

一、萬用字元是什麼？

在勾選「使用萬用字元」後主要會發生兩個變化。

1. 「大小寫須相符」、「全字拼寫須相符」、「前置詞須相符」、「後置詞須相符」、「全半形須相符」這五個無法勾選。其中「大小寫須相符」、「全半形須相符」這兩個是變成固定開啟。



2. 部分「指定方式」有變動，有些是勾選後無法使用，有些是改變名字。比如說「^p」、「^g」無法使用，「^?」變為「?」。

【REF】 <https://home.gamer.com.tw/artwork.php?sn=5505075>

二、使用介紹。

1. 搜尋數量「1」的字元「?」

「?」表示的是任何一個字：

輸入「?糴」可以找到「麻糴」、「馬糴」、「A 糴」。

輸入「麻?糴」可以找到「麻 A 糴」、「麻麻糴」，不能找到「麻糴」。

2. 搜尋數量「大於或等於 0」的字元「*」

「*」表示的是任何數量的字，包括 0 個字：

輸入「*糴」可以找到「糴」、「麻糴」、「麻 A 糴」、「A 麻 B 糴」。

輸入「麻*糴」可以找到「麻糴」、「麻 A 糴」、「麻 ABC 糴」。

輸入「麻*」可以找到「麻」、「麻糴」、「麻 A 糴」、「麻 A 糴 B」。

使用「*」一定要注意，像是第一和三個例子的「*糴」、「麻*」是很難使用的，因為沒有給它起始和結尾的限制，

以「*糴」來說，這樣就會直接從文章的開頭一直尋找到第一個「糴」，然後再尋找到第二個「糴」。

要記住「*」的前後都要有字包著才不會有範圍過大的問題。

3. 字首「<」

這個就是無法勾選的「前置詞須相符」。

輸入「<app」會找到在開頭的「app」，
「apple」的「app」，不會找到「bapp」的「app」。

「<」一定要輸入在開頭，雖然不輸入在開頭也能使用，
但是不可能搜尋的到東西，
比如說輸入「a<pple」，不可能尋找到「a」在「pple」前面，
但同時「pple」又是開頭的詞。



4. 字尾「>」

這個就是無法勾選的「後置詞須相符」。

輸入「le>」會找到在結尾的「le」，像是「apple」的「le」，
不會找到「applex」的「le」。

和「<」同理，「>」一定要輸入在結尾，
比如說輸入「app>le」，不可能找到必須在詞的最後的「app」的後面又有「le」
的詞。

字首和字尾可以一起使用，一起使用的話功能就像是無法勾選的「全字拼寫須相符」：

輸入「<app>」只會找到「app」，
不會找到「apple」的「app」、「appapp」的「app」，
因為「<app>」代表的是尋找位於開頭的「app」，且位於結尾的「app」，
「appapp」看似應該要被找到，
但其實位於開頭和結尾的「app」不是同一個，所以不會被找到。

5. 範圍內的字元「[]」

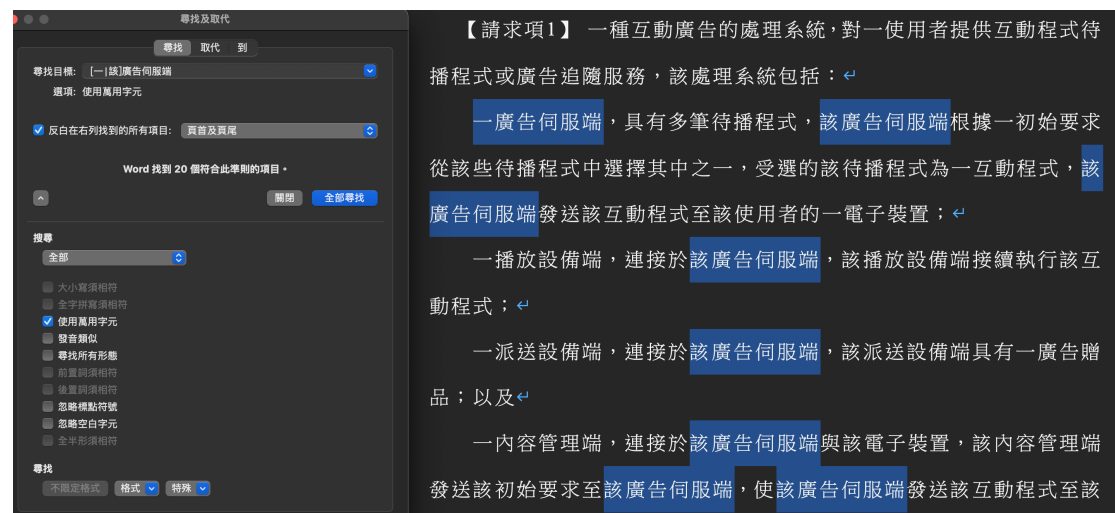
用來界定所指定的範圍，範圍的種類可以是數字、符號或字元

5.1 用法 1

加入"|"的 or 符號，用於搜尋「[a|b]」中字元 a 或字元 b

範例：

[一|該]元件



5.2 用法 2

加入"-"的串接範圍的符號，用於搜尋串接範圍內的字元或數字

範例

「[0-9]」為搜尋所有數字，最小值一定要最前面。

「[a-z]」為尋找所有小寫英文字母，起始的字元要放置最前面的字元。

「[A-Z]」為尋找所有大寫英文字母。

那可不可以直接簡化成「[A-z]」呢？

答案是要求準確的話不行，

因為「Z」和「a」還有著「[\]^_」這幾個符號，

所以簡化成「[A-z]」有可能會搜尋到那些符號。

可以一次用更多，比如說「[A-Za-z0-9]」的意思就是「大寫字母」或是「小寫字母」或是「數字」。

如果要搜尋中文內容，可以採用「[一-龜]」

[一-龜]的範圍基本上涵蓋常用的中文字，除非遇到自定義字型跟超冷門字。

5.3 用法 2

在中括號[]內部的開頭加上「!」，
比如說「[!a]」。加上驚嘆號後的功能和沒加的是相反的，
驚嘆號是「不等於」的意思，所以「[!a]」就是所有不為「a」的字元。

輸入「[!a]pple」不會找到「apple」，
而是會找到「bapple」或是「capple」等等的。

6. 發生次數「{a,b}」。

a 和 b 是為了方便說明而給的代號，實際輸入時是要輸入數字。

這功能是設定要尋找的字出現的次數，
a 代表的是最小次數，b 代表的是最大次數。
a 的最小值要是 1，b 必須大於等於 a，b 省略不輸入就是無限大的意思。

用法是放在欲查詢的字元後面，
比如說「ap{1,2}le」，這串的意思是「p」這個字出現 1 次或 2 次的話就會被找到，可以找出「apple」和「appale」。

9. 運算式「()」。

9.1

「()」的功能是對字元的群組化，使用時只能配合「{}」，
其餘一切完全沒有效果，
「()」可以和所有的功能混用，像是「[]」、「*」等等的。

範例

「麻糬{1,2}」的意思是「糬」這個字要出現 1 到 2 次，
如果加了小括號「(麻糬){1,2}」就變成「麻糬」這兩個字要出現 1 到 2 次，都會被列入搜尋的結果中。

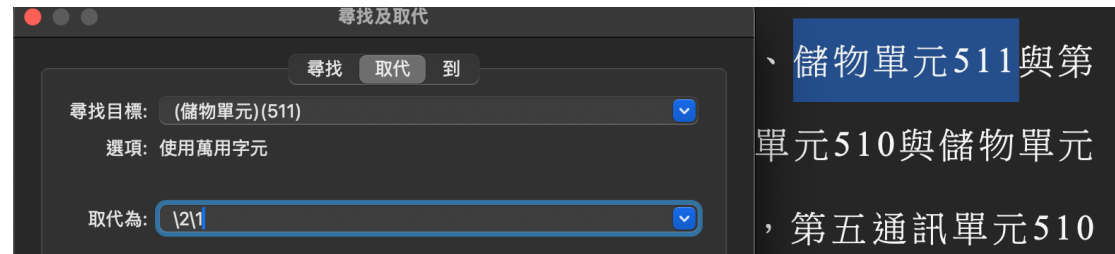
9.2

「()」可以實現批量取代的處理
在尋找及取代的視窗中
"尋找目標"的欄位中記錄多組由「()」所形成的字串群組
各字串群組根據順序自動賦予一組編號，編號的順序從 1、2、3....依此增加。

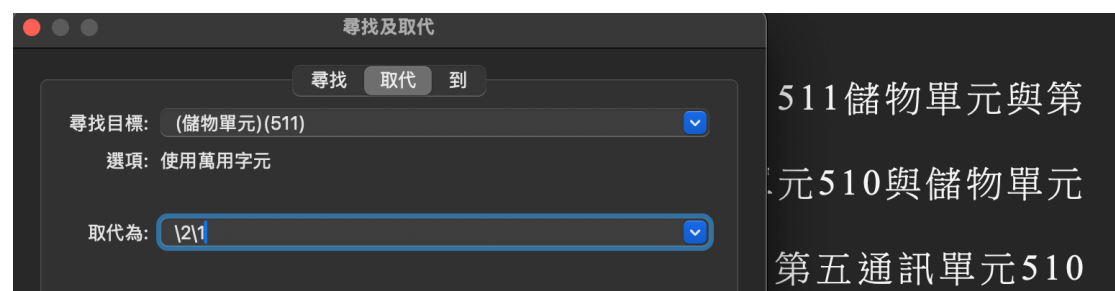
在"取代為"的欄位中可以直接採用前述編號作為取代時的替代，
填寫時要在編號前面加上「\」

範例

取代前



取代後



這個功能是用在尋找目標要保留的時候，或者是順序要改變的時候。假如現在有一串電話號碼「0912345678」，我想要將電話號碼最後 3 個數字改為「X」，但是我沒有辦法直接尋找到最後 3 個數字，所以需要尋找整串電話號碼，保留前 7 個，改掉後 3 個。

「尋找目標」輸入「([0-9]{7,7})[0-9]{3,3}」，取代為輸入「\1XXX」，這樣就可以保留「0912345678」的「0912345」，並將「678」給改為「XXX」了。

尋找及取代



第 6 點說的「有些情況尋找目標會很長」就像是上面的情況，你如果不想使用「{ }」的話當然也可以，但就會非常長「([0-9][0-9][0-9][0-9][0-9][0-9][0-9])[0-9][0-9][0-9]」。

改變順序情況有可能是日期的格式轉換，要把 1 月 15 日「1/15」改成「15/1」時，這時候「尋找目標」要輸入「([0-9]{1,2})/([0-9]{1,2})」，「取代為」要輸入「\2/\1」，這個意思是尋找「(出現 1 到 2 次的數字)/(出現 1 到 2 次的數字)」，取代為第二個括號的內容後接著斜線，再接著第一個括號的內容。

找標號用

尋找目標的欄位：

(AAA)([!0-9])

AAA 是元件

取代為的欄位：

\1BBB\2

BBB 是元件的標號