
東吳大學 X 理律學堂

法律科技學 - 正規表達法

吳貫綸助教

正規表達法

- 如果一個字符串包含某些特徵的東西, 那麼它就是我正在找的東西



身份證號碼

規則為首字為字母
後緊接著九個數字

生日

年份若為民國則為兩個數字
西元則為四個數字

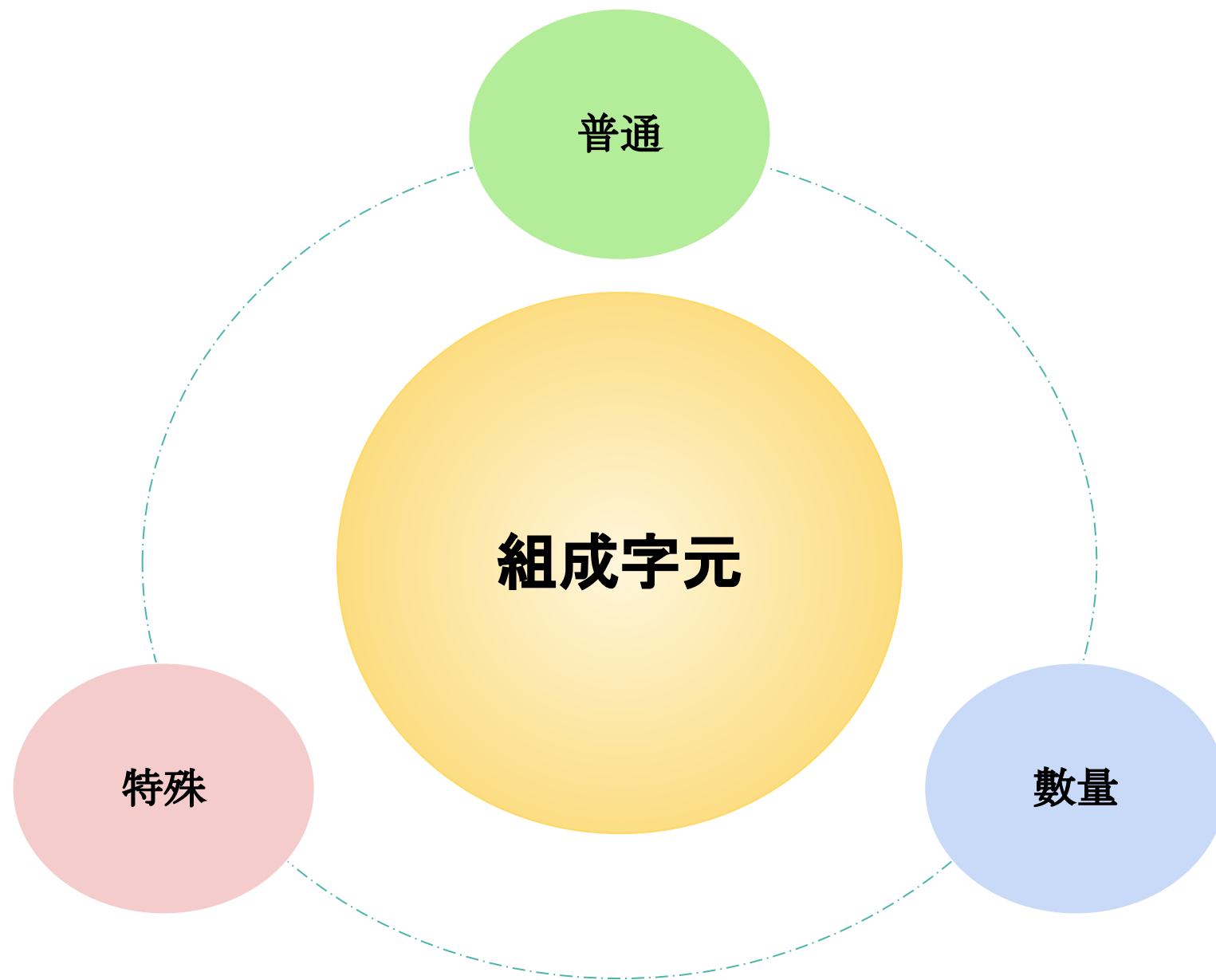
使用差別

未使用正規表達法

```
>>> def isPhoneNumber(text):
>>>     if len(text) != 12:
>>>         return False
>>>     for i in range(0,2):
>>>         if not text[i].isdecimal():
>>>             return False
>>>     if text[2] != '-':
>>>         return False
>>>     for i in range(4,7):
>>>         if not text[i].isdecimal():
>>>             return False
>>>     if text[7] != '-':
>>>         return False
>>>     for i in range(8,11):
>>>         if not text[i].isdecimal():
>>>             return False
>>>     return True
>>>
>>> print('this is a number: 02-1234-5555')
>>> print(isPhoneNumber('02-1234-5555'))
>>> print('this is a number: oh no! regular expression')
>>> print(isPhoneNumber('oh no! regular expression'))
```

使用正規表達法

```
>>> import re
>>> phoneNumRegex = re.compile(r'\d\d-\d\d\d\d-\d\d\d\d')
>>> result = phoneNumRegex.findall('Please call David at 02-8888-1688 by today. 02-9888-9898 is his office number.')
>>> number = ', '.join(result)
>>> print('Phone Number: ' + number)
>>> print('Please call him/her as soon as possible.')
```



普通字元

通用式	說明	比對成立之字串
/a/	含字母“a”的字串。	"ab", "bac", "cba"
/a./	含字母“a”以及其後任一個字元的字串。	"ab", "bac"
/^xy/	以“xy”開始的字串。	"xyz", "xyab"
/xy\$/	以“xy”結尾的字串。	"axy", "abxy"
[13579]	包含 "1" 或 "3" 或 "5" 或 "7" 或 "9" 的字串,	"a3b", "1xy"
b[aeiou]t	"bat", "bet", "bit", "bot", "but"	"bat", "bit"
[0-9]	含數字之字串	-
[a-z0-9]	含數字或小寫字母之字串	-
[a-zA-Z0-9]	含數字或字母之字串	-
[^0-9]	不含數字之字串	-

特殊字元

通用式	說明	等效正規表達法
^	符合輸入字串的開始位置。	-
\$	符合輸入字串的結束位置。	-
d	匹配一個數字字元。[0-9]。	[0-9]
D	匹配一個非數字字元。	[^0-9]
w	匹配包括底線的任何單詞字元	[A-Za-z0-9_]
W	匹配任何非單詞字元	[^A-Za-z0-9_]
s	匹配任何空白字元，包括空格、制表符、換頁符	[\r\t\n\f]
S	匹配任何非空白字元	"bxt", "bzt"

數量字元

通用式	說明	比對成立之字串
*	符合前面的子表達式零次或多次。	zo* 符合「z」、「zo」以及「zoo」。
+	符合前面的子表達式一次或多次。	「zo+」符合「zo」以及「zoo」，但不能符合「z」。
?	符合前面的子表達式零次或一次。	「do(es)?」符合「does」中的「do」和「does」。? 等價於 {0,1}。
{n}	n是一個非負整數。符合確定的 n次。	「o{2}」不能符合「Bob」中的「o」，但是符合「food」中的兩個o。
{n,}	n是一個非負整數。至少符合 n次。	「o{2,}」不能符合「Bob」中的「o」，但能符合「foooooo」中的所有o。「o{1,}」等價於「o+」。 「o{0,}」則等價於「o*」。
{n,m}	m和n均為非負整數，其中 $n \leq m$ 。最少符合 n次且最多符合 m次。	「o{1,3}」符合「foooooo」中的前三個o。 「o{0,1}」等價於「o?」。請注意在逗號和兩個數之間不能有空格。



正規表達法

常用函式又有哪些呢?



- ◎ **group** - 返回群組, 可賦予id名稱尋找
- ◎ **findall** - 尋找所有匹配
- ◎ **sub** - 替換字元
- ◎ **spilt** - 拆開
- ◎ **compile** - 包裝

動手試試吧

網站連結：<https://regexone.com/>



RegexOne

Learn Regular Expressions with simple, interactive exercises.



Lesson 1: An Introduction, and the ABCs

Regular expressions are extremely useful in extracting information from text such as code, log files, spreadsheets, or even documents. And while there is a lot of theory behind formal languages, the following lessons and examples will explore the more practical uses of regular expressions so that you can use them as quickly as possible.

The first thing to recognize when using regular expressions is that **everything is essentially a character**, and we are writing patterns to match a specific sequence of characters (also known as a string). Most patterns use normal ASCII, which includes letters, digits, punctuation and other symbols on your keyboard like %#\$@!, but unicode characters can also be used to match any type of international text.

Below are a couple lines of text, notice how the text changes to highlight the matching characters on each line as you type in the input field below. To continue to the next lesson, you will need to use the new syntax and concept introduced in each lesson to write a pattern that matches all the lines provided.

Go ahead and try writing a pattern that matches all three rows, **it may be as simple as the common letters on each line.**

Exercise 1: Matching Characters

Task	Text
------	------

Match	abcdefg
-------	---------

Match	abcde
-------	-------

Match	abc
-------	-----

Continue >

Solve the above task to continue on to the next problem, or read the [Solution](#).