

# 正規表示式入門

郭東覺

[kth@cna.isu.edu.tw](mailto:kth@cna.isu.edu.tw)

義守大學校園網路策進會



# Outline

- 簡介
- 應用範圍
- 入門

# 正規表示式 (Regular Expression)

- Regular Expression 定義 -  
一連串可用以表達資料中我們欲搜尋之子  
字串形式的字元集合。
- 不限於 PHP，其他如 Perl、Python 也可使  
用。

# 分析資料的特性

以下資料有何特性？試分析：

- B8903041、A9003121
- 90/12/31、901231
- ABC12345、CD59988
- A123456789、A123456799
- [abc@isu.edu.tw](mailto:abc@isu.edu.tw)
- 01,13,33,34,35、32,08,23,41,06
- 1.23.253.8

# 特殊字元

■ 特殊字元： . \* ? + [ ] ( ) { } ^ \$ | \

■ 遇到特殊字元需再加一個 \ 字元

Ex1: **[AB]** -> **\[AB\]**

Ex2: **C:\** -> **C:\\**

# 集合與出現次數

- $[a-z]$  代表 a-z 的集合
- $[^abc0-9]$  代表不屬於 abc、0-9 的集合
- $A^+$  代表 A 出現 1 次以上
- $[A-Z]^*$  代表 A-Z 出現 0 次以上
- $[AB]?$  代表 A or B 出現 0~1 次
- $[0-9]\{3\}$  代表 0-9 恰巧出現 3 次
- $[0-9]\{5,6\}$  代表 0-9 出現 5~6 次

# 定義 Pattern

以判斷學生學號為例：

■ A9303003、B9403111

1. **開頭**必為 A，也可省略。 -> [AB]?

2. 其後必接 6~7 個數字**結束**。 -> [0-9]{6,7}

學生學號格式 ->  $^[AB]?[0-9]{6,7}\$$

# 程式範例 (1) - 判別

- `bool ereg ( string pattern, string string [, array regs] ) ;`
- `$Student_ID = "A9001123";`  
`if(ereg("^B", $Student_ID))`  
    `echo "開頭為 B";`  
`else`  
    `echo "開頭不為 B";`



## 程式範例 (2) - 切割

- array **split** ( string pattern, string string [, int limit]);
- `$Date1 = "90/01/02";`  
`$Date2 = "91|02|03";`  
`$Array1 = split("\\|/", $Date1);`  
`$Array2 = split("\\|/", $Date2, 2);`  
`echo $Array1[0] . "\t" . $Array2[1];`
- 輸出結果：90 02|03