

Basic String & List Operations

ภาควิชาวิศวกรรมคอมพิวเตอร์

จุฬาลงกรณ์มหาวิทยาลัย

๒๕๖๒

เรื่องที่ต้องรู้

- Length, Concatenation, Repetition
- Indexing and slicing
- String splitting

String and List

- String: ลำดับของอักขระ
 - "ABCDEF"
- List: รายการของข้อมูล
 - [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
 - ["MO", "TU", "WE", "TH", "FR", "SA", "SU"]
- Operations
 - ความยาว `len("ABC")` `len([1,2,3])`
 - การต่อ `"A"+"BC"` `[1] + [2,3]`
 - การต่อกันซ้ำ ๆ `"A" * 3` `[0,0] * 3`
 - Indexing `x[k]`
 - Slicing `x[start : stop : step]`

String & List: Length, Concatenation, Repetition

```
print(len("123"), len([1,2,3]))  
H = "HBD"  
T = "2U"  
HBD = (H + T)*2 + " " + H*2 + " " + H+T  
print(2*(HBD + " -- "))  
  
x = ([0,1]*2 + [3,4])*3  
print(x)
```

program

```
3 3  
HBD2UHBD2U HBDHBD HBD2U -- HBD2UHBD2U HBDHBD HBD2U --  
[0, 1, 0, 1, 3, 4, 0, 1, 0, 1, 3, 4, 0, 1, 0, 1, 3, 4]
```

output

String: Indexing

s = "HELLO World"

index	0	1	2	3	4	5	6	7	8	9	10
	H	E	L	L	O		W	o	r	l	d
index	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

สตริงขนาด N ตัวอักษร มี index ตั้งแต่ -N ถึง N - 1

s[0] คือ "H"
s[1] คือ "E"
s[10] คือ "d"

s[-1] คือ "d"
s[-2] คือ "l"
s[-11] คือ "H"



s[**index นอกช่วง**] **แจ้งเตือน**
เช่น s[11] s[-12]

String: Slicing

s = "HELLO World"

index	0	1	2	3	4	5	6	7	8	9	10
	H	E	L	L	O		W	o	r	l	d
	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

s[start:stop:step]

รวมตัวนี้   ไม่รวมตัวนี้

s[0:5:1]	คือ	"HELLO"
s[1:11:2]	คือ	"EL ol"
s[10:5:-1]	คือ	"dlroW"
s[-1:-11:-2]	คือ	"drWOL"

s[50:99:1] หรือ s[-12:-99:-1] ไม่เจ้ง ได้ ""

String: Slicing

s = "HELLO World"

index	0	1	2	3	4	5	6	7	8	9	10
	H	E	L	L	O		W	o	r	l	d
index	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

s[start:stop:step]

ไม่ใช่ start ถือว่า 0
ไม่ใช่ stop ถือว่า len+1
ไม่ใช่ step ถือว่า 1

ถ้า step ติดลบ
ไม่ใช่ start ถือว่า -1
ไม่ใช่ stop ถือว่า -(len+1)

s[0:11:1] เหมือน s[:11:1] เหมือน s[0::1]
 เหมือน s[::1] เหมือน s[::]
 เหมือน s[:] เหมือน s

s[-1:-12:-1] เหมือน s[:-12:-1] เหมือน s[::-1]

ข้อควรระวัง: แก้ไขข้อมูลในสตริงไม่ได้

```
s = "Python"
```

```
s[0] = "J"      # ผิด
```



```
s = "Python"
```

```
s = "J" + s[1:]  # ได้ เป็นการสร้างสตริงใหม่
```



ตัวอย่าง: รหัสนิสิต คณะอะไร เข้าปีอะไร ระดับใด

```
stu_id = input()
print("Student ID:", stu_id)
print("Faculty code:", stu_id[-2:])
print("Enrollment year:", "25" + stu_id[:2])
print("Academic degree:", stu_id[2])
```

program

หลังจากสั่งทำงานแล้วป้อน 6230012021 จะได้ผลเป็น

```
Student ID: 6230012021
Faculty code: 21
Enrollment year: 2562
Academic degree: 3
```

output

รับเลขประจำตัวเป็นสตริง
เลือกหลักต่าง ๆ ได้ง่ายกว่ารับเป็นจำนวนเต็ม

แบบฝึกหัด: เลขประจำตัวบัตรประชาชน

$$n_{12} = \left(11 - \left(\sum_{k=0}^{11} (13 - k)n_k \right) \% 11 \right) \% 10$$

k =	0	1	2	3	4	5	6	7	8	9	10	11	12
	3	7	2	0	8	0	1	4	4	1	1	5	1

$$13-k = 13 \quad 12 \quad 11 \quad 10 \quad 9 \quad 8 \quad 7 \quad 6 \quad 5 \quad 4 \quad 3 \quad 2$$

$$13 \times 3 + 12 \times 7 + 11 \times 2 + 10 \times 0 + 9 \times 8 + 8 \times 0 + 7 \times 1 + 6 \times 4 + 5 \times 4 + 4 \times 1 + 3 \times 1 + 2 \times 5 = 285$$
$$(11 - 285 \% 11) \% 10 = 1$$

เขียนโปรแกรมรับเลขประจำตัวบัตรประชาชนมา 12 หลักทางซ้าย
แล้วคำนวณหลักขวาสุดด้วยสูตรข้างบน

List: Indexing & Slicing (เหมือนของ String)

```
#      0  1  2  3  4  5  6  7  8  9 10
x = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
# -11-10 -9 -8 -7 -6 -5 -4 -3 -2 -1

print("x =", x)
print(x[0], x[-1], x[:3], x[2:4])
print(x[-3:-1], x[1:-1])
print(x[::-1])
```

program

```
x = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
0 55 [0, 1, 1] [1, 2]
[21, 34] [1, 1, 2, 3, 5, 8, 13, 21, 34]
[55, 34, 21, 13, 8, 5, 3, 2, 1, 1, 0]
```

output

แก้ไขข้อมูลในลิสต์ได้ (แต่แก้ไขในสตริงไม่ได้)

```
x = [0]*7          # [0,0,0,0,0,0,0]
x[1] = 1           # [0,1,0,0,0,0,0]
x[2] = x[0] + x[1] # [0,1,1,0,0,0,0]
x[3] = x[1] + x[2] # [0,1,1,2,0,0,0]
x[4:7] = [3,5,8]   # [0,1,1,2,3,5,8]
```

ตัวอย่าง: เลขวันเป็นชื่อวัน

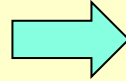
```
days_of_week = ["Monday", "Tuesday", "Wednesday",  
                "Thursday", "Friday", "Saturday",  
                "Sunday"]  
  
d = int(input())  
print( days_of_week[d - 1] )
```

1	→	Monday
2	→	Tuesday
3	→	Wednesday
4	→	Thursday
5	→	Friday
6	→	Saturday
7	→	Sunday

แบบฝึกหัด: รับจำนวนเต็ม แสดงคำอ่าน

Input

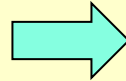
0



Output

zero

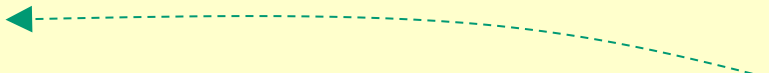
1



one

```
n = int(input())
```

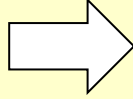
split แยกสตริงออกเป็นลิสต์

- **`x = s.split()`** 
 - แยกสตริง `s` ออกเป็นส่วน ๆ ได้เป็นลิสต์ (แยกด้วยช่องว่าง)
 - `s = "11 2 33"`
 - `s.split()` ได้ `["11", "2", "33"]`
- **`x = s.split(sep)`**
 - แยกสตริง `s` ออกเป็นส่วน ๆ ได้เป็นลิสต์ ใช้ `sep` เป็นตัวแยก
 - `s = "11:2: 33"`
 - `s.split(":")` ได้ `["11", "2", " 33"]`

```
"a,,,b".split(",")   ได้ ["a", "", "", "b"]  
"a  b".split(" ")   ได้ ["a", "", "", "b"]  
"a  b".split()       ได้ ["a", "b"]  
"a,,,b".split(",,") ได้ ["a", ",b"]
```

รูปแบบการรับข้อมูล

```
x = input()  
t = x.split()  
...
```



```
t = input().split()  
...
```

623102021 John Wick

```
t = input().split()  
student_id = t[0]  
first_name = t[1]  
last_name = t[2]
```

623102021,90,87,2.54

```
t = input().split(",")  
student_id = t[0]  
midterm = int(t[1])  
final = int(t[2])  
gpa = float(t[3])
```


เปลี่ยน วัน/เดือน/ปี ค.ศ. เป็น วัน/เดือน/ปี พ.ศ

```
inp = input()
t = inp.split("/") # "d/m/y" -> [d,m,y]
y = int(t[2]) + 543
out = t[0] + "/" + t[1] + "/" + str(y)
print( out )
```

program

หลังจากสั่งทำงานแล้วป้อน 5/12/2019 จะได้ผลเป็น

5/12/2562

output

"5/12/2019" → ["5", "12", "2019"]

ข้อสังเกต

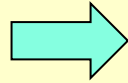
`t = x.split()` ทำไมไม่เขียน \rightarrow `t = split(x)`

- `split` ไม่ใช่ฟังก์ชันทั่วไป
 - `split` เป็นฟังก์ชันที่ใช้งานเฉพาะกับสตริง เรียกว่าเป็น *method* ของสตริง
 - สตริงมีหลายเมทอดให้ใช้
 - รูปแบบการใช้งาน: **สตริง . ชื่อเมทอด (...)**
 - `a = x.upper()` # สร้างสตริงใหม่ที่เป็นอังกฤษตัวใหญ่
 - `b = x.lower()` # สร้างสตริงใหม่ที่เป็นอังกฤษตัวเล็ก
 - `c = x.strip()` # สร้างสตริงใหม่ที่ลบช่องว่างซ้ายขวาออก
- *** ช่วงแรกนี้ยังไม่ได้ใช้ จะได้ศึกษาในบทถัด ๆ ไป ***

แบบฝึกหัด: รับ 20/5/2010 แสดง May 20, 2010

Input

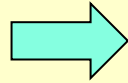
1/1/2019



Output

January 1, 2019

31/12/2020



December 31, 2020

