

第 1 週，第 1 課：一個最小的工作範例

目標

- * 提供一個例子以開始談論 Julia

在我們的第一課中，我們廣泛借用了第 0 課—關於你必須準備好來參加本課程的內容¹。

我還想提醒你課程口號：**小步走，沒有空隙，永遠有意義**。目的是讓緊張的初學者保持參與！

這節課我們從零開始，所以我們不能完全遵守口號的所有部分。這個想法是首先在本課中看到編碼的實際操作，然後在第 2 課中詳細解釋。

輸入 REPL 代碼

打開 REPL，輸入 `"Hello, world"`。

示範：在本課的影片中，我們精確地向你展示如何做到這一點。

`"Hello, world"` 是一個字串值。Julia 有許多其他類型的值：我們將看到其中的一些，例如數字值和字符值²。

現在輸入 `mystringexample1 = "Hello, world"`。這稱為指派值到變數。

重要：`=` 符號將右側的字串值綁定到左側的變數名稱。這在三個地方更改了你的電腦的記憶體：

- 名稱 `mystringexample1` 被放入所謂的命名空間³。
- 字串值 `"Hello, world"` 被建立⁴。
- 左側名稱和右側值之間的 `=` 符號建立了名稱和值之間的綁定。
- 透過將字串值綁定到名稱，Julia 將字串值儲存在你的電腦記憶體中，以便以後需要時可以使用該值。

¹啟動 Julia 並進入 REPL；編輯純文字檔案並將它們儲存在本課程的專用文件夾中。

²即，作為字符的值。

³實際上，當 Julia 運行時，它可以有多個命名空間，但這是我們在本課程中會不涉及的進階主題。一旦名稱位於命名空間中，它就會一直存在，直到你關閉整個命名空間。關閉 Julia 會話也會關閉所有命名空間。

⁴與上面的分開但方式相同。

輸入 `println(mystringexample1)` 。

示範：`println` 是內建函數

當這一行運行時會發生什麼⁵：

函數 `println()` 接收變數名稱 `mystringexample1`，獲取它的值（它是一個字串），重新格式化它，然後在螢幕上顯示字串並接上空行。

函數在 Julia 中非常非常重要。許多是內建的，例如 `println()`，但 Julia 程式還建立了更多。在本課程中，你將學到很多關於 Julia 函數的知識！

建立和運行代碼檔案

最後，建立 `myfirstfile.jl`，作為純文字檔案（NB!），確切地包含我們上面使用的兩行代碼，將其儲存在你的課程文件夾^a。確保你的課程文件夾是你的工作目錄^b，然後輸入 `include("myfirstfile.jl")`

示範：在影片中，我們展示了結果與我們之前使用的 REPL 代碼相同。

^a即建立一個 Julia 代碼檔案—此類檔案是第 0 課的主題之一。

^b使用 `pwd()` 來檢查你的工作目錄是什麼，並使用 `cd()` 來更改它。

恭喜！你的第一個 Julia 程式！編碼就是這麼簡單。

回顧與總結

- * `"Hello, world"` 是一個字串值。
- * `println()` 是一個函數。
- * `mystringexample1` 是一個變數名稱。
- * `=` 是指派：右側的值綁定到左側的名稱。
- * 函數 `include()` 運行它從 Julia 代碼檔案接收的代碼行。
- * Julia 代碼檔案是副檔名為 `.jl` 的純文字檔案

我們在本課中所做的，隨著課程的進行我們將一遍遍地做：一些新想法、一些範例（你可以隨著課程進行嘗試）以及一些供你編寫和執行的代碼檔案。

請在第 1 週第 2 課之前做測驗、練習和自評作業。他們很短！在繼續之前最好先做這些。

⁵人們也說：“當這一行被執行時”，以及“當這一行被評估時”。