

APPENDIX

D

Jupyter Notebook 的神奇函數和快捷鍵

D-1 / Jupyter Notebook 的神奇函數

D-2 / Jupyter Notebook 的快捷鍵

D-1

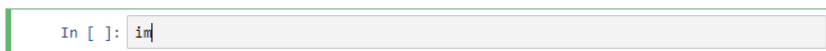
Jupyter Notebook 的神奇函數

Jupyter 這個名稱是三種程式語言所組成，即 Julia、Python 和 R 語言，Python 語言是指 IPython (Interactive Python)，這是一種命令列 Shell 來執行多種程式語言的互動運算。

事實上，Jupyter Notebook 就是源於 IPython Notebook，支援 IPython 提供的**神奇函數** (Magic Functions)，換句話說，所謂的 Jupyter Notebook 神奇函數，就是指 IPython 的神奇函數。

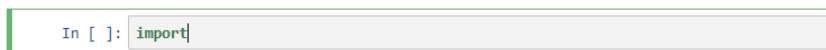
Tab 鍵：自動程式碼完成

在 Jupyter Notebook 的單元 (Cell) 輸入 Python 程式碼時，如果忘了關鍵字的英文拼法，我們可以按下 **Tab** 鍵來進行自動程式碼完成，例如：準備輸入 `import`，只需輸入 `im`，如下圖所示：



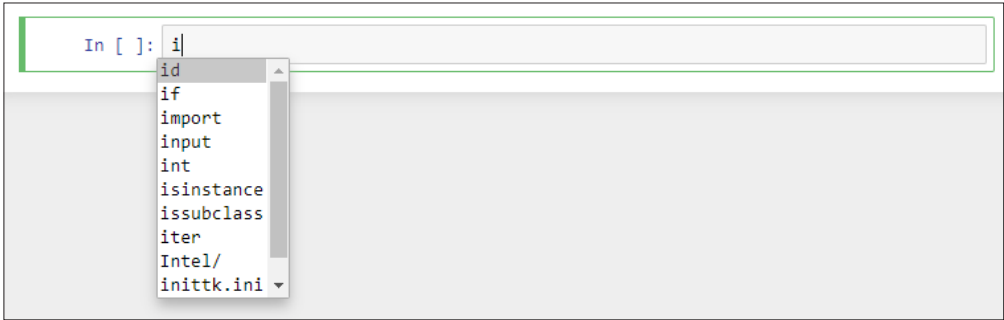
```
In [ ]: im
```

按 **Tab** 鍵，就可以馬上自動完成 `import`，如下圖所示：

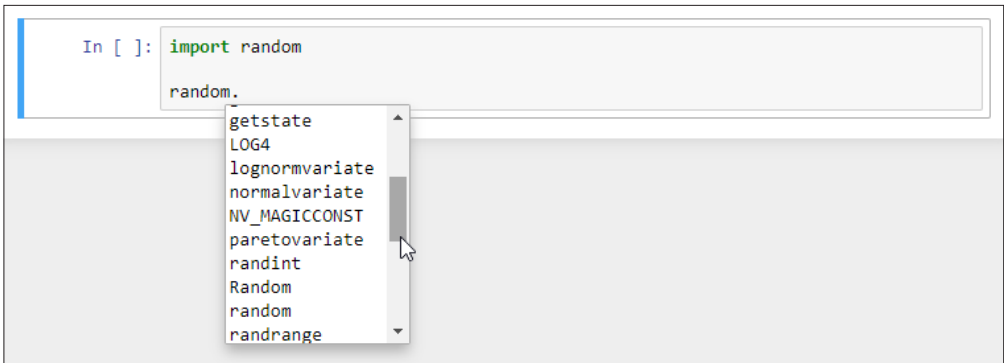


```
In [ ]: import
```

上述單元可以自動程式碼完成，因為「im」只有一種選擇，如果輸入「i」，按 **Tab** 鍵，因為有多種可能，所以顯示清單來選擇關鍵字，如下圖所示：

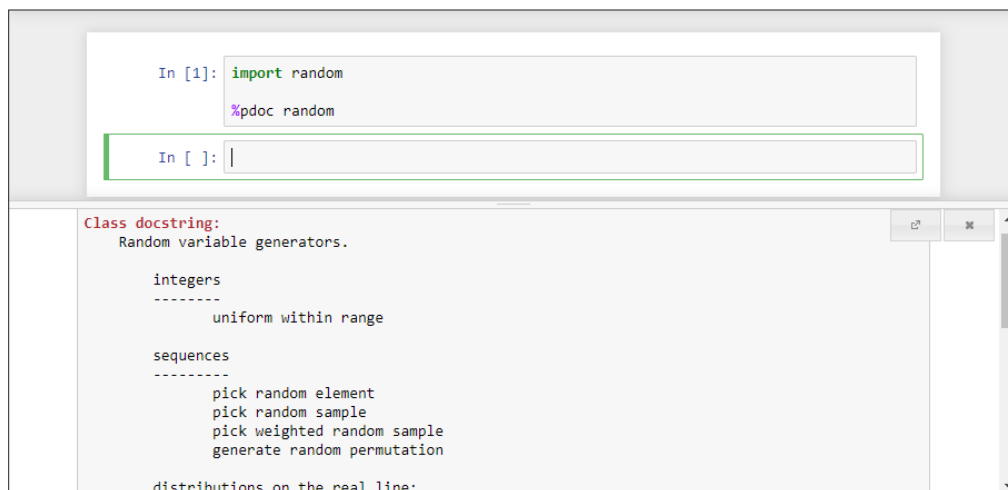


同樣的，我們可以使用 **Tab** 鍵查詢模組支援的函數，或物件支援的屬性和方法，例如：首先匯入 `random` 模組後，輸入「`random.`」，再按 **Tab** 鍵，可以看到清單顯示和選擇支援的函數，如下圖所示：



%pdoc：查詢模組和函數說明

在匯入模組後，我們可以使用 `%pdoc` 神奇函數加上模組名稱來查詢模組說明，例如：查詢 `random` 模組的說明，如下圖所示：



A screenshot of a Jupyter Notebook interface. The top part shows two input cells. The first cell contains the code `In [1]: import random` and `%pdoc random`. The second cell is empty and has a green border. Below the input cells, the output of the first cell is displayed. It shows the docstring for the `random` module, which includes a class docstring for `Random variable generators`, a list of integers, a list of sequences, and a list of distributions on the real line.

```
In [1]: import random
        %pdoc random

In [ ]: |

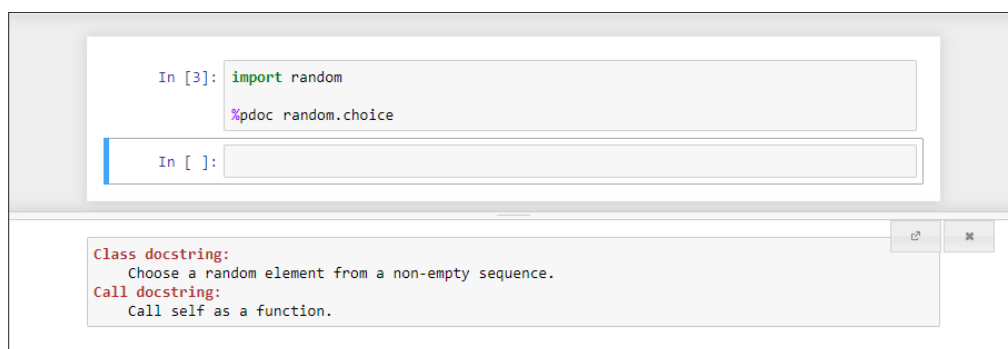
Class docstring:
    Random variable generators.

    integers
    -----
        uniform within range

    sequences
    -----
        pick random element
        pick random sample
        pick weighted random sample
        generate random permutation

    distributions on the real line:
```

同理，我們可以查詢 `random.choice` 函數的說明，如下圖所示：



A screenshot of a Jupyter Notebook interface. The top part shows two input cells. The first cell contains the code `In [3]: import random` and `%pdoc random.choice`. The second cell is empty and has a blue border. Below the input cells, the output of the first cell is displayed. It shows the docstring for the `random.choice` function, which includes a class docstring for `Choose a random element from a non-empty sequence` and a call docstring for `Call self as a function`.

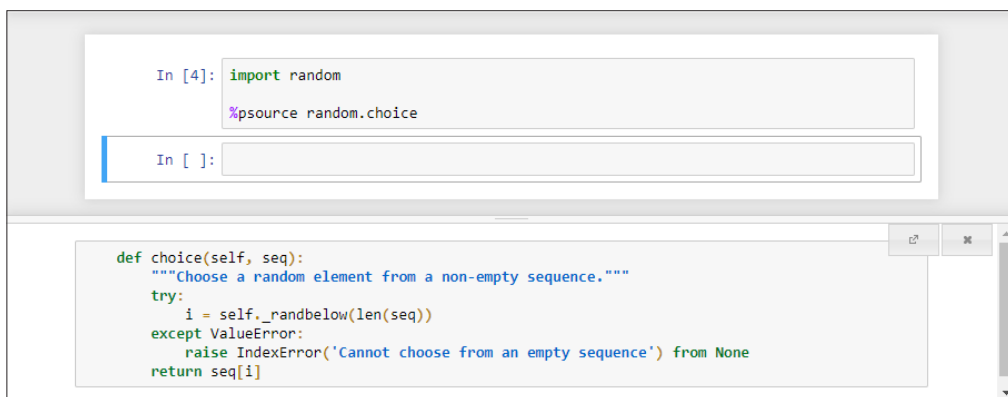
```
In [3]: import random
        %pdoc random.choice

In [ ]: |

Class docstring:
    Choose a random element from a non-empty sequence.
Call docstring:
    Call self as a function.
```

%psource：查詢模組和函數的原始程式碼

在匯入模組後，我們可以使用 `%psource` 神奇函數加上模組或函數名稱來查詢模組或函數的原始程式碼，例如：查詢 `random.choice` 函數的原始程式碼，如下圖所示：



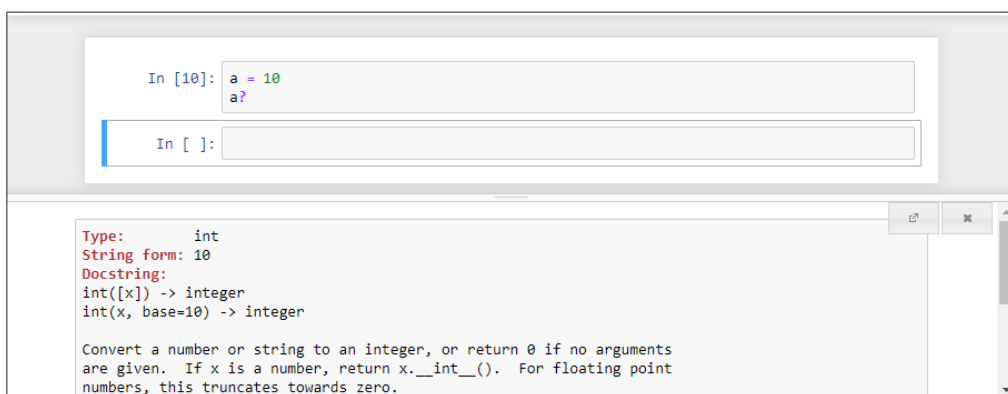
```
In [4]: import random
        %psource random.choice

In [ ]:
```

```
def choice(self, seq):
    """Choose a random element from a non-empty sequence."""
    try:
        i = self._randbelow(len(seq))
    except ValueError:
        raise IndexError('Cannot choose from an empty sequence') from None
    return seq[i]
```

「？」：查詢變數和函數說明

我們只需在變數和函數後加上「？」神奇函數即可查詢變數或函數說明，例如：在單元的 Python 程式碼指定變數 `a` 的值是 10 後，使用「`a?`」查詢變數 `a`，如下圖所示：



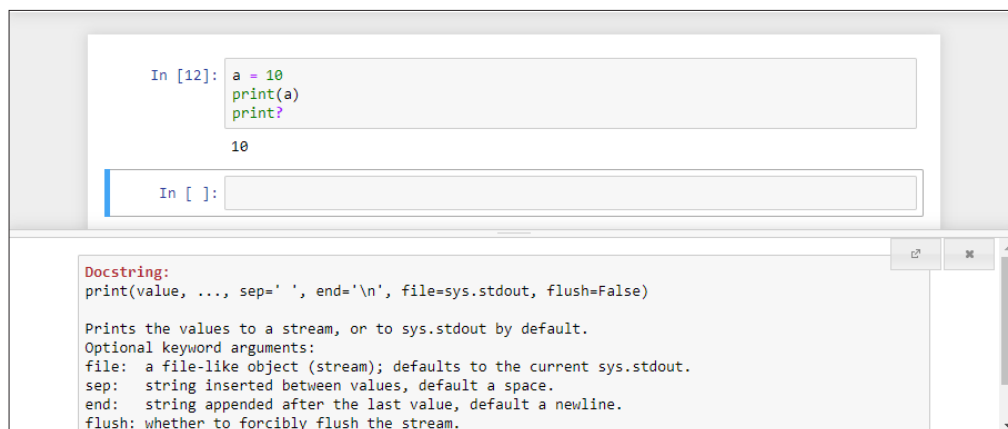
```
In [10]: a = 10
         a?

In [ ]:
```

```
Type:      int
String form: 10
Docstring:
int([x]) -> integer
int(x, base=10) -> integer

Convert a number or string to an integer, or return 0 if no arguments
are given. If x is a number, return x.__int__(). For floating point
numbers, this truncates towards zero.
```

同理，我們可以查詢 `print()` 函數，即「`print?`」，如下圖所示：



```
In [12]: a = 10
         print(a)
         print?

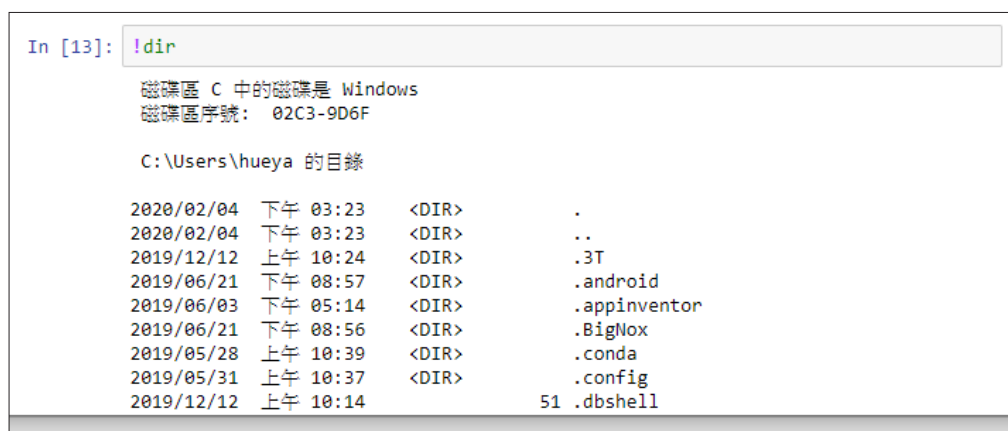
10

In [ ]:
```

Docstring:
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.

「!」：執行 DOS 指令

我們只需使用「!」神奇函數加上 DOS 指令，即可直接在 Jupyter Notebook 執行 DOS 指令，例如：使用「!`dir`」查詢根目錄的檔案和目錄清單，如下圖所示：



```
In [13]: !dir
```

磁碟區 C 中的磁碟是 Windows
磁碟區序號： 02C3-9D6F

C:\Users\hueya 的目錄

2020/02/04	下午 03:23	<DIR>	.
2020/02/04	下午 03:23	<DIR>	..
2019/12/12	上午 10:24	<DIR>	.3T
2019/06/21	下午 08:57	<DIR>	.android
2019/06/03	下午 05:14	<DIR>	.appinventor
2019/06/21	下午 08:56	<DIR>	.BigNox
2019/05/28	上午 10:39	<DIR>	.conda
2019/05/31	上午 10:37	<DIR>	.config
2019/12/12	上午 10:14		51 .dbshell

%run：執行 Python 程式檔

對於現存的 Python 程式檔，例如：第 3 章的 Ch3_3_2.py，我們可以使用 %run 神奇函數來執行 Python 程式檔，首先將 Python 程式檔案複製到「!dir」神奇函數查詢結果的根目錄，即可在 Jupyter Notebook 執行此 Python 程式檔案，如下圖所示：

```
In [17]: %run Ch3_3_2.py
[0 0] 0
[0 1] 0
[1 0] 0
[1 1] 1

In [ ]:
```

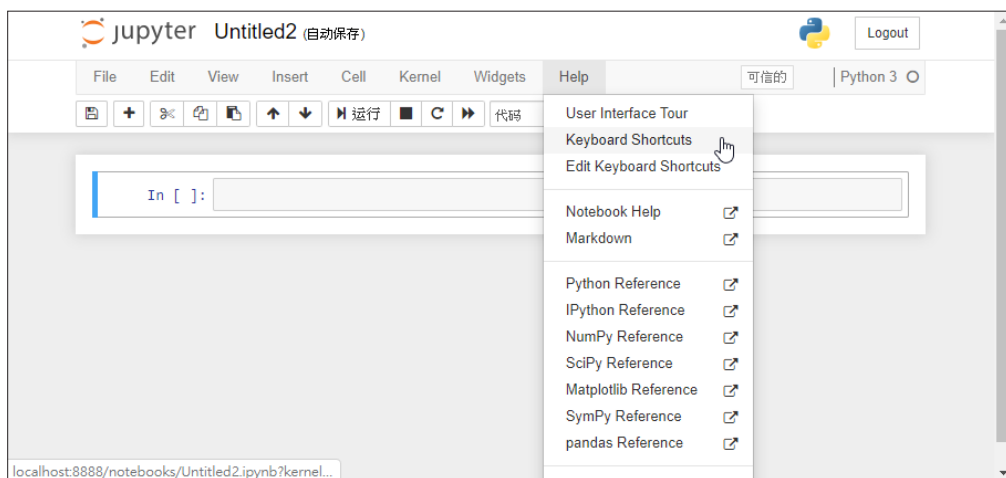
%magic 和 %quickref：檢視說明文件和快速查詢神奇函數

我們可以使用 %magic 神奇函數查詢 IPython 神奇函數的說明文件，%quickref 神奇函數快速查詢支援的神奇函數清單。

D-2

Jupyter Notebook 的快捷鍵

在 Jupyter Notebook 可以使用快捷鍵來幫助我們快速編輯程式碼和執行單元的程式碼，請執行「Help/Keyboard Shortcuts」命令顯示 Jupyter Notebook 支援的快捷鍵，如下圖所示：



首先看到的是命令模式的快捷鍵清單，如下圖所示：



請捲動視窗至最後，可以看到編輯模式的快捷鍵清單，如下圖所示：

