตริง
- เราใช้เครื่องหมาย [ ] หรือ [:] เพื่อทำงานกับข้อความย่อย
- ตำแหน่งการจัดเก็บอักษรแต่ละตัว เริ่มที่ 0 จบที่ end-1

การจัดเก็บ สตริง อธิบายได้ดังนี้

- ตัวอย่างสตริงเก็บคำว่า Hello World!



จากตัวอย่างข้อความ **Hello world!** มีทั้งหมด **12** ตัวอักษร

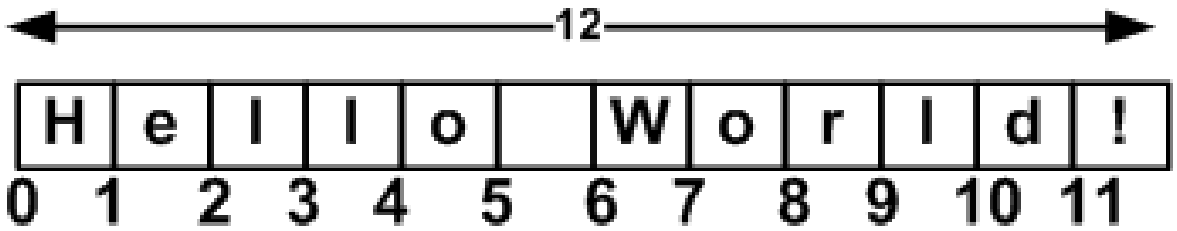
ตำแหน่งการจัดเก็บตัวอักษร เริ่มที่ **0** เก็บตัวอักษร **H**

ตัวสุดท้ายอยู่ที่ตำแหน่ง **11** คือ**end-1 = 12-1** เก็บตัวอักษร **!**

ตัวอย่างการทำงานกับสตริง

```
name = 'Hello World!'
print (name)        # Prints complete string
print (name[0])     # Prints first character of the string
print (name[2:5]) # Prints characters starting from 3rd to 5th (end-1 คือ 5-1 = 4)
print (name[2:])   # Prints string starting from 3rd character
print (name * 2)              # Prints string two times
print (name + "TEST ")     # Prints concatenated string
```

This will produce the following result:



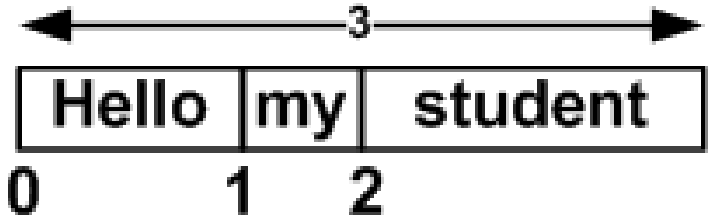

```
Hello World!
H
llo
llo World!
Hello World!Hello World!
Hello World!TEST
```

Data type : (5) ลิสต์ (list)

- list เป็นชนิดข้อมูล มี item หรือสมาชิก อยู่ใน list
- item มีชนิดข้อมูลที่ต่างกันได้ (compound data type)
- item อยู่ในเครื่องหมาย [ ]
- แต่ละ item แยกกันด้วยเครื่องหมาย , (comma)
- ค่าที่เก็บใน list เรียกมาใช้งานด้วยเครื่องหมาย [ ] [:]
- item หรือสมาชิก เริ่มที่ 0 จบที่ end-1 อธิบายดังนี้

item หรือสมาชิก ใน list เรียงต่อเนื่องกัน คือ ตัวแรกเริ่มที่ 0 ตัวสุดท้ายอยู่ที่ end-1

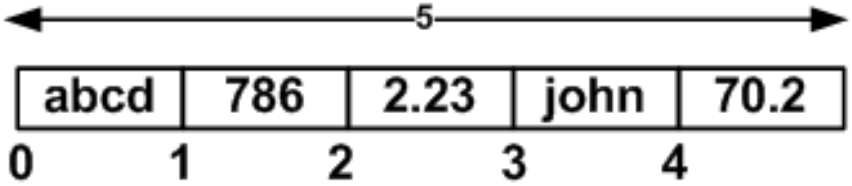
ตัวอย่าง myList = [‘Hello’ , ’my’ , ‘student’] การจัดเก็บแสดงได้ดังนี้



จากตัวอย่างจำนวน item หรือขนาด ของ list คือ 3
item เริ่มต้นอยู่ที่ตำแหน่ง 0
item สุดท้ายอยู่ที่ตำแหน่ง 2 (คือ end-1นั่นคือ 3-1=2)

ตัวอย่างการทำงานกับ ลิสต์

```
exlist1 = ['abcd', 786 , 2.23, 'john', 70.2 ]
exlist2 = ['Hello' ,'my' , 'student']
```




```
print (exlist1)          # Prints complete list
print (exlist1[0])       # Prints first element of the list
print (exlist1[1:3])     # Prints elements starting from 2nd till 3rd (end-1 คือ 3-1=2)
print (exlist1[2:])      # Prints elements starting from 3rd element
print (exlist2 * 2)      # Prints list two times
print (exlist1 + exlist2)          # Prints concatenated lists
```

This will produce the following result:

```
['abcd', 786, 2.23, 'john', 70.2]
abcd
[786, 2.23]
[2.23, 'john', 70.2]
['Hello', 'my', 'student', 'Hello', 'my', 'student']
['abcd', 786, 2.23, 'john', 70.2, 'Hello', 'my', 'student']
```

5.ตัวดำเนินการ (Operator)

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)
- ตัวดำเนินการเปรียบเทียบ (Comparison (i.e., Relational) Operators)
- ตัวดำเนินการกำหนดค่า (Assignment Operators)
- ตัวดำเนินการตรรกะ (Logical Operators)
- Bitwise Operators
- Membership Operators
- Identity Operators

ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)

Assume variable a holds 10 and variable b holds 20, then:

Operator	Description	Example	Result
+	Addition	a + b	30
-	Subtraction	a - b	-10
*	Multiplication	a * b	200
/	Division	b / a	2
%	Modulus	b % a	0
**	Exponent	2**3	8
/ /	Floor Division	9//2	4
		9.0//2.0	4.0

```
7 / 3      # classic division returns a float 2.3333333333333335
17 // 3    # floor division discards the fractional part 5
```

ตัวดำเนินการเปรียบเทียบ (Comparison Operators)

Assume variable a holds 10 and variable b holds 20, then:

Operator	Description	Example	result
==	equal	a == b	False
!=	not equal	a != b	True
>	greater than	a > b	False
<	less than	a < b	True
>=	greater than or equal	a >= b	False
<=	less than or equal	a <= b	True

ตัวดำเนินการกำหนดค่า (Assignment Operators)

Operator	Example	Explanation
=	c = a + b	
+=	c += a	c = c + a
-=	c -= a	c = c - a
*=	c *= a	c = c * a
/=	c /= a	c = c / a
%=	c %= a	c = c % a
**=	c **= a	c = c ** a
//=	c //= a	c = c // a

ตัวดำเนินการตรรกะ (Logical Operators)

Assume variable a holds 10 and variable b holds 20, then:

Operator	Example	Result
and	a and b	True
or	a or b	True
not	not(a and b)	False

ตารางค่าความจริง

P	Q	P and Q	P or Q	P	Not P
True	True	True	True	True	False
True	False	False	True	False	True
False	True	False	True		
False	False	False	False		

Operators

นอกจากนี้ยังมี ตัวดำเนินการอื่น ๆ ซึ่งในที่นี้ยังไม่กล่าวถึง

- **Bitwise Operator ตัวดำเนินการทางบิต : >> , << , & , |**
- **Python Membership Operators : in , not in**
- **Python Identity Operators: is , is not**

ลำดับความสำคัญของเครื่องหมาย (Python Operators Precedence)

Operator	Description
()	parentheses
**	Exponentiation (raise to the power)
+ - ~	Unary plus , unary minus , bitwise NOT
* / // %	Multiplication, division, floor division , modulus
+ -	Addition and subtraction
<< >>	Bitwise shift operation
&	Bitwise AND
^	Bitwise XOR
|	Bitwise OR
in, not in, is, is not, <, <=, >, >=, !=, ==	Comparisons, including membership tests and identity tests
not	Logical NOT
and	Logical AND
or	Logical OR

นิพจน์(expression)

เราอาจคุ้นเคยกับรูปแบบการเขียนสมการคณิตศาสตร์ เช่น **xy - 5z** แต่เมื่อเขียนโปรแกรมเราต้องเขียนเป็นนิพจน์คือ รูปแบบคำสั่งที่ภาษาเข้าใจ

เช่น สมการคณิตศาสตร์

$xy - 5z$

$x^2 + 4y + 5$

$8y^2 - 8z$

เขียนเป็นนิพจน์ในภาษาไพทอน ได้ดังนี้

```
x * y – 9 * z
x**2 + 4 * y + 5
8 * y**2 - 8 * z
```

ตัวอย่างลำดับความสำคัญของตัวดำเนินการ (ต่อ)

ตัวอย่าง 1 จงหาผลลัพธ์ที่ได้จาก นิพจน์ c * d - x / y

กำหนดให้ตัวแปรแต่ละตัวมีค่าดังนี้

x = 16

y = 4

c = 2.5

d = 0.25

วิธีคิด

c * d - x / y

แทนค่า 2.5 * 0.25 - 16 / 4

0.625 – 4

-3.375 **ตอบ**

ตัวอย่างลำดับความสำคัญของตัวดำเนินการ (ต่อ)

ตัวอย่าง 2 จงหาผลลัพธ์ที่ได้จาก นิพจน์ -(-5) * (x % y – x // (y+1))

กำหนดให้ตัวแปรแต่ละตัวมีค่าดังนี้

x = 16

y = 4

c = 2.5

d = 0.25

วิธีคิด

-(-5) * (x % y – x // (y+1))

แทนค่า -(-5) * (16 % 4 – 16 // (4+1))

-(-5) * (16 % 4 – 16 // (5))

5 * (16 % 4 – 16 // 5)

5 * (0 – 3)

5 * (– 3)

-15 **ตอบ**

ตัวอย่างลำดับความสำคัญของตัวดำเนินการ (ต่อ)

ตัวอย่าง 3 จงหาผลลัพธ์ที่ได้จาก นิพจน์ c // d + y % x != 2*c – d

กำหนดให้ตัวแปรแต่ละตัวมีค่าดังนี้

x = 16

y = 4

c = 2.5

d = 0.25

วิธีคิด

c // d + y % x != 2*c – d

แทนค่า

2.5 // 0.25 + 4 % 16 != 2*2.5 – 0.25

10 + 4 != 5 – 0.25

14 != 4.75

True ตอบ

6. คำสั่งรับและแสดงผล

การทำงานเพื่อติดต่อกับผู้ใช้ เราสามารถเรียกใช้ฟังก์ชันเพื่อรับ หรือแสดงผล ลักษณะของฟังก์ชัน คือ คำสั่งหลาย ๆคำสั่ง ที่สร้างไว้แล้ว เราสามารถเรียกใช้ได้ แต่ต้องเป็นไปตามรูปแบบที่กำหนด

ในส่วนนี้ เพื่อรับ และแสดงผล เราจะศึกษาดังนี้

- คำสั่งแสดงผล ใช้ฟังก์ชัน print()
- คำสั่งรับข้อมูล ใช้ฟังก์ชัน input()

คำสั่งแสดงผล print()

ฟังก์ชัน print() แสดงผลที่จอภาพ

ในเครื่องหมาย ( ) จะส่งข้อมูล : ตัวแปร,ข้อความ เพื่อแสดงผลที่จอภาพ ข้อมูลแยกด้วยเครื่องหมาย ,

ตัวอย่าง 1

```
>>> print ("Welcome ")
Welcome
>>> print ("to python" )
to python
>>>
```

ตัวอย่าง 3

```
>>> name='Suda'
>>> ver=3.8
>>> print("Hello ! ", name, "Welcome to python", ver)
Hello !  Suda Welcome to python 3.8
>>>
```

ตัวอย่าง 2

```
>>> print ("Python is really a great language," , "isn't it? ")
Python is really a great language, isn't it?
```

คำสั่งแสดงผล print()

นอกจากนี้ หากเราต้องการแสดงผล การขึ้นบรรทัดใหม่ ใช้ \n

ตัวอย่าง

print ("Welcome \nto \nPython")

ผลลัพธ์

```
>>>
Welcome
to
Python!
>>>
```

ส่วนการแสดงผลในรูปแบบอื่น เช่น การแสดงจำนวนตัวเลขทศนิยม การจัดรูปแบบช่องว่างที่สวยงาม จะกล่าวถึงในส่วนที่เกี่ยวข้องต่อไป

คำสั่งรับข้อมูล input()

ฟังก์ชัน input() **จะรอรับข้อมูลจากผู้ใช้ ดังนั้นเราต้องพิมพ์ข้อมูลเข้าผ่านทางคีย์บอร์ด ปกติเราใส่ข้อมูล** 1 **ตัว (เช่น ข้อความ หรือ ตัวเลข) แล้วเคาะ** enter

ข้อมูลที่เราพิมพ์นั้นจะเป็น string (function return string)

เราต้องสร้างตัวแปร เพื่อเก็บข้อมูลดังกล่าว

ตัวอย่าง

```
test = input("Enter your input: ")
print ("Received input is : ", test)
```

สมมุติผู้ใช้พิมพ์ข้อความ Hello Python **ตัวอย่างการทำงานและผลลัพธ์ จะได้ดังนี้**

```
Enter your input: Hello Python
Received input is : Hello Python
```

จากตัวอย่าง ข้อความ Hello Python ที่เราพิมพ์ จะถูกเก็บไว้ที่ตัวแปร test

คำสั่งรับข้อมูล input() (cont.)

ฟังก์ชัน input จะรอให้ผู้ใช้พิมพ์ข้อมูลจากคีย์บอร์ด ซึ่งข้อมูลที่รับมานั้นจะเป็น สตริง นำไปคำนวณไม่ได้ถึงแม้เราจะพิมพ์ตัวเลขก็ตาม หากเราต้องการนำไปคำนวณ เราต้องแปลงสตริงเป็นตัวเลขที่ต้องการ เช่น

แปลงจาก string เป็น int หรือ แปลงจาก string เป็น float

ตัวอย่าง

```
1  inp1=input("Input  integer number  : ")
2  no1=int(inp1)
3  inp2=input("Input  float number  : ")
4  no2=float(inp2)
```

อธิบายได้ดังนี้

บรรทัดที่ 1 จะปรากฏข้อความ Input integer number ที่จอภาพ เคอร์เซอร์กระพริบ รอให้ผู้ใช้พิมพ์ สมมติผู้ใช้พิมพ์ 55 เคาะ enter ผลคือ string 55 จะถูกเก็บที่ตัวแปร inp1

บรรทัดที่ 2 จะแปลง string 55 ซึ่งเก็บที่ ตัวแปร inp1 ให้เป็นตัวเลขจำนวนเต็มเก็บไว้ที่ตัวแปร no1 จากนี้เราสามารถนำ no1 ไปคำนวณได้

บรรทัดที่ 3 และ 4 ก็เช่นเดียวกันกับบรรทัดที่ 1 และ 2 เพียงแต่แปลง จาก string เป็น float

คำสั่งรับข้อมูล input() (cont.)

ข้อควรระวัง เราต้องทราบว่าโปรแกรมเรามีข้อมูลเข้าอะไรบ้าง เพื่อที่ว่าเราจะได้ออกแบบ ออกแบบชนิดข้อมูล และนำไปเขียนโปรแกรมได้ถูกต้อง

เช่น

```
inp=input("Input  number  : ")
no=int(inp)
```

เมื่อโปรแกรมทำงาน

เราใส่ 1.2 เก็บที่ตัวแปร string ok ไม่มีปัญหา

แต่ตอนแปลงใช้ int เกิด error

7.ปฏิบัติการ

(1) เรียนรู้การใช้โปรแกรม **Python**

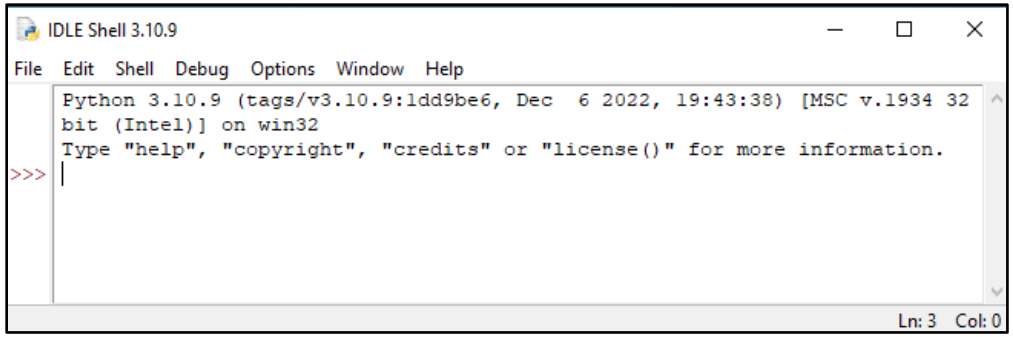
(2) เริ่มต้นเขียนโปรแกรมอย่างง่าย

(3) แบบฝึกหัด

(1) เรียนรู้การใช้โปรแกรม Python

ภาษาไพทอน execute ได้ 2 mode คือ

(1.1) INTERACTIVE MODE PROGRAMMING: เป็น mode ที่เราพิมพ์คำสั่ง ภาษาไพทอน จะแปลและทำงานทันที เช่น



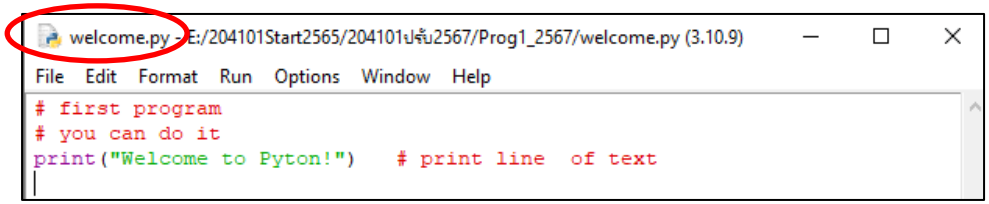
(1.2) SCRIPT MODE PROGRAMMING:

พิมพ์คำสั่งหรือโปรแกรมที่ editor หรือ IDLE ของไพทอน (write a simple Python program in a script) จากนั้น save file และกำหนด file type เป็น .py

เมื่อรันโปรแกรมจะได้ผลลัพธ์ เช่น

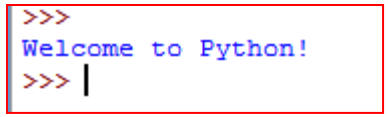
source code in a welcome.py file

print "Welcome to Python!"



This will produce the following result:

Welcome to Python!



(2) เริ่มต้นเขียนโปรแกรมอย่างง่าย

ตัวอย่าง ใช้คำสั่ง print() แสดงผลที่จอภาพ save file, run module

```
# first program
# you can do it
print(“Welcome to Python!”)   #  print line of text
```

```
print ("Welcome ")
print ("to python" )
```

```
print ("Welcome \nto \nPython")
```

```
print ("Python is really a great language," , "isn't it? “)
```

```
name=‘Suda'
ver=3.8
print ("Hello ! "    , name , "Welcome to python“ , ver)
```

ตัวอย่าง ใช้คำสั่ง input()

```
test = input("Enter your input: ")
print ("Received input is : ", test)
```

```
inp1=input("Input  integer number  : ")
no1=int(inp1)
inp2=input("Input  float number  : ")
no2=float(inp2)
```

ที่ต้องระวังคือ

```
inp1=input("Input  number  : ")
no1=int(inp1)
```

เมื่อโปรแกรมทำงาน

เราใส่ 1.2 เก็บที่ตัวแปร string ok ไม่มีปัญหา

แต่ตอนแปลงใช้ int เกิด error

ตัวอย่าง

จงเขียนโปรแกรมรับ ตัวเลข 2 จำนวนเต็ม นำตัวเลขมาบวกกัน และแสดงผลลัพธ์ที่จอภาพ

```
inp1=input("Input  integer number  : ")
no1=int(inp1)
inp2=input("Input  float number  : ")
no2=float(inp2)
print('result is = ' no1+no2)
```

save file ชื่อ pr01.py
run module
ดูผลการทำงาน

(3) แบบฝึกหัด ข้อ 1

แสดงผลข้อความที่จอภาพ ตามรูปแบบที่แสดงด้านล่าง กำหนดให้ใช้ คำสั่ง print 1 คำสั่ง

```
>>>
Dear Pa and Ma
        I will get 'A' for this course
        80 is my score
love you very much so much
your baby-Ja
>>>
```

แบบฝึกหัด ข้อ 2

จงเขียนโปรแกรมรับ ชื่อ นามสกุล และอายุ แล้วแสดงผลที่จอภาพ ตามที่กำหนด โดยอายุที่รับให้บวก อีก 4 เพื่อบอกว่าจะสำเร็จการศึกษาเมื่ออายุเท่าไร

เช่น

Input name: *Chalee*

Input surname: *Buddee*

Input age: *18*

ผลลัพธ์

My name is Chalee

And surname Buddee

Now I'm 18 years old.

I'll finish undergraduate in 22 years old.

Reference

- Deitel ,"Python How to program" , "Prentice-Hall,Inc.", 2002.
- Matt Telles , "Python Power !" , "Thomson Course Teachnology",2008.
- *python 3.4.3 help documentation*
- PYTHON TUTORIAL *Simply Easy Learning by tutorialspoint.com*