

Python 程式設計入門 (四)

函數

余家瑞 林口長庚醫院 檢驗醫學科

課程目標

- 能自行了解、操作內建函數、第三方模組的函數
- 了解如何定義簡單的函數

什麼是函數？

- 數學上的函數以 $f(x)$ 的形式出現
- 在Python中函數也是以 $f(x)$ 的形式存在，例如：
 - `print()`
 - `input()`
 - `str.strip()`
 - `str.just()`
 - `list.append()`
 - `list.extend()`

函數有什麼好處？

- 方便重複使用。
- 方便維護。

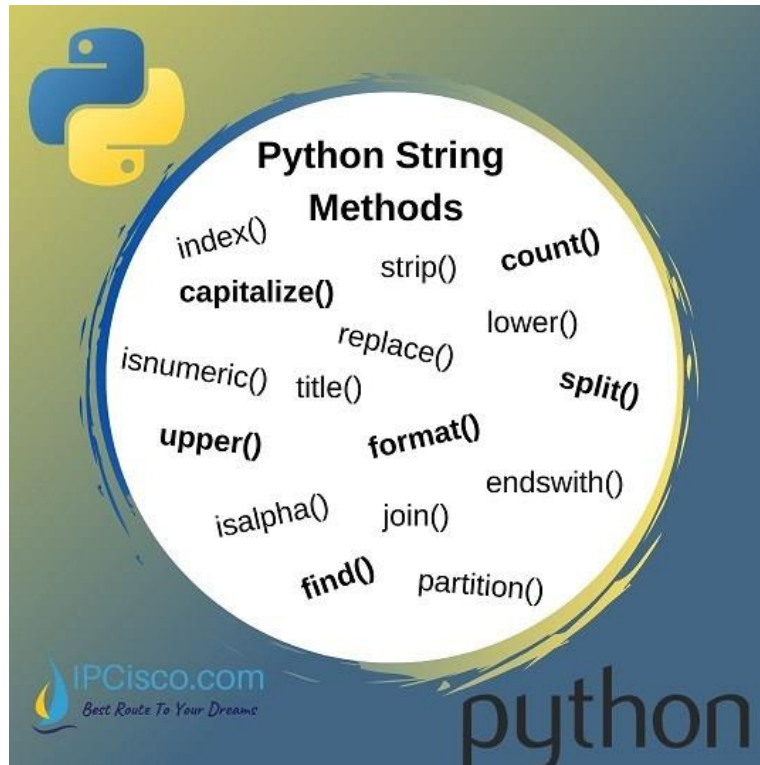
Build-in Function 內建函數

Built-in Functions			
A abs() aiter() all() any() anext() ascii()	E enumerate() eval() exec()	L len() list() locals()	R range() repr() reversed() round()
B bin() bool() breakpoint() bytearray() bytes()	F filter() float() format() frozenset()	M map() max() memoryview() min()	S set() setattr() slice() sorted() staticmethod() str() sum() super()
C callable() chr() classmethod() compile() complex()	G getattr() globals()	N next()	T tuple() type()
D delattr() dict() dir() divmod()	H hasattr() hash() help() hex()	O object() oct() open() ord()	V vars()
	I id() input() int() isinstance() issubclass() iter()	P pow() print() property()	Z zip() _import_()

常見的功能，可以直接呼叫使用。

如何理解沒看過的函數？

- 官方文檔：<https://docs.python.org/3/index.html>
- 其他中文網站



Parameter & Argument 參數與引數

- 引數和參數是一體兩面的事情。
- 參數是創建函數時，宣告程式所需的數值。
- 引數是使用函數時，提供給程式的數值。

Default parameter 預設參數

- 函數如果有預設值，則不需要每個參數都寫出。
- 無預設之參數在前，有預設之參數在後！
- `function(參數_1, 參數_2, 參數_3 = 預設值_3)`

引數傳入的兩種方式

- 位置引數(Positional Argument)：引數依照前後順序對應到參數
- 關鍵字引數(Keyword Argument)：引數依照標籤名字對應到參數

兩種傳遞方式可以同時並用，
規則是位置引數必須放在關鍵字引數前面。

範例：

- `str.split()`
- `str.startswith()`
- `str.replace()` 練習

如何自行定義函數？

- 冒號 & 縮排！
- 以方便他人理解的方式命名
- 以小寫命名

```
def function(參數_1, 參數_2, ..., 參數_n = 預設值_n):  
    函數內容
```

Return 回傳/返回

- 回傳值即是函數物件的賦值：

$$f(x) = \text{回傳值}$$

- 當程式運行到 `return` 時將會無條件直接結束！
- 一個函式也可以選擇不回傳結果。

Local / global variable 區域/全域變數

- 函式中宣告的變數為區域變數，有效範圍是整個函式，離開函式則無法存取。
- 相對而言，此前所使用的變數為全域變數，變數適用範圍為整個程式。

進階：函數中的 *args 和 **kwargs

```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

Print *objects* to the text stream *file*, separated by *sep* and followed by *end*. *sep*, *end*, *file*, and *flush*, if present, must be given as keyword arguments.

All non-keyword arguments are converted to strings like `str()` does and written to the stream, separated by *sep* and followed by *end*. Both *sep* and *end* must be strings; they can also be `None`, which means to use the default values. If no *objects* are given, `print()` will just write *end*.

The *file* argument must be an object with a `write(string)` method; if it is not present or `None`, `sys.stdout` will be used. Since printed arguments are converted to text strings, `print()` cannot be used with binary mode file objects. For these, use `file.write(...)` instead.

Whether the output is buffered is usually determined by *file*, but if the *flush* keyword argument is true, the stream is forcibly flushed.

Changed in version 3.3: Added the *flush* keyword argument.

* 和 ** 在Python中視為一種「解包」語法

*[a, b, c, d] \longleftrightarrow a, b, c, d

**{k1: v1, k2: v2} \longleftrightarrow k1=v1, k2=v2

<https://blog.csdn.net/be5yond>

* 解包tuple / list ; ** 解包dictionary 。

* 和 ** 運算子

- 定義函數時，用 * / ** 將多個 positional / keyword argument 一併傳入。
- 使用函數時，用 * / ** 將 tuple / dictionary 解開傳入。

Python 程式設計入門 (四)

函數

提問與回饋！