

D01 NumPy 套件基本操作

- 📖簡報閱讀
- 💻範例與作業
- 💬問題討論
- ✅學習心得(完成)

- >
- 重要知識點>
- NumPy 簡介>
- 安裝與載入 NumPy>
- 建立 NumPy array (陣列)>
- NumPy陣列的索引和切片 (Slicing)>
- NumPy 陣列的常用屬性>



重要知識點



- 介紹如何安裝及載入 NumPy。
- 依照陣列產生的需求，使用相對應的函式，建立 NumPy 陣列。
- 了解陣列屬性，在操作陣列時用來查看陣列資訊。
- NumPy 相關單元如果沒有特別說明的話，陣列均指 NumPy 陣列 (而非其他，例如 Python 陣列)。

NumPy 簡介

- NumPy 是廣受歡迎的 open source Python 程式庫，支援多維度陣列與矩陣運算，也針對陣列運算提供大量的數學函式庫。
- NumPy 的核心是由 C 語言開發，所以在陣列與矩陣運算時具有高效能的特性。
- 2019 年時 GitHub 統計指出有 74% 的機器學習專案中使用 NumPy。



安裝與載入 NumPy

- 安裝 NumPy 可以使用 `pip install numpy` 或是 `pip3 install numpy`；在 Anaconda 環境下的話，執行 `conda install numpy` 進行安裝。
- 載入 NumPy

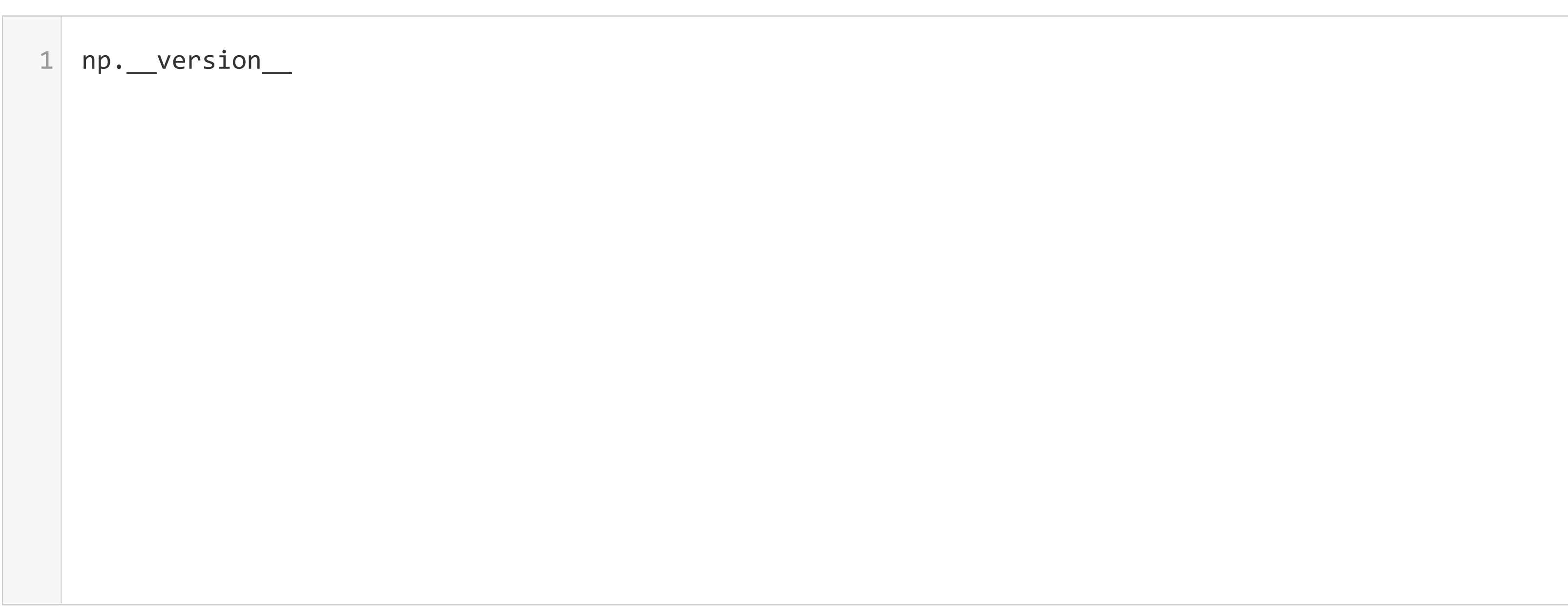
載入 NumPy

```
[2]: import numpy as np
```

目前安裝的 NumPy 版本

```
[3]: np.__version__
```

```
[3]: '1.18.5'
```



建立 NumPy array (陣列)

- 建立陣列的方式是透過執行 NumPy 函式，依照不同的目的，以下逐一介紹常用來建立陣列的函式。
- 最基本的方式是呼叫 `array()` 函式，可將 Python list 或元組 (tuple) 的值建立為 NumPy array。
- 要用序列數字產生等差一維陣列的話，可以使用 `arange()` 與 `linspace()` 函式，兩者的函式引數很類似，其中結束值為必輸，起始值、間隔值非必輸。產生的序列數字包含起始值但不包含結束值，也就是 `[start, stop)` 的表示方式。
- 不同的地方在於 `linspace()` 對於產生的元素可以有更多的控制。
- 呼叫 `zeros()`、`ones()` 函式，可以依照傳入的形狀引數，建立元素全為 0、全為 1 的陣列。
- 使用 `empty()` 函式則是不需要給定起始值，但是可以建立給定形狀的陣列，元素值則會隨機給定。

- NumPy 也提供隨機產生陣列元素的函式，可以用來建立陣列。

函式	說明	產生數值區間	隨機數資料型別	隨機數分佈
<code>rand()</code>	隨機產生指定形狀(shape)的陣列	<code>[0, 1)</code>	浮點數	連續型均勻分布
<code>randn()</code>	隨機產生指定形狀(shape)的陣列	<code>(-1, 1)</code>	浮點數	常態分佈
<code>randint([low[, high, size, dtype]])</code>	隨機產生指定區間元素	<code>[low, high)</code>	整數	離散型均勻分布
<code>random_sample([size])</code>	隨機產生指定大小的一維陣列	<code>[0.0, 1.0)</code>	浮點數	連續型均勻分布
<code>random([size])</code>	隨機產生指定大小的一維陣列	<code>[0.0, 1.0)</code>	浮點數	連續型均勻分布
<code>randf([size])</code>	隨機產生指定大小的一維陣列	<code>[0.0, 1.0)</code>	浮點數	連續型均勻分布
<code>sample([size])</code>	隨機產生指定大小的一維陣列	<code>[0.0, 1.0)</code>	浮點數	連續型均勻分布

- 隨機陣列也可以產生不同分佈的元素，NumPy 提供的函式和分佈非常豐富，所有函式與詳細說明可以參考官方文件。

NumPy陣列的索引和切片 (Slicing)

- 透過索引存取陣列元素或進行切片 (slicing)，可以使用索引值，或是 `[start:stop:step]` 語法取得範圍內的元素，要注意的是起始-結束範圍仍是 **half-open** 的，所以回傳的元素將不包含結束索引的元素。
- 索引 -1 表示取得最後一個元素。切片如果只有給定 `step` 值為 -1 的話，則代表是反向取出，元素值是從最後一筆開始取出。
- 若沒有給定 `start` 或 `stop` 值的話則代表是取出該索引之前或之後的所有元素。若 `start` 和 `stop` 值都沒有給定的話，就是取出所有元素值。

NumPy 陣列的常用屬性

陣列的屬性在操作的過程中經常需要用到，常用屬性及說明如下表：

屬性	說明
<code>shape</code>	陣列的形狀
<code>ndim</code>	陣列的維度數目，也就是軸(axis)的數目
<code>dtype</code>	陣列元素的資料型別
<code>size</code>	陣列元素的數目
<code>flat</code>	陣列的一維迭代器
<code>T</code>	陣列轉置
<code>real</code>	陣列元素複數(complex number)的實數部分
<code>imag</code>	陣列元素複數(complex number)的虛數部分
<code>data</code>	顯示buffer物件，指向陣列資料的開始位址
<code>itemsize</code>	每個元素的記憶體使用量
<code>nbytes</code>	陣列所有元素的記憶體使用量
<code>strides</code>	往相鄰元素移動時需要的byte數

知識點回顧

- NumPy 的安裝及載入。
- 如何依照需求，使用相對應的函式產生陣列。
- 陣列的索引、切片、以及屬性，都是 NumPy 基本操作時常會用到的工具。

[下一步：閱讀範例與完成作業](#)