

## ฟังก์ชัน print() และการจัดรูปแบบ output

## คำอธิบาย

```

help(print)
Help on built-in function round in module builtins:
print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file: a file-like object (stream); defaults to the current sys.stdout.
    sep: string inserted between values, default a space.
    end: string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
    round(number[, ndigits]) -> number

```

## Formatted printing – การจัดรูปแบบ output มี 2 รูปแบบ ดังนี้

1. ใช้ string modulo operator (%) (อ้างอิงซีท ดร.เสกฐวิทย์ เกิดผล)

โดยสร้าง string ที่ประกอบด้วยค่าคงที่ในรูปแบบต่างๆ ที่ต้องการ พร้อมด้วย placeholder เพื่อใช้สำหรับจัดรูปแบบนิพจน์ ตามด้วยสัญลักษณ์ % และนิพจน์ที่จะถูกนำมาเติมเต็มในแต่ละ placeholder ตามลำดับจากซ้ายไปขวา ก่อนนำไปแสดงผลด้วยฟังก์ชัน print()

รูปแบบของการจัดรูปแบบให้กับ placeholder

```
%[flags][width][.precision]type
```

## เมื่อ type ประกอบด้วย

Type	ความหมาย
d	Signed integer decimal.
i	Signed integer decimal.
o	Unsigned octal.
u	Unsigned decimal.
x	Unsigned hexadecimal (lowercase).
X	Unsigned hexadecimal (uppercase).
e	Floating point exponential format (lowercase).
E	Floating point exponential format (uppercase).
f	Floating point decimal format.
F	Floating point decimal format.
g	Same as "e" if exponent is greater than -4 or less than precision, "f" otherwise.
G	Same as "E" if exponent is greater than -4 or less than precision, "F" otherwise.
%	No argument is converted, results in a "%" character in the result.

## และ flag ประกอบด้วย

Flag	ความหมาย
#	Used with o, x or X specifiers the value is preceded with 0, 0o, 0O, 0x or 0X respectively.
0	The conversion result will be zero padded for numeric values.

Flag	ความหมาย
-	The converted value is left adjusted
	If no sign (minus sign e.g.) is going to be written, a blank space is inserted before the value.
+	A sign character ("+" or "-") will precede the conversion (overrides a "space" flag).

**แบบฝึกหัดที่ 1** จงพิจารณาว่าฟังก์ชัน print() ต่อไปนี้จะแสดงผลอย่างไร

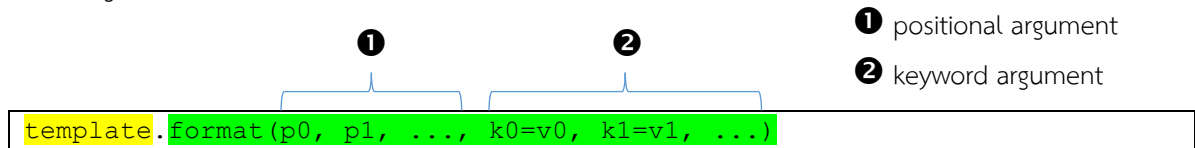
```
p = 12.5
q = 6
print(q, p, p*q, sep=",")
print(q, p, p*q, sep=" :-) ")
print(str(q) + " " + str(p) + " " + str(p*q))
print(p, p*q, "\n")

print("Pencil: %d\tPrice: %5.2f" % (32, 15.75))
print("Pencil: %d      Price: %5.2f" % (32, 15.75))
print("%-12s: %5d  Price: %5.2f" % ("Pencil", 32, 15.75))
print("%-12s: %5d  Price: %5.2f" % ("Ruler", 120, 8.25))
print("%-12s: %5d  Price: %5.2f" % ("Pen", 65, 25))

s = "Price: $ %8.2f" % (356.08977)
print(s)

print("%10.3e" % (356.08977))
print("%10.3E" % (356.08977))
print("%6.2f" % (23.789))
print("%6.2f" % (0.039))
print("%6.2f" % (199.8))
print("%6.2f" % (23))
print("%6.2f" % (2324.17))
print("Only one percentage sign: %" % ())
```

## 2. ใช้ string method format



- **template** เป็น string ที่ประกอบด้วยค่าคงที่ในรูปแบบต่างๆ ที่ต้องการ พร้อมด้วย placeholder ที่เขียนอยู่ระหว่างสัญลักษณ์ { และ } โดยนิพจน์ที่จะนำมาแทนแต่ละ placeholder มาจากอาร์กิวเมนต์ **p0, p1, ... k0=v0, k1=v1, ...** ของ method **format** ซึ่งอาจมีจำนวนตั้งแต่ 0 นิพจน์ขึ้นไป  
รหัสการจัดรูปแบบของแต่ละ placeholder (ถ้ามี) จะเขียนต่อจาก : โดยใช้รหัสการจัดรูปแบบที่ประกอบด้วย [flags][width][option][.precision]type ส่วนด้านหน้า : ใช้ระบุลำดับของ positional argument หรือระบุชื่อของ keyword argument ที่ถูกจัดรูปแบบ
- **p0, p1, ...** เรียกว่า positional argument เป็นนิพจน์ที่จะนำมาแทนใน placeholder โดยมีลำดับ (index) ในการอ้างอิงจากซ้ายไปขวา นั่นคือนิพจน์ p0 มีลำดับเป็น 0 และนิพจน์ถัดๆ มา มีลำดับเพิ่มขึ้นทีละ 1
- **k0=v0, k1=v1, ...** เรียกว่า keyword argument เป็นนิพจน์ที่จะนำมาแทนใน placeholder โดยใช้ชื่อในการอ้างอิงกำกับแต่ละนิพจน์ ในที่นี้ k0 เป็นชื่อของนิพจน์ v0

ถ้าต้องการเข้าถึง positional argument ตามลำดับที่เขียนเป็นอาร์กิวเมนต์ (นั่นคือ ตามลำดับจากซ้ายไปขวา) เราอาจเขียนได้ 2 รูปแบบ คือ ไม่ระบุเลขลำดับของนิพจน์ "{ } { } { }" หรือ ระบุลำดับของนิพจน์ "{0} {1} {2}" แต่ถ้าต้องการเข้าถึงในลำดับแบบอื่นให้ระบุลำดับตามที่ต้องการ เช่น "{1} {2} {0}"

ถ้าต้องการเข้าถึง keyword argument ใดก่อนหรือหลัง ให้ระบุชื่อ keyword argument นั้น ๆ ใน placeholder

### ตัวอย่าง

```
print("{:12s}: {:5d} Price: {:.2f}".format("Pencil", 32, 15.75))
print("{:12s}: {:5d} Price: {:.2f}".format("Ruler", 120, 8.25))
print("{:12s}: {:5d} Price: {:.2f}".format("Pen", 65, 25))
print("%12s" % ("Pencil"))
```

```
Pencil      :    32 Price: 15.75
Ruler       :   120 Price:  8.25
Pen         :    65 Price: 25.00
Pencil
```

\* สังเกต {:12s} พิมพ์ชิดซ้ายความกว้าง 12 ช่อง ขณะที่ %12s พิมพ์ชิดขวาความกว้าง 12 ช่อง

Option	ความหมาย
'<'	The field will be left-aligned within the available space. This is usually the default for strings.
'>'	The field will be right-aligned within the available space. This is the default for numbers.
'0'	If the width field is preceded by a zero ('0') character, sign-aware zero-padding for numeric types will be enabled.
','	This option signals the use of a comma for a thousand separator.

Option	ความหมาย
'='	Forces the padding to be placed after the sign (if any) but before the digits. This is used for printing fields in the form "+000000120". This alignment option is only valid for numeric types.
'^'	Forces the field to be centered within the available space.
'+'	Indicates that a sign should be used for both positive as well as negative <i>numbers</i> .
'-'	Indicates that a sign should be used only for negative <i>numbers</i> , which is the default behavior.
space	Indicates that a leading space should be used on positive <i>numbers</i> , and a minus sign on negative <i>numbers</i> .

## อ้างอิงข้อมูล:

[http://www.python-course.eu/python3\\_formatted\\_output.php](http://www.python-course.eu/python3_formatted_output.php)

## แบบฝึกหัดที่ 2 จงพิจารณาว่าฟังก์ชัน print() ต่อไปนี้จะแสดงผลอย่างไร

```
s1 = "First argument: {0}, second one: {1}".format(47, 11)
s2 = "Second argument: {1}, first one: {0}".format(47, 11)
s3 = "Second argument: {1:3d}, first one: {0:7.2f}".format(47.42, 11)
s4 = "First argument: {}, second one: {}".format(47, 11)
s5 = "various precisions: {0:6.2f} or {0:6.3f}".format(1.4148)
s6 = "Art: {a:5d}, Price: {p:8.2f}".format(a=453, p=59.058)
print(s1, s2, s3, s4, s5, s6, sep="\n")

print("The value is {:06d}".format(378))
print("The value is {:06d}".format(-378))

x = 78962324245
print("The value is {:,}".format(x))
x = 589765342
print("The value is {0:6,d}".format(x))
x = 5897653423.89676
print("The value is {0:12,.3f}".format(x))
```

