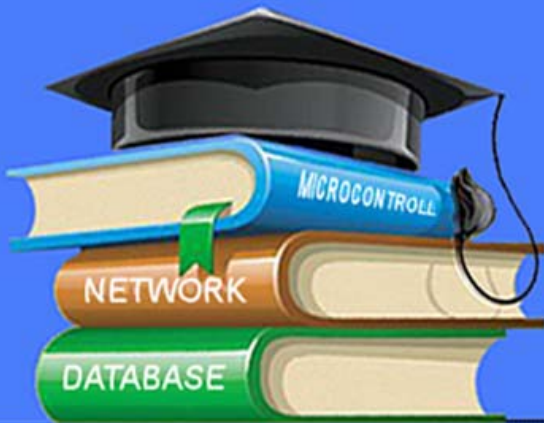


Chapter 10: Python String



Computer Engineering, Kasetsart University Kamphaeng Sean Campus

Outline

- ◆ Overview
- ◆ String Compare
- ◆ String Function & Method

Python String

- ◆ String คือ สายอักขระที่อยู่ภายในเครื่องหมาย(" ") หรือ (' ') เรียงต่อกันเป็นข้อความ ซึ่งตัวแปรสตริงนั้นมีลักษณะคล้ายกับลิสต์ที่บรรจุตัวอักษรไว้ในแต่ละ element

```
str1="Hello"
```

```
str2="Ex@mp!e"
```

```
str3='123'
```

'H'	'e'	'l'	'l'	'o'
-----	-----	-----	-----	-----

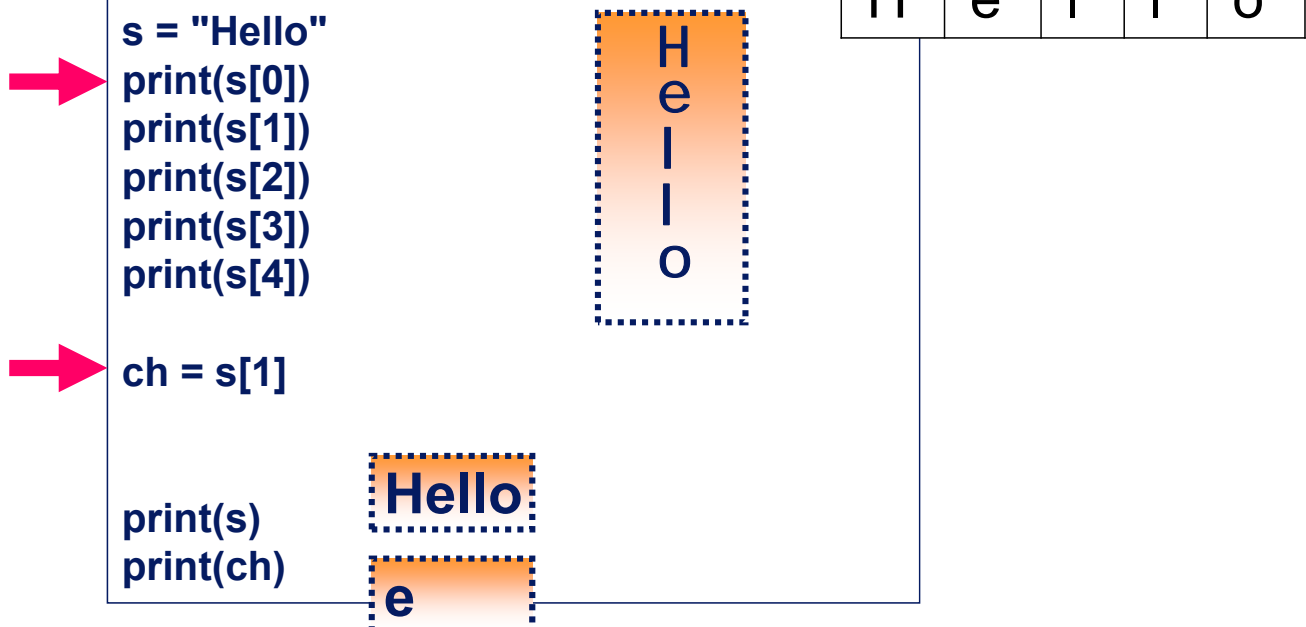
'E'	'x'	'@'	'm'	'p'	'l'	'e'
-----	-----	-----	-----	-----	-----	-----

'1'	'2'	'3'
-----	-----	-----

การอ้างถึงตัวอักษรใน string

- ◆ การอ้างถึงตัวอักขระใน String ทำได้โดยระบุหมายเลขของตัวอักขระ (Index) ซึ่งอักขระตัวแรกเริ่มต้นที่หมายเลข 0

- ◆ ตัวอย่าง



การรับค่า **string**

◆ การรับค่า string โดยคำสั่ง **input**

→ `s = input()`

→ `s = input("Enter your full name : ")`
`print("Hello," + " " + s)` **Kim**
Hello Kim

ใน Python เมื่อใช้ `input()` รับข้อความ โปรแกรมจะแยกข้อความแต่เครื่องหมายขึ้นบรรทัดใหม่ (`\n`)

ทำให้ไม่สามารถรับข้อความที่มีหลายบรรทัดได้ในครั้งเดียว

String Concatenation

◆ การต่อ String (String Concatenation) ทำได้โดยใช้เครื่องหมาย +

◆ ตัวอย่าง

→ `s = "Hello,"`
`t = "World"`
`print(s + " " + t)` **Hello, World**

→ `name = "Harry"`
`surname = "Potter"`
`stdname = name + " " + surname`
`print(stdname)` **Harry Potter**

Python String

◆ การหาความยาวของ string ทำได้โดยใช้ฟังก์ชัน len()

◆ ตัวอย่าง

0	1	2	3	4
'H'	'e'	'l'	'l'	'o'

```
s = "Hello"
```

```
print(len(s))
```

5

```
for i in range(len(s)-1):  
    print(s[i] , end = '-')
```

H-e-l-l-

```
print(s[len(s)-1])
```

Practice1

◆ จงเขียนโปรแกรมรับข้อความจากผู้ใช้ แล้วแสดงข้อความแบบย้อนกลับ เช่น Hello ได้ output เป็น olleH

String Compare

◆ ตัวดำเนินการสัมพันธ์ `==`, `!=`, `>`, `>=`, `<` และ `<=` สามารถนำมาใช้ในการเปรียบเทียบข้อความได้ โดย Python จะเปรียบเทียบข้อความทีละตัวอักษร โดยใช้ตารางรหัส ASCII

◆ ตัวอย่าง

S	c	a	r		
T	c	a	r	g	o

S < T (True or False ?)

T	c	a	r	g	o
U	c	a	t		

T > U (True or False ?)

S1	a	i	d
S2	b	e	

S1 >= S2 (True or False ?)

Practice2

◆ แสดงผลการทำงานของโปรแกรมต่อไปนี้

```
s1 = ""
s2 = "HaHaHa"
while(s1 != s2):
    s1 += "Ha"

print(s1)
```

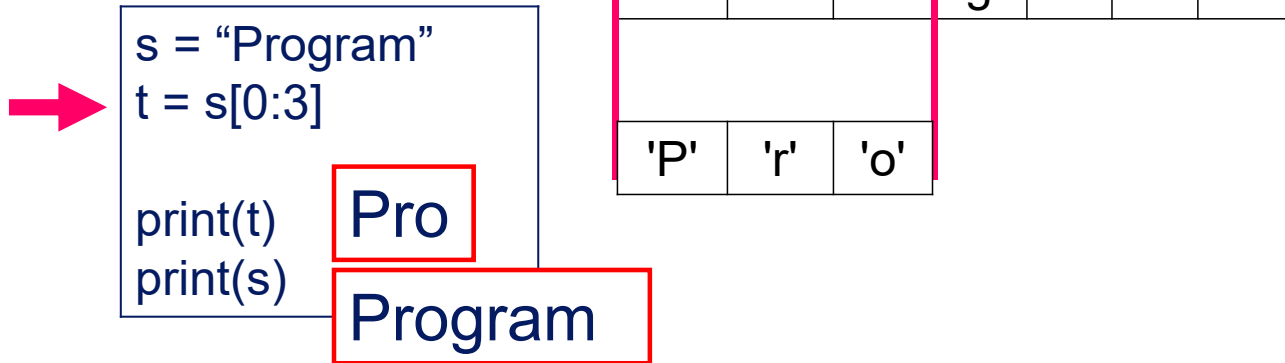
String substr

◆ การคัดลอกข้อความย่อยของ string

◆ รูปแบบ (syntax)

ตัวแปรชนิด string[ดัชนีเริ่มต้น:ดัชนีสิ้นสุด]

◆ ตัวอย่าง



String Method

◆ find() การหาค้นหาข้อความย่อย

◆ replace() การแทนที่ข้อความย่อย

◆ split() การแบ่งข้อความ

String Method: find()

◆ find() ค้นหาข้อความย่อยในข้อความหลัก ได้ผลลัพธ์เป็นตำแหน่งข้อความย่อยที่พบ และมีค่า -1 ถ้าไม่พบข้อความย่อย

◆ รูปแบบ (syntax)

ตัวแปรชนิด string.find(ข้อความย่อย, ตำแหน่งที่ต้องการเริ่มค้นหา)

◆ ตัวอย่าง

```
a = "cat"
s = "application"
pos = s.find(a, 0)
print(pos)
print(s.find(a, 6))
```

0	1	2	3	4	5	6	7	8	9	10
a	p	p	l	i	c	a	t	i	o	n

String Method: find()

◆ ตัวอย่าง

```
b = "rat"
s = "application"
pos = s.find(b)
print(pos)
```

-1

```
if (pos != -1):
    print("b is contained in s.")
else:
    print("b is not contained in s.")
```

b is not contained in s.

String Method: replace()

◆ การแทนที่ข้อความย่อยใน string

◆ รูปแบบ (syntax)

ตัวแปรชนิด string.replace(คำที่ต้องการหา,คำที่ต้องการแทนที่)

◆ ตัวอย่าง

```
s = "abcdef"
```

```
s = s.replace("bc","dd")
```

```
print(s)
```

0	1	2	3	4	5
'a'	'b'	'c'	'd'	'e'	'f'
	'd'	'd'			

adddef

String Method: replace()

◆ นอกจากนี้เมทอด replace ยังสามารถใช้ในการลบข้อความย่อยใน string ได้ด้วยการแทนที่คำที่ต้องการลบด้วย "" (ข้อความว่าง)

◆ ตัวอย่าง

```
s = "abcdef"
```

```
s = s.replace("bc","")
```

```
print(s)
```

'a'	'b'	'c'	'd'	'e'	'f'
-----	-----	-----	-----	-----	-----

'a'	'd'	'e'	'f'
-----	-----	-----	-----

adef

String Method: split()

◆ split() เป็นเมทอดที่ใช้สำหรับการแตก-แยกข้อความด้วยช่องว่างหรือสัญลักษณ์อื่นๆ ตามที่เราต้องการและส่งผลลัพธ์กลับเป็นลิสต์

◆ รูปแบบ (syntax)

ตัวแปรชนิด string.split(สัญลักษณ์ที่ใช้แยกข้อความ)

หากไม่ได้ใส่สัญลักษณ์ใดเมทอด split() จะแยกข้อความด้วย เว้นวรรค " "

String Method: split()

◆ ตัวอย่าง

```
s = "Hello World"  
s = s.split()  
  
print(s)
```

['Hello' , ' World']

◆ ตัวอย่าง

```
s = "Hello,World"  
s1,s2 = s.split(",")  
  
print(s1 + "-" + s2)
```

Hello-World

Review: Python string

การกระทำ	ฟังก์ชัน/ตัวดำเนินการ	ตัวอย่าง
การเข้าถึงแต่ละตัวอักษร	[]	S = "My name is Ann" Print(S[3],S[7],S[11])
การต่อข้อความ	+, +=	S = "he" T = "s" print(T+S) S += T
การเปรียบเทียบข้อความ	==, !=, <, <=, >, >=	if (S == T): print("S and T are the same.")
การหาความยาวข้อความ	len(string)	S = "LENGTH" print(len(s))

Review: Python string

การกระทำ	ฟังก์ชัน/ตัวดำเนินการ	ตัวอย่าง
การหาค้นหาข้อความย่อย	string.find(str, position)	S = "application" print(S.find("cat", 3))
การคัดลอกข้อความย่อย	string[start: end]	S = "application" print(S[5:10])
การลบข้อความย่อย	string.replace(str, "")	S = "ABCDEF"; S. replace(BCD, "")
การแทนที่ข้อความ	string.replace(str1, str2)	S = "ABCDEF" S.replace("EF", "CBA")
การแบ่งข้อความด้วยสัญลักษณ์	string.split() หรือ string.split(symbol)	s = "Hello,World" s = s.split(",")

Practice3

```
str1 = "Programming and Problem Solving"  
str2 = "gram"
```

Function Call	Result
str1.find("and")	
str1.find("Pro")	
str2.find("and")	
str1.find("Pro", 10)	
str1[3:7] == str2	
str1.replace("a", "A")	
str1.replace("and", "")	

Practice4

กำหนดให้มีการประกาศตัวแปรต่อไปนี้

```
str1 = "I am a teacher"
```

```
str2 = "You are my student"
```

- ◆ นิสิตจะต้องใช้คำสั่งใดเมื่อต้องการทราบว่า str1 และ str2 มีค่าเท่ากันหรือไม่
- ◆ หาตำแหน่งของ teacher ใน str1
- ◆ เปลี่ยน str2 จาก "You are my student" เป็น "You_are_my_student"
- ◆ ทราบตำแหน่งของ 'a' ตัวแรกใน str1

Practice4

- ◆ กำหนดให้มีการประกาศตัวแปรต่อไปนี้

```
str1 = "I am a teacher"
```

```
str2 = "You_are_my_student"
```

- ◆ ลบ "Yo" จาก str2

- ◆ นำข้อความ "my" จาก str2 ไปเก็บในตัวแปร str3

- ◆ เพิ่มข้อความ " dear!" ไว้ท้าย str3 เพื่อให้ str3 มีค่าเป็น "my dear!"

- ◆ แยกคำที่อยู่ใน str2 ทั้งหมด 4 คำใส่ตัวแปร a,b,c,d

Thank you!
Any questions?

