**String ในภาษา Python**

April 3, 2017 · 1,546 views

ในบทนี้ คุณจะได้เรียนรู้เกี่ยวกับ String ในภาษา Python เราจะพูดเกี่ยวกับการประกาศและใช้งาน String ในรูปแบบต่างๆ เพราะว่าเนื้อหาเกี่ยวกับ String นั้นมีค่อนข้างมาก ดังนั้นเราจึงได้รวบรวมเนื้อหาทั้งไว้ในบทนี้ และเราจะแนะนำการใช้งาน built-in function ในภาษา Python ที่สำคัญในการจัดการกับข้อมูลประเภท String

**การประกาศตัวแปร String**

String เป็นลำดับของตัวอักษรหลายตัวเรียงต่อกัน ซึ่งในภาษา Python นั้นการที่จะประกาศ String ค่าของมันจะอยู่ในเครื่องหมาย Double quote หรือ Single quote เท่านั้น มาดูตัวอย่างการประกาศตัวแปรของ String

name = "Mateo"
site = 'marcuscode.com'
str1 = "This is my string"
str2 = 'This is my string'

ในตัวอย่าง เราได้ประกาศ 4 ตัวแปรของ String คุณจะสังเกตว่าในการกำหนดค่าให้กับตัวแปรนั้น String literal จะถูกภายในเครืองหมาย Double quote ("") หรือ Single quote ('') เท่านั้น ซึ่งได้ผลการทำงานที่เหมือนกัน ขนาดของ String นั้นจะขึ้นกับจำนวนตัวอักษรภายใน String

อย่างไรก็ตาม ถึงแม้เราจะสามารถใช้ Double quote หรือ Single quote กับ String ได้ แต่มีสิ่งที่แตกต่างกันเล็กน้อยสำหรับการใช้งานทั้งสองแบบคือการใช้ตัวอักษรพิเศษใน String หรือเรียกว่า Escape character ลองมาดูตัวอย่างต่อไปนี้

sentent1 = "What's your name?"
sentent2 = 'I\'m Mateo.'
sentent3 = "He said \"I would learn Python first\"."
sentent4 = 'His teach replied "Oh well!"'
print(sentent1)
print(sentent2)
print(sentent3)
print(sentent4)

ในตัวอย่าง เป็นสิ่งที่แตกต่างของการประกาศ String ทั้งสองแบบกับ Escape character ซึ่งตัวอักษร ' และ " นั้นเป็น Escape character ดังนั้นในการใช้งานตัวอักษรเหล่านี้ เราต้องใส่เครื่องหมาย \ ลงไปข้างหน้าเสมอ แต่ในภาษา Python เมื่อคุณใช้ Double quote ในการประกาศ String คุณไม่ต้องทำการ Escape character สำหรับ Single quote และในทางกลับกัน

What's your name?
I'm Mateo.
He said "I would learn Python first".
His teach replied "Oh well!"

นี่เป็นผลลัพธ์การทำงานของโปรแกรมที่แสดงให้เห็นถึงความแตกต่างของการใช้ Double quote และ Single quote

raw\_str = r"Python\tJava\tPHP\n"
print(raw\_str)

ในภาษา Python มีอีกวิธีหนึ่งในการประกาศตัวแปร String คือการกำหนดค่าแบบ raw string ในตัวอย่างข้างบน เราใช้ตัวอักษร r ก่อนหน้าเครื่องหมาย ' หรือ " ซึ่ง raw string เป็นการแปลง String ให้เหมือนกับที่กำหนดใน String literal เสมอ

นอกจากนี้ เรายังสามารถประกาศ String แบบหลายบรรทัดได้ โดยการครอบด้วยเครื่องหมาย """ หรือ ''' String literal ที่ปรากฎภายในเครื่องหมายนี้จะเป็น raw string ที่จะทำการ Escape ให้อัตโนมัติ มาดูตัวอย่าง

str = """\
HTTP response code
 200 Success
 404 Not found
 503 Service unavailable
"""

print(str)

ในตัวอย่าง เป็นการประกาศตัวแปร String แบบการใช้หลายบรรทัด String literal ทั้งหมดนั้นจะต้องอยู่ภายใน """...""" เนื่องจากเราต้องการจัดรูปแบบของ String ให้สวยงาม ดังนั้นเราจึงต้อง Newline ลงมาเขียนในบรรทัดถัดมา ดังนั้นเราสามารถใช้เครื่องหมาย \ เพื่อลบ Newline ออกไปได้ นี่เป็นผลลัพธ์ของโปรแกรม

HTTP response code
 200 Success
 404 Not found
 503 Service unavailable

**String concatenation**

เช่นเดียวกับในภาษาอื่นๆ ในภาษา Python คุณสามารถเชื่อมต่อตั้งแต่สอง String เข้าด้วยกันได้ โดยการใช้เครื่องหมายบวก (+) ถึงแม้ว่านี่เป็นตัวดำเนินการทางคณิตศาสตร์ที่ใช้สำหรับบวกค่าของตัวเลข แต่เมื่อนำมาใช้กับ String จะเป็นการเชื่อมต่อ String เข้าด้วยกันแทนหรือเรียกว่า *Operator overloading* มาดูตัวอย่างการเชื่อมต่อ String

first\_name = 'Matt'
last\_name = 'Williams'
full\_name = first\_name + ' ' + last\_name
bless = 'Merry' + 'Christmas!'

print(full\_name)
print(bless)

ในตัวอย่าง เป็นการเชื่อมต่อ String ในภาษา Python เรามีตัวแปร first\_name สำหรับเก็บชื่อ และตัวแปร last\_name สำหรับเก็บนามสกุล และเราสร้างตัวแปร full\_name เพื่อเก็บชื่อแบบเต็มโดยการนำ String จากทั้งสองตัวแปรก่อนหน้ามาเชื่อมต่อกัน ซึ่งเป็นการเชื่อมต่อ String จากตัวแปร และในตัวแปร bless เป็นการเชื่อมต่อจาก String literal โดยตรงเพื่อสร้างคำอวยพรในวันคริสมาสต์ นี่เป็นผลลัพธ์การทำงานของโปรแกรม

Matt Williams
MerryChristmas!

ในภาษา Python คุณยังสามารถเชื่อมต่อ String ได้ด้วยวิธีอื่นอีก เช่น White-space Tab หรือ Newline โดย String literal ที่คั่นด้วยช่องว่างในบรรทัดเดียวกันจะถูกนำมาต่อกันอัตโนมัติโดยตัวแปรของภาษา และคุณสามารถเชื่อมต่อโดยที่ String อยู่คนละบรรทัดได้ด้วยการใช้ \ต่อท้ายสำหรับบอกว่าคำสั่งมีอยู่ในบรรทัดต่อไป มาดูตัวอย่าง

name = 'Matt' 'Williams'
my\_number = 'One '\
'Two '\
'Three '

print(name)
print(my\_number)

ในตัวอย่าง เป็นการเชื่อมต่อ String โดยการใช้ช่องว่างและการขึ้นบรรทัดใหม่ และในการเขียนโปรแกรม เราแนะนำให้คุณใช้วิธีการใช้เครื่องหมาย + เพื่อให้การเขียนโปรแกรมเป็นไปในรูปแบบเดียวกัน อย่างไรก็ตาม คุณสามารถเลือกใช้แบบต่างๆ ได้ตามความเหมาะสมของโปรแกรม

MattWilliams
One Two Three

นี่เป็นผลลัพธ์การทำงานของโปรแกรมในการเชื่อมต่อ String เข้าด้วยกัน

**Charterers of string**

เนื่องจาก String นั้นเกิดจากตัวอักษรหลายๆ ตัวต่อกันจนเกิดเป็น String หรือกล่าวอีกนัยหนึ่งก็คือ String คืออาเรย์ของตัวอักษร ดังนั้นเราจึงสามารถเข้าถึงตัวอักษรตำแหน่งต่างๆ ของ String ได้ผ่านทาง Index ของมัน เช่นเดียวกับการเข้าถึงข้อมูลใน List มาดูตัวอย่าง

s = 'Mountain'
print(s[0]) # M
print(s[4]) # t
print(s[7]) # n

ในตัวอย่าง เป็นการเข้าถึงตำแหน่งของตัวอักษรในตัวแปร String s เราสามารถเข้าถึงด้วยการใช้ Index ในเครื่องหมาย [] โดยที่ Index ของ String นั้นจะเริ่มจาก 0 สำหรับตำแหน่งแรก และเพิ่มขึ้นทีละ 1 ไปจนถึงตำแหน่งสุดท้าย

s = 'Mountain'
s[0] = 'a' # invalid

อย่างไรก็ตาม การทำงานกับ String ผ่านทาง Index นั้นสามารถอ่านค่าได้เพียงอย่างเดียว และไม่สามารถแก้ไขค่าได้ จากตัวอย่างข้างบนจึงทำให้เกิดข้อผิดพลาดขึ้น เพราะเราพยายามที่จะเปลี่ยนแปลงค่าของ String ผ่านทาง Index ของมัน

s = 'Python'
for c in s:
 print(c)

print()

s = 'marcuscode.com'
for i in range(len(s)):
 print('s[%d] = %c' % (i, s[i]))

print()

เนื่องจาก String นั้นเป็นอาเรย์ของตัวอักษร ดังนั้นมันจึงสะดวกที่จะใช้คำสั่ง For loop วนอ่านค่าจากใน String ทีละตัวอักษร ในตัวอย่างข้างบน เป็นการวนอ่านค่าโดยการใช้คำสั่ง For วนอ่านค่าใน String สำหรับลูปแรก และในลูปที่สอง เราใช้คำสั่ง For สร้าง Index ตั้งแต่ 0 ถึงตำแหน่งสุดท้ายของ String และวนอ่านค่า

P
y
t
h
o
n

s[0] = m
s[1] = a
s[2] = r
s[3] = c
s[4] = u
s[5] = s
s[6] = c
s[7] = o
s[8] = d
s[9] = e
s[10] = .
s[11] = c
s[12] = o
s[13] = m

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

นอกจากนี้ ในภาษา Python นั้นยังสนับสนุนการตัดคำใน String ด้วย Index โดยการตัดเอาส่วนย่อยๆ ภายใน String มาดูตัวอย่างการใช้งาน การตัดคำใน String

s1 = 'Mountain'
print(s1[0:4]) # Moun
print(s1[4:6]) # ta
print(s1[0:1]) # M

s2 = 'marcuscode'
print(s2[:6]) #marcus
print(s2[6:]) #code

ในตัวอย่าง เป็นการตัดคำด้วย Index ในการใช้งานนั้นจะมีรูปแบบเป็น [start:end] โดยที่ start นั้นเป็นตำแหน่งของ Index เริ่มต้นที่ต้องการ และ end นั้นเป็นตำแหน่งก่อนหน้าตำแหน่งสุดท้ายของตัวอักษรที่ต้องการ เหมือนในตัวอย่างของการตัดคำในตัวแปร s1

และถ้าหากคุณปล่อยให้ตำแหน่ง start ให้ว่าง นั้นหมายความว่าเป็นการเริ่มมาจากตำแหน่งแรกของ String หรือหากคุณปล่อยให้ end ให้ว่าง เป็นการตัดจากตำแหน่งที่กำหนดไปจนสิ้นสุด String เหมือนในตัวแปร s2

Moun
ta
M
marcus
code

นี่เป็นผลลัพธ์การทำงานของของโปรแกรมในการตัด String ด้วย Index

และนอกจากนี้ เรายังสามารถหาความยาวของ String โดยการใช้ฟังก์ชันที่มากับภาษา Python คือฟังก์ชัน len() สำหรับหาขนาดของตัวอักษรของ String ที่ระบุ ซึ่งจะได้ผลลัพธ์กลับมาเป็น Integer นี่เป็นตัวอย่างการหาความยาวของ String

s1 = 'Mountain'
s2 = 'marcuscode'
s3 = 'Python'

print('length of s1 =', len(s1))
print('length of s2 =', len(s2))
print('length of s3 =', len(s3))

ในตัวอย่าง เป็นการใช้ฟังก์ชัน len() สำหรับหาความยาวของตัวแปร String สามตัวแปร และแสดงผลความยาวนั้นออกทาง console ซึ่งจะได้ผลลัพธ์ดังข้างล่าง

length of s1 = 8
length of s2 = 10
length of s3 = 6

ในบทนี้ คุณได้เรียนรู้เกี่ยวกับข้อมูลประเภท String ในภาษา Python การประกาศและใช้งานตัวแปรในรูปแบบต่างๆ รวมถึงการเชื่อมต่อ String เข้าด้วยกัน นอกจากนี้ คุณยังทราบว่าเราสามารถเข้าถึงตัวอักษรใน String ผ่านทาง Index ของมันและเราพูดถึงฟังก์ชันในการหาความยาวของ String ในบทต่อไป จะเป็นการใช้งานฟังก์ชันมาตรฐานในภาษา Python เพื่อจัดการกับ String