

String Numbers and Date & Time

1. String

เป็นชนิดข้อมูลที่สามารถนำมาใช้ประโยชน์ได้อย่างมีประสิทธิภาพคือ String เนื่องจาก String สามารถบรรจุได้ทั้ง ตัวอักษร ตัวเลข และสัญลักษณ์

```
name = "Ryan"
```

```
age = "19"
```

```
food = "cheese"
```

รูปที่ 1 การสร้างตัวแปร String

ในรูปที่ 1 จะเป็นประกาศตัวแปร name และกำหนดให้ค่าของ String name มีค่าเท่ากับ “Ryan” String age มีค่าเท่ากับ “19” และ String food มีค่าเท่ากับ “cheese” ซึ่งการกำหนดค่าของ String นั้นจะต้องอยู่ในเครื่องหมาย “ ” เท่านั้น

การจัดรูปแบบ String (String Concatenation)

การรวม String นั้นสามารถทำได้โดยการใช้การดำเนินการทางคณิตศาสตร์ ซึ่งสามารถทำได้ด้วยการใช้ เครื่องหมาย + ดังตัวอย่างต่อไปนี้

```
print"Life " + "of " + "Brian"
```

```
//จะแสดงผล เป็น Life of Brian ออกทางหน้าจอ
```

```
'There's a snake in my boot!'
```

ถ้าพิจารณาจากส่วนของโปรแกรมข้างต้นจะเห็นได้ว่า Python จะปิดประโยคที่เครื่องหมายอัฒประกาศ (“ ”) ผลที่ได้จะเป็นดังนี้ **'There'** สามารถแก้ไขปัญหานี้ได้โดยการเติม \ (backslash) ดังตัวอย่างต่อไปนี้

```
'There\'s a snake in my boot!'
```

การจัดรูปแบบ String โดยใช้ %

เราสามารถแสดงผลข้อความร่วมกับตัวแปรได้โดยนำ % ไปใส่หลัง String ที่จะนำไปรวมกับตัวแปร ดังตัวอย่างต่อไปนี้

```
name = "Mike"

print"Hello %s" % (name)
```

การเข้าถึงด้วยดัชนี (Access by Index)

การเข้าถึง Character แต่ละตัวใน String จะสามารถทำได้โดยการกำหนดตำแหน่งตัวเลขในแบบ Array โดยที่ตัวเลขที่กำหนดนั้นจะทำการชี้ไปยัง Character ที่ตำแหน่งนั้นใน String เช่น

```
c = "cats"[0]
```

จากตัวอย่างจะเป็นการสร้างตัวแปร c ที่มีการกำหนดค่าให้เท่ากับ “c” เนื่องจาก ในตำแหน่งที่ 0 ของ “cats” นั้นคือ c เนื่องจากใน python จะเริ่มต้นการกำหนดค่า Index ให้เริ่มที่ค่า ศูนย์

c	a	t	s
[0]	[1]	[2]	[3]

และเรายังอ้างถึงช่วงของ String โดยการใช้อำนาจเครื่องหมาย :(colon) ซึ่งเรียกช่วงของ String นี้ว่า Slice ดังตัวอย่างต่อไปนี้

```
Word = "String"

>>>Word[0:2] → "r"

>>>Word[2:4] → "rin"
```

ถ้าไม่ได้ค่า Index ข้างหน้าจะถูกตีค่าเริ่มต้นเป็น ศูนย์ และตัวหลังจะถูกตีความเป็นความยาวของตัว String

```
Word = "String"

0]    [1]    [2]    [3]    [4]    [5]
s     t     r     i     n     g

>>>Word[0:2] → "st"

>>>Word[2:4] → "ri"
```

ถ้าค่า Index ที่เป็นค่านอกช่วงที่มีจริง Python จะแสดงเป็นอักขระว่างให้โดยไม่รายงานความผิดพลาด

```
Word = "String"

>>>Word[1:100] → "tring"

>>>Word[10:] → ""

>>>Word[2:1] → ""
```

ถ้าค่า Index เป็นค่าติดลบ จะเป็นการนับจากขวาไปซ้าย

Word = "String"

>>>Word[-1] → "g" #g เป็นอักขระตัวสุดท้าย

>>>Word[-2] → "n"

>>>Word[-2:] → "ng" #n และ g เป็นอักขระสองตัวสุดท้ายใน String

>>>Word[:-2] → "Stri"

>>>Word[-0] → "S" #จะยกเว้น -0 เนื่องจากมีค่าเท่ากับ 0

S	t	r	i	n	g
[0]	[1]	[2]	[3]	[4]	[5]
[0]	[-5]	[-4]	[-3]	[-2]	[-1]

เมธอดของ String (String methods)

1. len() เป็นคำสั่งที่คืนค่าเป็นความยาวของ String

```
Parrot = "String"

printlen(Parrot)    //จะแสดงผลเป็นเลข 6 ออกทางหน้าจอ
```

2. lower() เป็นคำสั่งที่จะทำการเปลี่ยน String เป็นตัวอักษรพิมพ์เล็กทั้งหมด

```
Parrot = "String"

Print Parrot.lower()    //จะแสดงผล string ออกทางหน้าจอ
```

3. upper() เป็นคำสั่งที่จะทำการเปลี่ยน String เป็นตัวอักษรพิมพ์ใหญ่ทั้งหมด

```
Parrot = "String"

printParrot.upper()    //จะแสดงผล String ออกทางหน้าจอ
```

4. str() เป็นคำสั่งที่จะทำการเปลี่ยน Non-String เป็น String

```
Pi = 3.14

printstr(Pi)    //จะแสดงผล 3.14ออกทางหน้าจอ
```

2. Numbers

การกำหนดตัวแปรชนิดที่เป็นตัวเลขเป็นชนิดข้อมูลที่ไม่เปลี่ยนรูปซึ่งหมายความว่าเป็นการเปลี่ยนค่าของผลลัพธ์ชนิดข้อมูลตัวเลข ไปอยู่ในพื้นที่ที่ใช้ในการจัดเก็บข้อมูลและถูกสร้างขึ้นเมื่อผู้ใช้กำหนดค่าที่จะเก็บไว้ในตัวแปร ด้วยเครื่องหมายเท่ากับ (=)

```
Number = 2
```

```
Number = 44.50
```

กำหนดค่าที่หลายตัวแปรพร้อมกันก็ได้

```
>>>x = y = z = 0
```

```
>>>x      ➔ output = 0
```

```
>>>y      ➔ output = 0
```

```
>>>z      ➔ output = 0
```

ในการคำนวณทางคณิตศาสตร์ ซึ่งใน Python จะเรียงลำดับตามความสำคัญทางคณิตศาสตร์เหมือนภาษาอื่นๆ สามารถทำได้ดังนี้

```
>>>2+2      ➔ output = 4
```

```
>>>(50-5*6)/4 ➔ output = 5
```

```
>>>7/3      ➔ output = 2
```

```
>>>7/-3     ➔ output = -3
```

และ กำหนดค่าให้ตัวแปรด้วย = ตามปกติ

```
>>> width = 20

>>> height = 5*9

>>> area = width*height → output = 900

>>> print("%.2f" % area) → output = 900.00    #การ print เป็นทศนิยม 2 ตำแหน่ง
```

รองรับทศนิยมจุดลอยตัว (floating point) ด้วย โดยมีหลักว่า ถ้าต้นทางเป็นจำนวนเต็ม ผลจะเป็นจำนวนเต็ม ถ้าต้นทางเป็นทศนิยม ผลจะเป็นทศนิยม (ถ้าผสมกันก็จะเป็นทศนิยมเช่นกัน)

```
>>> 3 * 3.75 / 1.5    → output = 7.5

>>> 7 / 2            → output = 3

>>> 7.0 / 2         → output = 3.5
```

สำหรับ โหมดได้ตอบ ค่าที่ถูกพิมพ์ออกมาเป็นครั้งสุดท้าย จะถูกเก็บไว้ในตัวแปรพิเศษคือ `_` เราอาจนำตัวแปรนี้ไปใช้ในการคำนวณค่าต่อ ๆ ไปได้

```
>>> tax = 12.5/100

>>> price = 100.50

>>> price*tax        → output = 12.5625

>>> price + _        → output = 113.0625    # 100.50+12.5625

>>> round(_, 2)      → 113.06
```

3. Date & Time

ภาษา Python สามารถจัดการกับวันที่และเวลาได้หลายวิธี การใช้ date & time ในภาษา python จะใช้เพื่อแสดงวันที่และเวลาในรูปแบบที่เหมาะสม

ช่วงเวลาในหนึ่งหน่วยเวลาเป็นเลข floating-point โดยเฉพาะเวลาที่แสดงในรูปแบบวินาที ตั้งแต่ 12:00 am, January 1, 1970

โมดูลเวลานิยมใช้ในภาษาPython ซึ่งมีฟังก์ชันให้ใช้สำหรับทำงานเกี่ยวกับเวลาและใช้ฟังก์ชัน time.time () คืนค่าเวลา ตั้งแต่ 12:00 am, January 1, 1970

พิจารณาตัวอย่างโปรแกรมในภาษาPython ต่อไปนี้

```
#!/usr/bin/python

import time;

ticks = time.time()

print "Number of ticks since 12:00am, January 1, 1970:", ticks

#will print      Number of ticks since 12:00am, January 1, 1970: 7186862.73399
```

•TimeTuple

หลายฟังก์ชันที่เกี่ยวกับเวลาของภาษาPython มีการจัดการเวลาเป็น 9 อันดับ ดังตารางข้างล่างนี้

Index	Field	Values
0	4-digit year	2008
1	Month	1 to 12
2	Day	1 to 31
3	Hour	0 to 23
4	Minute	0 to 59
5	Second	0 to 61 (60 or 61 are leap-seconds)
6	Day of Week	0 to 6 (0 is Monday)
7	Day of year	1 to 366 (Julian day)
8	Daylight savings	-1, 0, 1, -1 means library determines DST

จากตารางข้างต้นเทียบกับ attributes มีโครงสร้าง struct_time ดังต่อไปนี้

Index	Attributes	Values
0	tm_year	2008
1	tm_mon	1 to 12
2	tm_mday	1 to 31
3	tm_hour	0 to 23
4	tm_min	0 to 59
5	tm_sec	0 to 61 (60 or 61 are leap-seconds)
6	tm_wday	0 to 6 (0 is Monday)
7	tm_yday	1 to 366 (Julian day)
8	tm_isdst	-1, 0, 1, -1 means library determines DST

●การรับค่าเวลาปัจจุบัน (Getting current time)

แปลงเวลาจากวินาที ตั้งแต่เลข floating-point เป็น time-tuple ใช้ฟังก์ชัน time.localtime () คืนค่าเป็น time-tuple

พิจารณาตัวอย่างโปรแกรมในภาษาPython ต่อไปนี้

```
#!/usr/bin/python
import time;
localtime = time.localtime(time.time())
print "Local current time :", localtime

#will print      Local current time : time.struct_time(tm_year=2014, tm_mon=11,
                  tm_mday=27, tm_hour=19, tm_min=6, tm_sec=42, tm_wday=3,
                  tm_yday=331, tm_isdst=0)
```

●การจัดรูปแบบ Time (Getting formatted time)

สามารถจัดรูปแบบ Time ตามความต้องการ แต่เมตซอดที่จะรับเวลาในรูปแบบที่สามารถอ่านได้
คือ asctime()

พิจารณาตัวอย่างโปรแกรมในภาษาPython ต่อไปนี้

```
#!/usr/bin/python
import time;

localtime = time.asctime( time.localtime(time.time()) )

print "Local current time :", localtime

#will print      Local current time : Thu Nov 27 19:15:12 2014
```

●การรับค่าปฏิทินจากเดือน (Getting calendar for a month)

โมดูลปฏิทินให้ช่วงความกว้างของเมตซอดที่จะใช้กับปฏิทินรายปีและรายเดือน สามารถพิมพ์
ปฏิทินให้ตรงกับเดือนที่ต้องการได้

พิจารณาตัวอย่างโปรแกรมในภาษาPython ต่อไปนี้ (ต้องการปฏิทินเดือนมกราคม, 2008)

```
#!/usr/bin/python
import calendar
cal = calendar.month(2008, 1)
print "Here is the calendar:"
print cal;

#will print      Here is the calendar:
                  January 2008
                  Mo   Tu   We   Th   Fr   Sa   Su
                   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
                  28   29   30   31
```

●Getting the Current Date and Time

การใช้ฟังก์ชัน `datetime.now()` เพื่อเรียกวันที่และเวลาในปัจจุบันในรูปแบบอย่างง่าย
พิจารณาตัวอย่างโปรแกรมในภาษาPython ต่อไปนี้

```
from datetime import datetime # เป็นการเรียกใช้ datetime เพื่อให้โปรแกรมสามารถใช้งานได้

now = datetime.now()

print now # จะแสดงผลออกมาเป็นวันที่และเวลาในปัจจุบัน

# will print: 2013-11-25 23:45:14.317454
```

ตัวอย่างที่ 1

ถ้าต้องการพิมพ์วันที่วันนี้ในรูปแบบอย่างง่าย เช่น `yyyy-mm-dd` หรือ `mm/dd/yyyy` ดังนี้

```
from datetime import datetime

now = datetime.now()

current_year = now.year      # current_year เก็บค่า ปีในปัจจุบัน

current_month = now.month   # current_month เก็บค่าเดือนในปัจจุบัน

current_day = now.day       # current_day เก็บค่าวันที่ในปัจจุบัน

print '%s-%s-%s' % (current_year, current_month, current_day)

# will print: 2014-02-19
```

ตัวอย่างที่ 2

ถ้าต้องการพิมพ์เวลาในรูปแบบของ hh:mm:ss ดังนี้

```
from datetime import datetime

now = datetime.now()

current_hour = now.hour

current_minute = now.minute

current_second = now.second

print '%s:%s:%s' % (current_hour, current_minute, current_second)

# will print: 13:49:54
```

ตัวอย่างที่ 3

```
from datetime import datetime

now = datetime.now()

print '%s/%s/%s' % (now.month, now.day, now.year) + '%s:%s:%s' % (now.hour, now.minute, now.second)

# will print: 11/13/2014 21:35:56
```

LAB

1. จากโปรแกรมจงพิจารณาโปรแกรมภาษา Python

```
#!/usr/bin/python
str = 'Hello World!'

print str           # Prints complete string
print str[0]       # Prints first character of the string
print str[2:5]     # Prints characters starting from 3rd to 5th
print str[2:]      # Prints string starting from 3rd character
print str * 2      # Prints string two times
print str + "TEST" # Prints concatenated string
```

ให้แสดงผลลัพธ์ของ string ในแต่ละบรรทัดให้ถูกต้อง

2. ให้ออกแบบปฏิทินของเดือน มกราคม ปี พ.ศ. 2015 โดยใช้ความรู้ในเรื่อง date & time

บรรณานุกรม

http://www.tutorialspoint.com/python/python_tutorial.pdf

<http://www.codecademy.com/en/tracks/python>

<https://docs.python.org/2/library/numeric.html>