

【Lua】table

模擬陣列用法

注意 lua 索引 從 **1** 開始

```
-- 宣告 table
lang = {
    "C",
    "C#",
    "C++",
    "Java",
    "Swift",
    "Python",
    "Haskell",
}

-- 使用 ipairs 迭代 lang
for i in ipairs(lang) do
    print(i, #lang[i], lang[i])
end

-- 以下寫法相同
for i,v in ipairs(lang) do
    print(i, #v, v)
end

-- 1 1 C
-- 2 2 C#
-- 3 3 C++
-- 4 4 Java
-- 5 5 Swift
-- 6 6 Python
-- 7 7 Haskell
```

新增資料 => table.insert

```
lang = {
    "C",
    "C#",
    "C++",
    "Java",
    "Swift",
    "Python",
    "Haskell",
}

-- 使用table.insert
table.insert(lang, "lua")
-- 使用索引
lang[9] = "Javascript"

-- 1 1 C
-- 2 2 C#
-- 3 3 C++
-- 4 4 Java
-- 5 5 Swift
-- 6 6 Python
-- 7 7 Haskell
-- 8 3 lua
-- 9 10 Javascript
```

刪除資料 => 設為 nil

```
lang = {
  "C",
  "C#",
  "C++",
  "Java",
  "Swift",
  "Python",
  "Haskell",
}

-- 刪除總是山最後一筆，不然會有意外錯誤
lang[8] = nil

for i,v in ipairs(lang) do
  print(i, #v, v)
end
-- 1 1 C
-- 2 2 C#
-- 3 3 C++
-- 4 4 Java
-- 5 5 Swift
-- 6 6 Python
-- 7 7 Haskell
```

類map(key,vlaue)宣告

key 不要文字,數字混和宣告

```
person = {
  ["name"] = "Bob",
  ["age"] = 25,
}

-- key不為數字－以上相同
person = {
  name = "Bob",
  age = 25,
}

-- 陣列用法
arr = {
  [1] = 1,
  [2] = 2,
  [3] = 3,
  [4] = 4,
}
```

Key 值範圍

key可以是除了 `nil` 和 `NaN` (Not a Number)以外的任何型別。

```
obj = {} -- 建立一個空表
obj[1] = 1 -- 整數是合法的key值
obj[1.0] = 2 -- 浮點數是合法的key值
obj["string"] = 1 -- 字串是合法的key值
obj[math.huge] = 1 -- inf是合法的key值

--[
要注意的是 obj[1] 和 obj[1.0]相同
其obj[1]和obj[1.0]最終值為2
--]
```

```
print(obj[nil]) --> nil
obj[nil] = 1 --> Error: nil不是合法的key值，儘管取值不會出錯

print(obj[0/0]) --> nil
print(0/0) --> -nan
obj[0/0] = 1 --> Error: NaN不是合法的key值
```

🕒修訂版本 #3

★由 treeman 建立於 22 🌟G🌟🌟 2023 23:25:48

✎由 treeman 更新於 5 🌟🌟🌟🌟 2023 10:14:59