

# Pandas 物件的定義與屬性

[簡報閱讀](#)[範例與作業](#)[問題討論](#)[學習心得\(完成\)](#)

重要知識點

Pandas

補充：函式與函式庫

Pandas 的發展史

Pandas 的重要特性



## 重要知識點



- 知道 Pandas 的特性與貢獻
- 能夠使用 DataFrame 與 Series 當中的屬性

## Pandas

根據官方的描述：「Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.」可以得知 Pandas 是 Python 當中用於資料分析的一個函式庫。



## 補充：函式與函式庫

函式庫是由一堆函式所組成的集合，又稱為是模組、套件或是大陸用語包、模塊等詞。而函式是指一段程式碼的片段封裝而成的函式，可以用於重複呼叫。

## Pandas 的發展史

Pandas 是一個開源的（BSD 協議許可的函式庫），為 Python 提供高性能的數據結構和數據分析工具。Pandas 是由 NumFOCUS 組織所贊助，確保其在開源界的地位。

- 2008: Development of pandas started

Analysis is published

- 2015: pandas becomes a NumFOCUS sponsored project
- 2018: First in-person core developer sprint



## Pandas 的重要特性

- 提供了快速高效的 DataFrame 結構 ( 底層使用 Cython 或 C 的實作對效能進行高度優化 )
- 廣泛地在學術與商業領域中使用，包括金融，神經科學，經濟學，統計學，廣告，Web分析等。
- 對於資料格式有高度的銜接性，包含 CSV、Excel 或資料庫 ( SQL ) 皆能提供彈性的讀寫工具

## 在程式中載入 Pandas

不管是在 .py 或是 Jupyter 的環境，都可以利用 import 的方法載入 numpy 函式庫，一般習慣命名成 np：

```
1 import pandas as pd
```

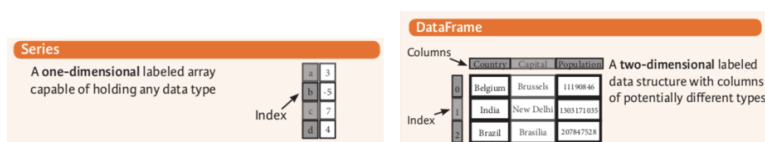
```
1 print(pd)
2 print(pd.__version__)
```

## DataFrame and Series

Pandas 最重要的貢獻是提出了兩種資料結構：

「DataFrame」和「Series」。Series 是一種一維的資料結構，用來一定序列型的資料。

DataFrame 是二維的資料結構，用表格的形式存放常見的資料內容。



## 從 Series 和 DataFrame 開始

在這個例子，我們先產生 Series 和 DataFrame 的變數，並且利用 `type(...)` 查看其屬性與方法：

```
1 import pandas as pd
2
3 s = pd.Series([1, 2, 3])
4 print(s)
5 # 0    1
6 # 1    2
7 # 2    3
8 # dtype: int64
9 print(type(s))
10 # <class 'pandas.core.series.Series'>
```

```
3 df = pd.DataFrame([1, 2, 3])
4 print(df)
5 #      0
6 # 0    1
7 # 1    2
8 # 2    3
9 print(type(df))
10 # <class 'pandas.core.frame.DataFrame'>
```

## Series 的常用屬性

Series 其實就是 NumPy 的 Array 的加工品，所以 Array 有的屬性，Series 絕大部分都可延用：

```
1 import pandas as pd
2
3 s = pd.Series([1, 2, 3])
4 print(s.shape) # (3, )
5 print(s.size) # 3
6 print(s.dtype) # int64
```

請先試試看印出結果，並推測這些屬性的含義是什麼。

補充：除此之外，其他完整屬性可以從 [官網文件](#) 中找到。

## DataFrame 的常用屬性

DataFrame 也是 NumPy 的 Array 的加工品，所以 Array 有的屬性，DataFrame 一樣可以使用：

```
3 print(df.size) # 3
4 print(df.dtypes)
5 # 0      int64
6 # dtype: object
```

請先試試看印出結果，並推測這些屬性的含義是什麼。

補充：除此之外，其他完整屬性可以從 [官網文件](#) 中找到。

## Pandas 中的資料型態

Pandas 大部分沿用 NumPy 定義的型態，只是沒有那麼嚴格。除此之外，Pandas 新增了兩種在資料分析時常用的型態「timedelta」和「category」。

Pandas dtype	Python type	NumPy type
object	str	string_, unicode_
int64	int	int_, int8, int16, int32, int64, uint8, uint16, uint32, uint64
float64	float	float_, float16, float32, float64
bool	bool	bool_
datetime64	NA	datetime64[ns]
timedelta[ns]	NA	NA
category	NA	NA

## Series、DataFrame 與 NdArray 的比較

- DataFrame 代表的是用「資料」的角度去思考程式中的實踐應該長什麼樣子、應該要

- 那你可以會問既然 Seies、DataFrame 都是由陣列所封裝而成的加工品，那為什麼不直接用陣列就好呢？陣列當中所有資料型態必須相同，而 DataFrame 是由 Seies 所組成，也就說同一個欄位形態相同，欄位與欄位間可不相同。

## 知識點回顧

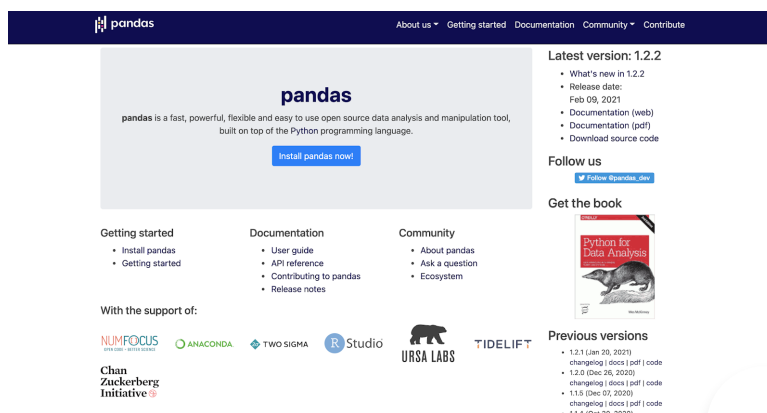
- 知道 Pandas 的特性與貢獻
- 能夠使用 DataFrame 與 Series 當中的屬性
- 初步理解 Seies、DataFrame 與 NdArray 的比較

## 參考資料

### Pandas 官方網站

網站：[pandas](https://pandas.pydata.org/)

Pandas 的官方網站，提供詳盡的使用說明與文檔。對於任何函式不熟的時候，都可以從文件中找到更深入的用法。



[下一步：閱讀範例與完成作業](#)

