

## 23. 正規表達式 Regular Expression



regular

正規表示法，又稱正規表示式、正規表達式；英文 Regular Expression，Regex 它不是一個程式語言，他只是一種「字串樣式規則」的「表示法」，用來表達字元符號在字串中出現的規則，大部分的程式語言都有支援 Regex 的用法，而任何工具只要支援這種表示法，你就可以在這工具上用 Regex 來處理字串。

在定義正規表達式上，可以用兩個 / 夾住條件

例如：驗證字串為 a~z 的字母

```
var str = 'hello world';

const rules = /[a-z]/; // 驗證字串為 a~z 的字母
console.log(rules.test(str));
```

### 驗證字元：

/./：任意字元。符合範例：「hello@@@@」

```
var str02 = 'hello world';

const rules02 = /.;/; // 驗任意字元
console.log(rules02.test(str02));
```

/a/：字串含 a。符合範例：「aa」、「dfas」

```
var str03 = 'hello world';

const rules03 = /w/; // 字串含w
console.log(rules03.test(str03));
```

/a./：字串含 a，以及其後面任一字元。不符合範例：「fa」

```
var str04 = 'hello world';

const rules04 = /e./; // 字串含 e，以及其後面任一字元。不符合範例：「we」(e不可以在最後一個字)
console.log(rules04.test(str04));
```

### 特定字元：

[A-Z]：大寫英文字母

```
var str05 = 'HELLO';
const rules05 = /[A-Z]/; // 大寫英文字母
console.log(rules05.test(str05));
```

[a-z]：小寫英文字母

```
var str06 = 'hello';  
const rules06 = /[a-z]/; // 小寫英文字母  
console.log(rules06.test(str06));
```

[0 - 9]：數字 0 - 9

```
var str07 = '2359';  
const rules07 = /[0-9]/; // 數字0-9  
console.log(rules07.test(str07));
```

\d：數字

\D：非數字

\w：數字字母與底線，等同 [A-Za-z0-9\_]

\W：非數字字母與底線，等同 [^A-Za-z0-9\_]

\s：空白字元

\S：非空白字元

/^em/：開頭是 em

/ma\$/：結尾是 ma

/^emma\$/：是否有 emma

/^em|ma\$/：開頭是 em，結尾是 ma

**次數：**

\*：0次以上

+：1次以上

?：0次或1次

{m}：m次

{n,}：最少n次

{m,n}：m次到n次

{m,n}?：m次到n次之間最少次的

練習一：常見密碼格式一

1. 至少 8 個字元
2. 要有大小寫字母
3. 至少一個數字

/^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)[a-zA-Z\d]{8,}\$/

```
var str08 = 'Ab12345678';
const rules08 = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{8,}$/;
console.log(rules08.test(str08));
```

(?=.\*[a-z]):

x(?=y) 的意思是有 x 且後面接 y

\* 則是任何字元出現 0 次以上的，[a-z] 則是小寫英文

## 練習二: 常見密碼格式二

1. 至少 8 個字元
2. 大小寫字母
3. 至少一個數字
4. 特殊符號

```
/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/
```

```
var str09 = 'Ab12345678@';
const rules09 = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/;
console.log(rules09.test(str09));
```

## 練習三: Email 格式

```
/^\w+([-+.'\w+)*@\w+([-+.'\w+)*\.\w+([-+.'\w+)*$/
```

```
//練習三: Email 格式
var str10 = 'Ab12345678@ddd.com';
const rules10 = /^\w+([-+.'\w+)*@\w+([-+.'\w+)*\.\w+([-+.'\w+)*$/;
console.log(rules10.test(str10));
```

## 練習四: 手機號碼

```
/^09\d{8}$/
```

```
//練習四: 手機號碼
var str11 = '0912345678';
const rules11 = /^09\d{8}$/;
console.log(rules11.test(str11));
```

## 練習五: 身分證字號

```
/^[A-Z]{1}[1-2]{1}[0-9]{8}$/
```

```
//練習五：身分證字號
var str12 = 'L123456789';
const rules12 = /^[A-Z]{1}[1-2]{1}[0-9]{8}$/;
console.log(rules12.test(str12));
```

RegExp	說明	範例
/a/	含有字母 a 的字串	"a","apple"
/./	含有任意字元的字串	"aaa","a","嗨"
/a./	含有字母 a 後面接一個任意字元的字串	"aaa","apple"

RegExp	說明	範例
/a*/	包含 0 次以上的 a	"apple","hello"
/ab*/	包含一個 a，後面至少 0 個 b	"a","ab"
/a?/	包含空字串或一個 a	"c","", "app"
/123a+ /	包含 123 後面有一個以上的 a	"123a","123app"
/123a{1,2}/	包含 123 後面出現一個或兩個 a	"1123a","123aaa"

RegExp	說明	範例
/^app/	開頭是 app 的字串	"app","apple"
/ry\$/	結尾是 ry 的字串	"Larry"
/^abcd\$/	開頭結尾中間只有 abcd 的字串	"abcd"
/^La.*le\$/	開頭是 La 尾巴是 le 的字串	"Larry loves apple"

RegExp	說明	範例
/^[aeiou]/	開頭是小寫母音的字串	"apple","oh"
/[^aeiouAEIOU]\$/	結尾不是母音的字串	"Larry","ok"
/^[aeiou]{3}\$/	三個小寫母音組成的字串	"aaa","aeu"
/^[^aeiou]*\$/	不包含小寫母音的字串	"hEllo","AppLe"
/[0-9]/	含數字的字串	"apple","123"
/[a-z]/	含小寫字母的字串	"12a45","aaa"
/^[^a-zA-Z]\$/	不含英文字母的字串	"123","345"

RegExp	說明	範例
--------	----	----

<code>/^\d{4}-\d{2}-\d{2}\$/</code>	西元生日格式	<code>"1996-08-06"</code>
<code>/^[A-Z]\d{9}\$/</code>	身分證字號	<code>"A123456789"</code>
<code>/^09\d{8}\$/</code>	手機號碼	<code>"0912345678"</code>
<code>/^[^aeiou]*\$/</code>	不包含小寫母音的字串	<code>"hEllo","ApplE"</code>
<code>/^.*@gmail\.com\$/</code>	gmail 信箱	<code>"test@gmail.com"</code>
<code>/^[0-9\+\-\*\\/]*\$/</code>	四則運算算式	<code>"1+2*3"</code>