**Python Type conversions**

April 21, 2017

ในบทนี้ คุณจะได้เรียนการแปลงประเภทข้อมูลในภาษา Python โดยการใช้ built-in ฟังก์ชัน เนื่องจากในการเขียนโปรแกรมนั้นเรามักจะทำงานกับข้อมูลหลายประเภท ดังนั้น มันจึงมีความจำเป็นที่คุณอาจจะต้องการแปลงข้อมูลประเภทหนึ่งไปเป็นประเภทหนึ่ง เช่น การแปลงตัวเลขให้เป็น String หรือแปลง String เป็นตัวเลข เป็นต้น

**ฟังก์ชันแปลงข้อมูลในภาษา Python**

ในภาษา Python คุณสามารถใช้ built-in ฟังก์ชันที่มีอยู่สำหรับแปลงประเภทข้อมูล โดยฟังก์ชันเหล่านั้นจะมีชื่อที่เหมือนกับประเภทของมัน (ซึ่งเป็นชื่อของคลาส) ยกตัวอย่างเช่น ฟังก์ชัน int() ใช้แปลงข้อมูลประเภทใดๆ ให้เป็นตัวเลขจำนวนเต็ม และ str() ใช้แปลงข้อมูลประเภทใดๆ ให้เป็น String นี่เป็นตารางของฟังก์ชันสำหรับแปลงข้อมูลในภาษา Python

|  |  |
| --- | --- |
| **Function** | **Description** |
| int(x [,base]) | แปลงออบเจ็ค x จากฐานที่กำหนด base ให้เป็น Integer |
| long(x [,base] ) | แปลงออบเจ็ค x จากฐานที่กำหนด base ให้เป็น Long |
| float(x) | แปลงออบเจ็ค x ให้เป็น Floating point number |
| complex(real [,im]) | สร้างตัวเลขจำนวนเชิงซ้อนจากค่า real และค่า imagine |
| str(x) | แปลงออบเจ็ค x ให้เป็น String |
| repr(x) | แปลงออบเจ็ค x ให้เป็น String expression |
| eval(str) | ประเมินค่าของ String |
| tuple(s) | แปลง Sequence ให้เป็น Tuple |
| list(s) | แปลง Sequence ให้เป็น List |
| set(s) | แปลง Sequence ให้เป็น Tuple |
| dict(d) | แปลงออบเจ็คให้เป็น Dictionary |
| frozenset(s) | แปลงออบเจ็คให้เป็น Frozen set |
| chr(x) | แปลงค่าของ Integer ให้เป็น Unicode Char |
| ord(x) | แปลง Charterer ให้เป็นค่า Integer |
| hex(x) | แปลง Integer ให้เป็น Hex string |
| oct(x) | แปลง Integer ให้เป็น Oct string |

**การแปลง String เป็นตัวเลข**

หลังจากที่คุณได้เห็นฟังก์ชันสำหรับแปลงข้อมูลในภาษา Python แล้ว ต่อไปจะเป็นตัวอย่างในการใช้งานฟังก์ชันเหล่านี้ ตัวอย่างแรกจะเป็นโปรแกรมในการแปลงข้อมูลจาก String ไปเป็นตัวเลข ซึ่งเป็นสิ่งที่พบได้บ่อยมาก เราจะเขียนโปรแกรมในการรับค่า String จากทางคีย์บอร์ดและแปลงให้เป็นตัวเลขแบบต่างๆ นี่เป็นโค้ดของโปรแกรม

# convert string to int  
s = input('Enter integer number: ')  
n = int(s)  
  
print('Type of s is ', type(s))  
print('Type of n is ', type(n))  
print('Value of n is', n)  
  
# convert string to float  
s2 = input('Enter floating number: ')  
f = float(s2)  
  
print('Type of s2 is ', type(s2))  
print('Type of f is ', type(f))  
print('Value of f is', f)

ในตัวอย่าง โปรแกรมของเรานั้นได้แบ่งออกเป็นสองส่วน ส่วนแรกเป็นการแปลง String ไปเป็น Integer และในส่วนต่อมาเป็นการแปลง String ไปเป็น Float โปรแกรมได้ทำการรับค่า String จากคีย์บอร์ดด้วยฟังก์ชัน input() และเราใช้ฟังก์ชัน int() และ float() เพื่อแปลงข้อมูลจาก String เป็น Integer และ Float ตามลำดับ และฟังก์ชัน type() นั้นใช้สำหรับดูประเภทข้อมูลของออบเจ็คในรูปแบบคลาส Prototype ของมัน

Enter integer number: 10  
Type of s is <class 'str'>  
Type of n is <class 'int'>  
Value of n is 10  
Enter floating number: 3.5  
Type of s2 is <class 'str'>  
Type of f is <class 'float'>  
Value of f is 3.5

นี่เป็นผลลัพธ์การทำงานของโปรแกรม โดยเราได้กรอกค่าเป็น 10 และ 3.5 ตามลำดับ ซึ่งค่าที่กรอกนั้นเป็น String และหลังจากนั้นเราแปลงเป็นตัวเลขและแสดงผลประเภทและค่าของมันออกทางหน้าจอ

print(str(10 + 20))  
print(str(1.234))  
print(str([1, 2, 3, 4, 5, 6]))

คุณยังสามารถแปลงข้อมูลจากประเภทอื่นๆ ให้เป็น String ได้โดยการใช้ฟังก์ชัน str() ซึ่งออบเจ็คส่วนมากนั้นสามารถใช้ได้กับฟังก์ชันนี้ ในตัวอย่างของทั้งสามคำสั่ง เป็นการแปลงจาก Integer Floating number และ List ไปเป็น String ตามลำดับ

30  
1.234  
[1, 2, 3, 4, 5, 6]

นี่เป็นผลลัพธ์การทำงานของโปรแกรมในการแปลงออบเจ็คอื่นให้เป็น String

**การใช้ฟังก์ชันแปลงข้อมูลในภาษา Python**

ในภาษา Python นอกจากฟังก์ชันที่ทำงานกับ String และตัวเลขแล้ว ยังมีฟังก์ชันอื่นๆ อีกเป็นจำนวนมากสำหรับการแปลงข้อมูล มาดูตัวอย่างการใช้งานฟังก์ชันเพิ่มเติม

# convert integer to char  
print('65 to char =', chr(65))  
print('70 to char =', chr(70))  
print('97 to char =', chr(97))  
print('196 to char =', chr(196))  
print('473 to char =', chr(473))  
  
# convert char to integer  
site = 'marcuscode.com'  
  
for s in site:  
 print(s, ' = ', ord(s))

โดยปกติแล้วตัวอักษรทุกตัวในคอมพิวเตอร์จะมีรหัสของมันที่เป็นตัวเลข ในตัวอย่าง เป็นการใช้งานฟังก์ชัน chr() สำหรับแปลงค่า Integer ให้เป็นรหัสตัวอักษร Unicode โดย 65 นั้นเป็นรหัสของตัวอักษรพิมพ์ใหญ่ตัวแรกในภาษาอังกฤษ (A) 70 นั้นเป็นรหัสของตัวอักษร F 97 เป็นรหัสของตัวอักษรพิมพ์เล็กตัวแรกในภาษาอังกฤษ (a) 196 เป็นของตัวอักษร Ä และ 473 เป็นของตัวอักษร Ǚ ตามลำดับ โดยที่รหัส Unicode นั้นสามารถแสดงถึงตัวอักษรทุกตัวที่มีในโลก

ในทางกลับกัน ฟังก์ชัน ord() ใช้สำหรับแปลงจากตัวอักษรไปเป็นค่า Integer ของมัน ในตัวอย่าง เรามี String ในตัวแปร siteที่เก็บค่า 'marcuscode.com' และใช้คำสั่ง For loop วนอ่านตัวอักษรและแสดงผลค่า Integer ของตัวอักษรแต่ละตัวภายใน String

65 to char = A  
70 to char = F  
97 to char = a  
196 to char = Ä  
473 to char = Ǚ  
m = 109  
a = 97  
r = 114  
c = 99  
u = 117  
s = 115  
c = 99  
o = 111  
d = 100  
e = 101  
. = 46  
c = 99  
o = 111  
m = 109

นี่เป็นผลลัพธ์การทำงานของโปรแกรมจากการใช้งานฟังก์ชัน chr() และฟังก์ชัน ord()สำหรับการแปลงข้อมูลระหว่างตัวอักษรกับค่า Integer ของมัน

**การใช้ฟังก์ชันแปลงข้อมูลประเภท Sequence และ Set**

ในภาษา Python มีฟังก์ชันสำหรับทำงานกับ Sequence และ Set โดยปกติแล้วฟังก์ชันเหล่านี้สามารถที่จะแปลงไปกลับได้ เช่น การแปลงจาก Tuple ไปยัง List ด้วยฟังก์ชัน list() และหลังจากนั้นทำการแปลงจาก List เป็น Tuple ด้วยฟังก์ชัน tuple() เป็นต้น ฟังก์ชัน dict()แปลงข้อมูลแบบ Mapping ให้เป็น Dictionary ส่วนฟังก์ชัน set() นั้นแปลงข้อมูลแบบ Sequence ใดๆ ให้เป็น Set มาดูตัวอย่าง

a = list((1, 2, 3, 4, 5)) # tuple to list  
b = tuple([1, 2, 3, 4, 5]) # list to tuple  
c = set([1, 2, 3, 4, 5]) # list to set  
d = dict(one= 1, two = 2, three = 3) # mapping to dict  
e = dict([('one', 1), ('two', 2), ('three', 3)]) # iterable list to dict  
  
print(a)  
print(b)  
print(c)  
print(d)  
print(e)

ในตัวอย่าง เป็นการแปลงข้อมูลโดยการใช้งานฟังก์ชันต่างๆ ในสามตัวแปรแรก เป็นการแปลงข้อมูลแบบ Sequence ให้เป็นออบเจ็คที่ต้องการ ในตัวแปร a เป็นการแปลงข้อมูลจาก Tuple ไปยัง List ในตัวแปร b เป็นการแปลงข้อมูลจาก List ไปยัง Tuple ส่วนในตัวแปร c เป็นการแปลงจาก List เป็น Set และในสองคำสั่งสุดท้าย ตัวแปร d เป็นการแปลข้อมูลแบบ Mapping เป็น Dictionary ตัวแปร e แปลงจาก List ของออบเจ็คเป็น Dictionary

[1, 2, 3, 4, 5]  
(1, 2, 3, 4, 5)  
{1, 2, 3, 4, 5}  
{'one': 1, 'two': 2, 'three': 3}  
{'one': 1, 'two': 2, 'three': 3}

นี่เป็นผลลัพธ์การทำงานของโปรแกรมในการใช้งานฟังก์ชันแปลงข้อมูลประเภท Sequence และ Set

ในบทนี้ เราได้แนะนำให้คุณรู้จักกับฟังก์ชันในการแปลงประเภทข้อมูลในภาษา Python ที่จะมีความจำเป็นเมื่อโปรแกรมของเรานั้นทำงานกับข้อมูลที่หลากหลายประเภทและต้องการเปลี่ยนประเภทข้อมูลประเภทหนึ่งไปอีกประเภทหนึ่ง เราได้แสดงตัวอย่างการใช้งานฟังก์ชันต่างๆ เช่น การแปลง String เป็นตัวเลข การแปลง Integer เป็น Char และการใช้งานฟังก์ชันเกี่ยวกับ Sequence และ Set อย่างไรก็ตามมีฟังก์ชันอีกจำนวนมากที่เราไม่สามารถให้ตัวอย่างได้ทั้งหมด แต่มันคงไม่อยากเกินไปสำหรับคุณที่จะลองใช้งานฟังก์ชันเหล่านั้น