### [From Gossip@Openhome](https://openhome.cc/Gossip/)

# [Java Gossip: 使用正則表示式（Regular expression）](https://openhome.cc/Gossip/JavaGossip-V1/index.html)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 如果您查詢J2SE 1.4之後String的線上API手冊說明，您會發現有**matches()、replaceAll()**等方法，您所傳入的參數是**「正則表示式」（Regular expression）**的字串，正則表示式的功能是J2SE 1.4之後加入的新功能。  正則表示式最早是由數學家Stephen Kleene于1956年提出，主要使用在字元字串的格式比對，後來在資訊領域廣為應用，現在已經成為ISO（國際標準組織）的標準之一。  Java在J2SE 1.4之後開始支援正則表示式，您可以在API文件的 [java.util.regex.Pattern](http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html) 類別中找到支援的正則表示式相關資訊。  如果您使用String類別來配置字串物件，您可以使用簡易的方法來使用正則表示式，並應用於字串的比對或取代等動作上，以下先介紹幾個簡單的正則表示式。  例如一些常用的範圍，我們可以使用預先定義的字元類別：   |  |  | | --- | --- | | . | 符合任一字元 | | \d | 等於 [0-9] 數字 | | \D | 等於 [^0-9] 非數字 | | \s | 等於 [ \t\n\x0B\f\r] 空白字元 | | \S | 等於 [^ \t\n\x0B\f\r] 非空白字元 | | \w | 等於 [a-zA-Z\_0-9] 數字或是英文字 | | \W | 等於 [^a-zA-Z\_0-9] 非數字與英文字 |   . 符合任一字元。例如有一字串abcdebcadxbc，使用.bc來比對的話，符合的子字串有abc、ebc、xbc三個；如果使用..cd，則符合的子字串只有bcd。  以上的例子來根據字元比對，您也可以使用**「字元類」（Character class）**來比較一組字元範圍，例如：   |  |  | | --- | --- | | [abc] | a、b或c | | [^abc] | 非a、b、c的其它字元 | | [a-zA-Z] | a到z或A到Z（範圍） | | [a-d[m-p]] | a到d或m到p（聯集） | | [a-z&&[def]] | d、e或f（交集） | | [a-z&&[^bc]] | a到z，除了b與c之外（減集） | | [a-z&&[^m-p]] | a到z且沒有m到p（a-lq-z）（減集） |   一次只指定一個字元不過癮，也可以用**Greedy quantifiers**來指定字元可能出現的次數：   |  |  | | --- | --- | | X? | X出現一次或完全沒有 | | X\* | X出現零次或多次 | | X+ | X出現一次或多次 | | X{n} | X出現n次 | | X{n,} | X出現至少n次 | | X{n,m} | X出現至少n次，但不超過m次 |   另外還有**Reluctant quantifiers**、**Possessive quantifiers**等的指定，您可以自行參考 [*java.util.regex.Pattern*](http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html) 類別中的說明。  在String類別中，**matches()**方法可以讓您驗證字串是否符合指定的正規表示式，這通常用於驗證使用者輸入的字串資料是否正確，例如 電話號碼格式；**replaceAll()**方法可以將符合正規表示式的子字串置換為指定的字串；**split()**方法可以讓您依指定的正規表示式，將符合的子 字串分離出來，並以字串陣列傳回。   下面這個程式示範幾個正則表示式的應用：   * UseRegularExpression.java   import java.util.Scanner;   public class UseRegularExpression {   public static void main(String args[]) {   Scanner scanner = new Scanner(System.in);    String str = "abcdefgabcabc";   System.out.println(str.replaceAll(".bc", "###"));     System.out.print("輸入手機號碼: ");   str = scanner.next();    // 簡單格式驗證   if(str.matches("[0-9]{4}-[0-9]{6}"))   System.out.println("格式正確");   else   System.out.println("格式錯誤");    System.out.print("輸入href標籤: ");   // Scanner的next()方法是以空白為區隔  // 我們的輸入有空白，所以要執行兩次  str = scanner.next() + " " + scanner.next();    // 驗證href標籤   if(str.matches("<a.+href\*=\*['\"]?.\*?['\"]?.\*?>"))  System.out.println("格式正確");   else  System.out.println("格式錯誤");    System.out.print("輸入電子郵件: ");   str = scanner.next();    // 驗證電子郵件格式   if(str.matches(  "^[\_a-z0-9-]+([.][\_a-z0-9-]+)\*@[a-z0-9-]+([.][a-z0-9-]+)\*$"))  System.out.println("格式正確");   else  System.out.println("格式錯誤");   }  }  執行結果：   |  | | --- | | ###defg######  輸入手機號碼: 0988-100432  格式正確  輸入href標籤: <a href="https://openhome.cc">  格式正確  輸入電子郵件: justin@openhome.cc  格式正確 |   最後兩個href標籤與email驗證例子是很常見的正規表示式應用，您可以仔細看看以瞭解如何比對，並不會特別困難。 |  |