# 第 02 天

## 暸解不同的變數類型

我們要花兩天的時間來暸解 R 語言變數類型，開始之前先用一張表格讓您很快地對接下來兩天的學習內容有一個概觀，今天會說明的我們以**粗體**表示：

|  |  |  |
| --- | --- | --- |
| 變數類型 | 變數類型英文 | 範例 |
| **數值** | numeric | 2 |
| **整數** | integer | 2L |
| **邏輯值** | logical | TRUE |
| 文字 | character | “Learning R the easy way” |
| 日期 | Date | Sys.Date() |
| 時間 | POSIXct POSIXt | Sys.time() |

我們不需要自己去猜測變數類型，可以使用一個函數 class() 讓 R 語言告訴我們輸入的變數是什麼類型，就像在 RStudio 的左下角命令列（Console）輸入這些程式，詢問 R 語言它們的類型為何？

> class(2)  
[1] "numeric"  
> class(2L)  
[1] "integer"  
> class(TRUE)  
[1] "logical"  
> class("Learning R the easy way")  
[1] "character"  
> class(Sys.Date())  
[1] "Date"  
> class(Sys.time())  
[1] "POSIXct" "POSIXt"

如果您對 class() 這個函數感到好奇，可以在命令列輸入 ?class 或者 help(class)，這兩個指令都會在右下角打開文件。即便您對 class() 不甚感興趣，知道如何使用 ? 或 help() 進行查詢也相當有幫助。

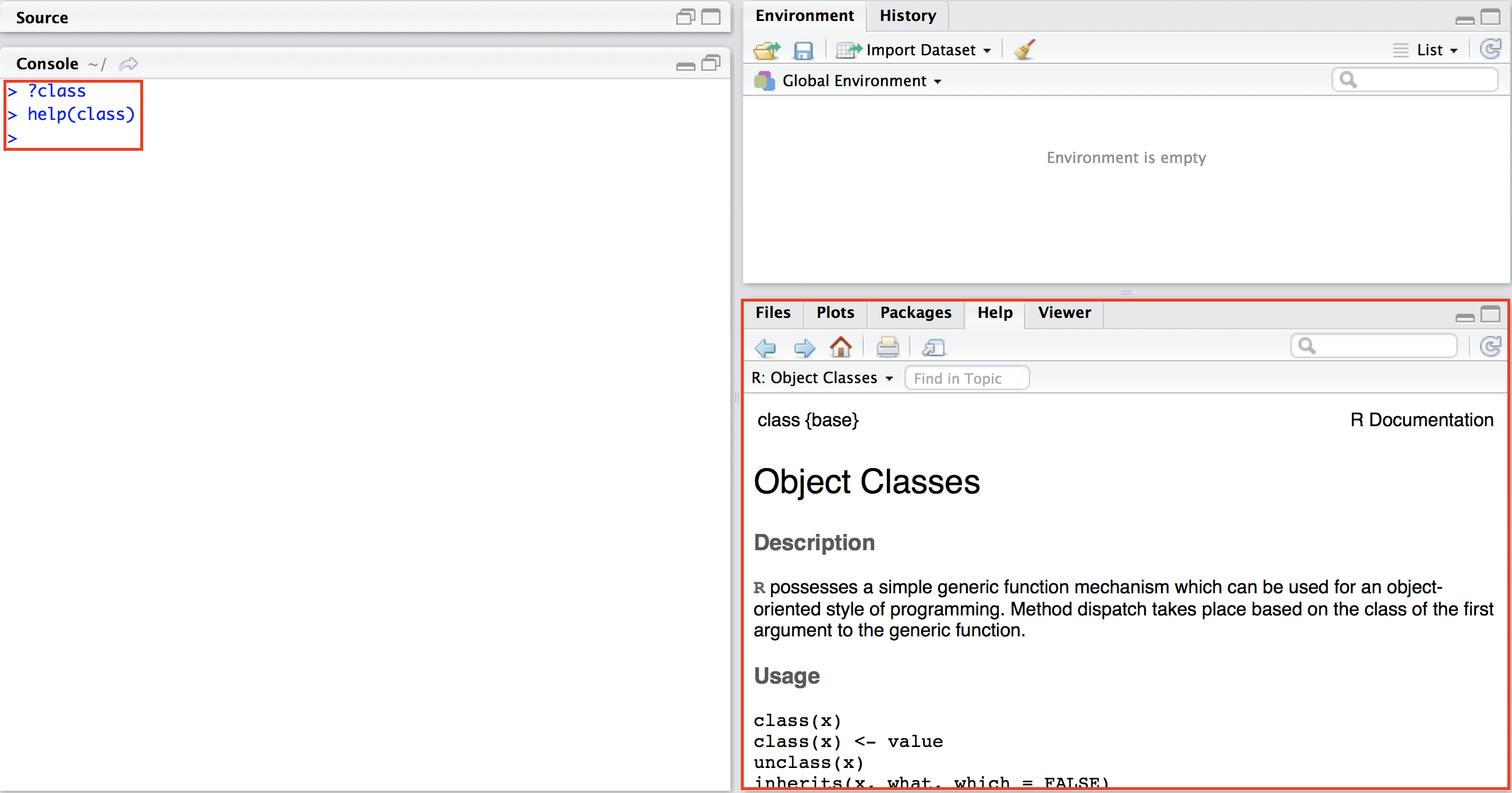


圖 2-1 查詢文件

### 賦值

在深入暸解不同變數類型之前，我們要先學習一個符號 <- 稱為賦值，它的作用是將符號右邊的值指派給符號左邊的物件，舉例來說我們可以將 2 指派給一個叫做 my\_num 的物件，當我們想知道 2 的變數類型為何，我們可以在 class() 中輸入 my\_num：

> my\_num <- 2  
> class(my\_num)  
[1] "numeric"

在 RStudio 中，我們可以按 alt 與 - 幫我們生成 <- 符號。<- 符號跟其他程式語言的 = 符號功能是完全相同的。事實上在 R 語言中使用 = 符號來賦值也是完全沒問題，但是因為絕大多數的 R 語言使用者仍然習慣使用 <- 符號，我還是推薦您使用 <- 符號。

### 數值

不論我們輸入的數字帶有小數位數或不帶有小數位數，R 語言預設儲存為數值（numeric）。

> my\_num <- 2.33  
> class(my\_num)  
[1] "numeric"  
> my\_num <- 2.0  
> class(my\_num)  
[1] "numeric"  
> my\_num <- 2  
> class(my\_num)  
[1] "numeric"

### 整數

當我們輸入一個整數並加入 L，R 語言就會儲存為整數（integer）。如果整數帶有不必要的小數位數，R 語言會回傳警示訊息但依舊會儲存為整數；但若在帶有小數位數的數字後加上 L 則 R 語言回傳警示訊息，並且忽略 L 儲存為數值。

> my\_int <- 2L  
> class(my\_int)  
[1] "integer"  
> my\_int <- 2.0L  
Warning message:  
integer literal 2.0L contains unnecessary decimal point   
> class(my\_int)  
[1] "integer"  
> my\_int <- 2.33L  
Warning message:  
integer literal 2.33L contains decimal; using numeric value   
> class(my\_int)  
[1] "numeric"

### 邏輯值

當我們進行**判斷條件**或者**篩選**的時候就會需要使用邏輯值（logical），邏輯值只有 TRUE 與 FALSE 這兩個值，或者可以簡寫為 T 與 F。

> class(TRUE)  
[1] "logical"  
> class(FALSE)  
[1] "logical"  
> class(T)  
[1] "logical"  
> class(F)  
[1] "logical"

這裡要提醒您一個觀念，R 語言對大小寫是敏感的（case-sensitive），像是 TRUE 會被 R 語言識別為邏輯值，但是 True 與 true 則不會喔！

> class(TRUE)  
[1] "logical"  
> class(True)  
Error: object 'True' not found  
> class(true)  
Error: object 'true' not found

除了直接輸入邏輯值，我們也可以透過**判斷條件**得到邏輯值的輸出：

> 8 > 7 # 判斷 8 是否大於 7  
[1] TRUE  
> 8 < 7 # 判斷 8 是否小於 7  
[1] FALSE  
> 8 >= 7 # 判斷 8 是否大於等於 7  
[1] TRUE  
> 8 <= 7 # 判斷 8 是否小於等於 7  
[1] FALSE  
> 8 == 7 # 判斷 8 是否等於 7  
[1] FALSE  
> 8 != 7 # 判斷 8 是否不等於 7  
[1] TRUE  
> 7 %in% c(8, 7) # 判斷 7 是否包含於一個 c(8, 7) 之中  
[1] TRUE

我們稍微停頓一下，因為這裡有三個新觀念需要您花點時間吸收：

1. 在 # 符號後面編寫註解，用口語方式告訴別人這段程式在做什麼事情
2. c(8, 7) 是一種叫做**向量**的資料結構，我們會在之後的章節細談
3. 常用的判斷運算子有：

|  |  |
| --- | --- |
| 判斷運算子 | 作用 |
| == | 等於 |
| > | 大於 |
| < | 小於 |
| >= | 大於等於 |
| <= | 小於等於 |
| != | 不等於 |
| %in% | 包含於 |

### 數學運算

今天的最後一個主題是對數值，整數與邏輯值進行數學運算，常用的數學運算子有六種：

|  |  |
| --- | --- |
| 數學運算子 | 作用 |
| + | 加 |
| - | 減 |
| \* | 乘 |
| / | 除 |
| ^ 或 \*\* | 次方 |
| %% | 回傳餘數 |

我們先進行僅有數值的運算：

> first\_num <- 8  
> second\_num <- 7  
>   
> first\_num + second\_num  
[1] 15  
> first\_num - second\_num  
[1] 1  
> first\_num \* second\_num  
[1] 56  
> first\_num / second\_num  
[1] 1.142857  
> first\_num^second\_num  
[1] 2097152  
> first\_num %% second\_num  
[1] 1

接著我們進行僅有整數的運算：

> first\_int <- 8L  
> second\_int <- 7L  
>   
> ans <- first\_int \* second\_int  
> ans  
[1] 56  
> class(ans)  
[1] "integer"  
> ans <- first\_int / second\_int  
> ans  
[1] 1.142857  
> class(ans)  
[1] "numeric"

相乘之後仍然是整數的變數類型，注意相除之後 R 語言自動幫我們把答案轉換成了適當類型，我們時不時都可以觀察到 R 語言的彈性：

> my\_num <- 8  
> my\_int <- 7L  
>  
> ans <- my\_num + my\_int  
> class(ans)  
[1] "numeric"

最後我們把邏輯值也放進來做四則運算，看看會發生什麼事情：

> my\_num <- 8  
> my\_int <- 7L  
> my\_logi <- TRUE  
>   
> ans <- my\_num + my\_int + my\_logi  
> ans  
[1] 16  
> class(ans)  
[1] "numeric"  
>  
> my\_logi <- FALSE  
> ans <- my\_num + my\_int + my\_logi  
> ans  
[1] 15  
> class(ans)  
[1] "numeric"

為什麼 TRUE 和 FALSE 納入四則運算沒有任何問題呢？原來在 R 語言中，TRUE 跟 1 或者 1L 是相等的；FALSE 跟 0 或者 0L 是相等的。

> my\_num <- 0  
> my\_int <- 0L  
> my\_logi <- FALSE  
>   
> my\_num == my\_logi  
[1] TRUE  
> my\_int == my\_logi  
[1] TRUE  
>  
> my\_num <- 1  
> my\_int <- 1L  
> my\_logi <- TRUE  
>   
> my\_num == my\_logi  
[1] TRUE  
> my\_int == my\_logi  
[1] TRUE

### 小結

第二天我們學會三種 R 語言的基本變數類別：數值（numeric）、整數（integer）與邏輯值（logical）。

### 練習題

###### 1. 將您的身高（公分）指派給 my\_height；體重（公斤）指派給 my\_weight

my\_height <- \_\_\_  
my\_weight <- \_\_\_

###### 2. 分別將 my\_height 與 my\_weight 輸出在命令列

\_\_\_  
\_\_\_

###### 3. 利用 my\_height 與 my\_weight 計算您的**身體質量指數**（Body Mass Index，BMI），BMI 計算公式為：

bmi <- (\_\_\_) / (\_\_\_ / 100)^2  
bmi

### 延伸資訊

* [An Introduction to R](https://cran.r-project.org/doc/manuals/r-release/R-intro.html)