

Lesson 01: 變數與基本資料型別

歡迎來到第一課！

我們要學習程式設計中最基礎也最重要的概念：變數和資料型別。如果把寫程式比作做菜，變數就像是廚房裡的各種容器，有些裝調味料、有些裝食材、有些裝工具。

什麼是變數（Variable）？

變數的概念

想像你面前有很多個盒子，每個盒子上面都有一個標籤。你可以：

- 在盒子裡放東西
- 從盒子裡拿東西
- 把盒子裡的東西換成別的東西
- 看看盒子裡現在裝了什麼

這些盒子就是「變數」，盒子上的標籤就是「變數名稱」。

讓我們看第一個例子：

```
name = "小明"  
age = 25
```

這裡我們做了兩件事：

1. 建立一個叫做 `name` 的盒子，裡面放了文字「小明」
2. 建立一個叫做 `age` 的盒子，裡面放了數字 25

變數命名規則

就像給盒子貼標籤一樣，變數的名稱也有一些規則：

可以使用的字元：

- 英文字母（a-z, A-Z）
- 數字（0-9，但不能當開頭）
- 底線

合法的變數名稱：

name	-- 簡單明瞭
user_age	-- 用底線分隔（蛇式命名法）
firstName	-- 用大小字母分隔（駝峰式命名）
score1	-- 數字在後面
_temporary	-- 以底線開頭

不合法的變數名稱：

2name	-- 不能以數字開頭
user-age	-- 不能使用連字號
first name	-- 不能有空格
if	-- 不能使用關鍵字

為什麼叫「變數」？

因為盒子裡的東西是可以改變的：

score = 85	-- 一開始分數是 85
print(score)	-- 顯示：85
score = 92	-- 分數變成 92
print(score)	-- 顯示：92

這就是為什麼叫「變數」——裡面的數值可以變化。

Lua 的基本資料型別

Lua 有幾種基本的資料型別，就像不同的容器適合裝不同的東西一樣。

1. Number（數字）

在 Lua 中，所有的數字都是 Number 型別，不管是整數還是小數：

```
-- 整數
student_count = 30
temperature = -5

-- 小數
height = 175.5
weight = 65.2
pi = 3.14159

-- 科學記號
speed_of_light = 3e8    -- 3 × 10^8
```

Lua 的數字可以做各種運算：

```
-- 基本運算
result = 10 + 5      -- 加法：15
result = 10 - 3      -- 減法：7
result = 6 * 7        -- 乘法：42
result = 20 / 4       -- 除法：5
result = 17 % 5       -- 取餘數：2
result = 2 ^ 3        -- 次方：8

print(result)        -- 顯示最後的結果：8
```

2. String (字串)

字串就是文字，用來儲存各種文字資訊：

```
-- 用雙引號包起來
message = "Hello, World!"
name = "Kitty"

-- 用單引號也可以
greeting = 'Hi there!'
title = '程式設計師'

-- 空字串
empty = ""
```

字串可以連接在一起，使用 `..` 運算子：

```
first_name = "張"
last_name = "小明"
full_name = first_name .. last_name
print(full_name)      -- 顯示：張小明

-- 可以跟數字混合
age = 25
message = "我今年 " .. age .. " 歲"
print(message)        -- 顯示：我今年 25 歲
```

特殊字元：

```
-- 換行
text = "第一行\n第二行"
print(text)
-- 顯示：
-- 第一行
-- 第二行

-- Tab 鍵
text = "姓名\t年齡"
print(text)          -- 顯示：姓名    年齡
```

3. Boolean（布林值）

Boolean 只有兩個值：`true`（真）和 `false`（假）。就像開關只有「開」和「關」兩種狀態：

```
is_student = true
has_license = false
is_raining = true

-- 用在判斷中
age = 20
is_adult = age >= 18    -- 結果是 true

print(is_adult)        -- 顯示：true
```

4. nil（空值）

`nil` 表示「什麼都沒有」，就像空盒子：

```
-- 還沒有給值的變數預設是 nil
local unknown_value
print(unknown_value)    -- 顯示：nil

-- 可以明確設定為 nil
user_email = nil
print(user_email)       -- 顯示：nil
```

變數的操作

宣告和賦值

在 Lua 中，你不需要事先「宣告」變數，直接給它一個值就可以了：

```
-- 同時建立變數並給值
player_name = "小華"
player_score = 1500
is_winner = true

-- 一次設定多個變數
x, y, z = 10, 20, 30
print(x, y, z)    -- 顯示：10    20    30
```

查看變數型別

使用 `type()` 函式可以查看變數的型別：

```
name = "小明"
age = 25
is_student = true
nothing = nil

print(type(name))    -- 顯示：string
print(type(age))     -- 顯示：number
print(type(is_student)) -- 顯示：boolean
print(type(nothing)) -- 顯示：nil
```

型別轉換

有時候我們需要把一種型別轉換成另一種：

```
-- 數字轉字串
age = 25
age_text = tostring(age)
print(type(age_text))    -- 顯示：string

-- 字串轉數字
score_text = "95"
score = tonumber(score_text)
print(type(score))       -- 顯示：number

-- 如果轉換失敗會得到 nil
bad_number = tonumber("abc")
print(bad_number)        -- 顯示：nil
```

運算子詳解

算術運算子

```
a = 10
b = 3

print(a + b)    -- 加法：13
print(a - b)    -- 減法：7
print(a * b)    -- 乘法：30
print(a / b)    -- 除法：3.333...
print(a % b)    -- 取餘數：1
print(a ^ b)    -- 次方：1000

-- 負數
print(-a)       -- 負數：-10
```

比較運算子

```
a = 10
b = 5

print(a == b)   -- 相等：false
print(a ~= b)   -- 不相等：true
print(a > b)    -- 大於：true
print(a < b)    -- 小於：false
print(a >= b)   -- 大於等於：true
print(a <= b)   -- 小於等於：false
```

邏輯運算子

```
-- and (且) : 兩邊都要是 true, 結果才是 true
print(true and true)    -- true
print(true and false)   -- false
print(false and true)   -- false

-- or (或) : 只要有一邊是 true, 結果就是 true
print(true or false)    -- true
print(false or true)    -- true
print(false or false)   -- false

-- not (非) : 把 true 變 false, 把 false 變 true
print(not true)         -- false
print(not false)        -- true
```

實用範例

讓我們來看幾個實際應用的例子：

範例 1：計算 BMI

```
-- 取得身高和體重
height = 1.75      -- 公尺
weight = 70        -- 公斤

-- 計算 BMI
bmi = weight / (height ^ 2)
print("你的 BMI 是：" .. bmi)

-- 判斷體重狀態
if bmi < 18.5 then
    status = "體重過輕"
elseif bmi < 24 then
    status = "體重正常"
else
    status = "體重過重"
end

print("體重狀態：" .. status)
```

範例 2：溫度轉換

```
-- 攝氏轉華氏
celsius = 25
fahrenheit = celsius * 9 / 5 + 32
print(celsius .. "°C = " .. fahrenheit .. "°F")

-- 華氏轉攝氏
fahrenheit = 77
celsius = (fahrenheit - 32) * 5 / 9
print(fahrenheit .. "°F = " .. celsius .. "°C")
```

範例 3：購物計算

```
-- 商品資訊
item_name = "Lua 程式設計書"
price = 450
quantity = 2
discount_rate = 0.1      -- 10% 折扣

-- 計算總價
subtotal = price * quantity
discount = subtotal * discount_rate
total = subtotal - discount

-- 顯示結果
print("商品：" .. item_name)
print("單價：" .. price)
print("數量：" .. quantity)
print("小計：" .. subtotal)
print("折扣：" .. discount)
print("總計：" .. total)
```

常見錯誤和注意事項

1. 使用未定義的變數

```
-- 錯誤示範
print(undefined_variable)  -- 顯示：nil（不會出錯，但可能不是你想要的）

-- 正確做法
my_variable = "Hello"
print(my_variable)          -- 顯示：Hello
```


2. 型別混合運算

```
-- 這樣會出錯
age = "25"
next_year = age + 1          -- 錯誤！不能把字串和數字相加

-- 正確做法
age = "25"
age_number = tonumber(age)
next_year = age_number + 1
print("明年" .. next_year .. "歲")
```

3. 變數名稱大小寫敏感

```
name = "小明"
Name = "小華"      -- 這是不同的變數！
NAME = "小李"      -- 這也是不同的變數！

print(name)        -- 小明
print(Name)        -- 小華
print(NAME)        -- 小李
```

練習時間

現在輪到你動手了！請嘗試以下練習：

基礎練習

1. 個人資訊：
建立變數儲存你的姓名、年齡、身高，然後用 `print` 顯示出來
2. 簡單計算：
計算一個矩形的面積（長 × 寬）
3. 字串操作：
組合你的姓氏和名字，並加上問候語

進階練習

1. 年齡計算：
根據出生年份計算現在的年齡
2. 單位轉換：
將公尺轉換成公分和英吋

grandMA2 燈光控制實戰應用

學習變數不只是為了程式設計，而是要實際控制燈光！讓我們看看變數在 grandMA2 plugin 中如何應用：

燈光控制基礎範例

```
-- 控制你的第一盞燈
function Start()
    -- 燈光控制變數
    local fixture_number = 1      -- 要控制的燈具編號
    local brightness = 75         -- 亮度設定 (0-100)
    local color = "red"           -- 顏色名稱

    -- 執行燈光指令
    gma.echo("正在控制 Fixture " .. fixture_number .. "，亮度設為 " ..
brightness .. "%")
    gma.cmd("Fixture " .. fixture_number .. " at " .. brightness)

    -- 等待 2 秒後改變顏色
    gma.sleep(2)
    gma.echo("改變顏色為：" .. color)
    gma.cmd("Fixture " .. fixture_number .. " at " .. color)
end
```

為什麼變數在燈光控制中很重要？

1. 記錄燈具資訊：像是燈具的「身分證」

```
local front_light_1 = 1      -- 前場第一盞燈
local back_light_1 = 10      -- 後場第一盞燈
local effect_light = 20      -- 效果燈編號
```

2. 儲存 DMX 數值：記住每盞燈的狀態

```
local dimmer_value = 85      -- 調光器數值
local pan_position = 127     -- 水平位置
local tilt_position = 64     -- 垂直位置
```

3. 處理使用者輸入：讓 plugin 可以互動

```
local user_input = gma.textinput("請輸入亮度", "0-100")
local brightness = tonumber(user_input) or 50 -- 預設值 50
```

實際應用：簡單燈光控制器

```
-- Plugin 名稱: Simple Light Controller
function Start()
  -- 設定變數
  local fixture_range = "1 thru 10"  -- 控制燈具 1 到 10
  local fade_time = 3                -- 漸變時間 (秒)
  local max_brightness = 100         -- 最大亮度

  -- 執行漸亮效果
  gma.echo("開始漸亮效果...")
  gma.cmd("Fixture " .. fixture_range .. " at " .. max_brightness .. "
fade " .. fade_time)

  -- 等待漸變完成
  gma.sleep(fade_time + 1)

  -- 詢問是否要關燈
  local turn_off = gma.gui.confirm("燈光控制", "是否要關閉燈光?")
  if turn_off then
    gma.echo("關閉燈光...")
    gma.cmd("Fixture " .. fixture_range .. " at 0 fade " ..
fade_time)
  end
end

function Cleanup()
  gma.echo("燈光控制器已結束")
end

return Start, Cleanup
```

變數類型在燈光控制中的具體用途

變數類型	燈光控制用途	實例
Number	DMX 數值、亮度百分比、時間	brightness = 75
String	燈具名稱、顏色、指令	color = "blue"
Boolean	燈光開關狀態、效果啟用	strobe_active = true
nil	檢查燈具是否存在	if fixture_handle == nil

小結

在這一課中，我們學會了：

1. **變數的概念**：像是有標籤的盒子，可以儲存各種資料
2. **基本資料型別**：Number、String、Boolean、nil
3. **運算子**：算術、比較、邏輯運算
4. **型別轉換**：在不同型別間轉換資料
5. **常見錯誤**：如何避免典型的新手陷阱
6. **實際應用**：如何用變數控制 grandMA2 燈光系統

這些是所有程式設計的基礎。掌握了變數和資料型別，你就有了建構更複雜程式的基本工具。更重要的是，你現在知道如何用變數來控制真實的燈光設備，讓程式設計不再是抽象概念，而是能創造視覺效果的實用技能！