

# Regular Expression

- Regular Expression
  - 是一種表達"具有某種特徵"字串的方式，可用來完全指定需要加以處理的資料，避免反覆判斷找尋的困擾
- Regular Expression的組成
  - 為了表達"特徵"需定義範本 (Pattern)，範本由 **普通字元 (ASCII)**、**特殊字元 (Metacharacter)**及**數量定義詞(Quantifier)**組成。
  - 其格式為[普通字元|特殊字元]數量定義詞
- 普通字元
  - 普通字元代表的就是字元本身。因此"A"對應到字串就是"A"或"a"

# Regular Expression

- 特殊字元(Metacharacter)—Match Metacharacter

Char	Description
.	代表任一個字元
[...]	代表字元集中的任一字元，例如 [abc] 可對應 a，b 或 c 連續字元的定義可用"-", 例如 [a-d] = [abcd]
[^...]	代表非字元集中的任一字元，例如 [^abc] 將不對應 a，b 或 c

- 特殊字元(Metacharacter)—Position Metacharacter

Char	Description
^	代表字串的開頭
\$	代表字串的結尾

# Regular Expression

- 特殊字元(Metacharacter)—Shorthand Metacharacter

Char	Description
<b>\d</b>	對應0-9的任一數字(= [0-9])
<b>\D</b>	對應非數字的任一字元(=[^0-9])
<b>\f</b>	對應換頁字元
<b>\n</b>	對應換行字元
<b>\r</b>	對應歸正字元
<b>\s</b>	對應空白字元，對等於 [ \f\n\r\t\v]
<b>\S</b>	對應非空白字元，對等於 [^ \f\n\r\t\v]
<b>\t</b>	對應 <b>tab</b> 字元
<b>\v</b>	對應垂直 <b>tab</b> 字元
<b>\w</b>	對應任何文數字元包括"_"，對等於 [a-zA-Z0-9_]
<b>\W</b>	對應任何非文數字元，對等於 [^a-zA-Z0-9_]

# Regular Expression

- 特殊字元(Metacharacter)—Other Metacharacter

Char	Description
	邏輯 "Or"
(pattern)	使用括號將pattern分組並提供記憶的功能，提供往後運算時再存取被括住的運算式功能。當有許多括號在pattern中使用時，被括住的運算式由左至右，可依序用\$1、\$2...\$9存取。 例如，"(a(bc)(d))" 運算式，被括號的運算式將有如下的對應 \$1="abcd"，\$2="bc"，\$3="d"。
\$1 .. \$9	依序對應pattern運算式中被括號刮住的部分

# Regular Expression

- 數量定義詞(Quantifier)一定義前一個字元的數量

Char	Description
?	一個字元或沒有
*	任意數目的字元或沒有
+	一個字元或以上的字元
{N}	N個字元
{N,}	至少N個字元
{N, M}	至少N個字元至多M個字元

# re

- re
  - 想找到某個規則的正則又有測式資料，可以到以下網址測試：  
<https://regex101.com/>

