

1 Expresions Regulares en Linux

Unha **expresión regular** é un patrón que nos permite buscar un texto formado por metacaracteres e caracteres ordinarios.

Aquí tedes unha lista de metacaracteres que usamos en expresións regulares:

- **[]** : Un calquera dos caracteres entre os corchetes. Exemplos:
 - **[aeiou]** : Unha vocal minúscula.
 - **[A-Z0-9]** : Unha letra maiúscula ou unha cifra.
- **[^]** : Calquera carácter distinto dos que figuran entre corchetes. Exemplos:
 - **[^0-9]** : Calquer carácter que non sexa unha cifra (unha letra, un símbolo, etc).
- **{ }** : Permítenos indicar o número de repeticións do patrón anterior que deben darse. Algúns exemplos:
 - **[0-9]{5}** : Representa un número de 5 díxitos.
 - **[a-zA-Z]{2,4}** : Representa unha palabra que ten entre dous e catro caracteres.

Máis metacaracteres:

- ***** : Indica que o elemento que lle precede debe estar cero ou mais veces. Nótese que este carácter ten distinto significado que cando é carácter comodín. Por exemplo en **ls ab*c** móstranse os nomes de ficheiros que comezan por **ab**, teñen cero ou mais caracteres e rematan en **c**.
- **+** : Indica unha ou mais repeticións do carácter anterior.
- ***?, +?, ??** : Os metacaracteres **'*', '+', e '?'** seleccionan todo o texto posible. É dicir, moitas veces non desexamos que devolva toda a frase onde existe esa coincidencia. Se temos o patrón **<.*>** e devolve

```
<H1>title</H1>
```

, pero nos queremos que devolva

```
<H1>
```

, temos que engadir o metacarácter **?** xustamente despois, do seguinte xeito: **<.*?>**, así devolverá simplemente

```
<H1>
```

.

- **.** : Concorda cun carácter.
- **\$** : Se aparece ao final da expresión significa fin de liña.
- **^** : Se aparece ao principio da expresión significa principio de liña.
- **|** : Permítenos indicar caracteres alternativos.
- **()** : Permítenos agrupar patróns.
- **** : Carácter de escape. Elimina ou dalle un significado especial ao carácter que lle segue. Vexamos os máis empregados:
 - **<** : Indica o comenzo dunha palabra.
 - **>** : Indica o final dunha palabra.
 - **\t** : Representa un tabulador.
 - **\r** : Representa o "retorno de carro" ou "regreso ao inicio", é dicir, o lugar onde a liña volve a iniciar.
 - **\n** : Representa a "nova liña" o carácter por medio do cal unha liña da inicio. Recordade que en Windows é necesaria unha combinación de **\r\n** para comezar unha nova liña, mentres que en Unix só se emprega **\n** e en Mac_OS clásico utilízase só **\r**.
 - **\a** : Representa unha "campana" ou "beep" que se produce ao imprimir este carácter.
 - **\e** : Representa a tecla "Esc" ou "Escape".
 - **\f** : Representa un salto de páxina.
 - **\v** : Representa un tabulador vertical.
 - **\x** : Emprégase para representar caracteres ASCII ou ANSI coñecendo o seu código. Por exemplo, se queremos buscar o símbolo de dereitos de autor é posible atopalo utilizando **\xA9**.
 - **\u** : Emprégase para representar caracteres Unicode coñecendo o seu código. Por exemplo **\u00A2** representa o símbolo de centavos.
 - **\d** : Representa un dígito do 0 ao 9.

- \w : Representa cualquier carácter alfanumérico.
- \s : Representa un espazo en branco.
- \D : Representa cualquier carácter que non sexa un dígito do 0 ao 9.
- \W : Representa cualquier carácter NON alfanumérico.
- \S : Representa cualquier carácter que NON sexa un espazo en branco.
- \A : Representa o inicio da cadea. Non un carácter senón unha posición.
- \Z : Representa o final da cadea. Non un carácter senón unha posición.
- \b : Marca o inicio e o final dunha palabra.
- \B : Marca a posición entre dous caracteres alfanuméricos ou dous non-alfanuméricos.

Outro método alternativo para especificar un rango de caracteres é o seguinte:

- [:alnum:] : Representa caracteres numéricos ou alfanuméricos. Equivalente a A-Za-z0-9.
- [:alpha:] : Representa caracteres alfanuméricos. Equivalente a A-Za-z.
- [:blank:] : Representa un "espazo" ou un "tab".
- [:cntrl:] : Representa caracteres de control.
- [:digit:] : Representa a un dígito. Equivalente a 0-9.
- [:graph:] : Significa *graphic printable characters* e representa caracteres no rango do ASCII 33 - 126. Isto é o mesmo que [:print:] pero sen contar o "espazo".
- [:lower:] : Representa aos caracteres alfabéticos en minúscula. Equivalente a a-z.
- [:print:] : Representa caracteres no rango ASCII 32 - 126. É equivalente a [:graph:] pero, ademais, tamén inclúe o "espazo".
- [:space:] : Representa caracteres en branco ("espazo" e "tab").
- [:upper:] : Representa aos caracteres alfabéticos en maiúsculas. Equivalente a A-Z.
- [:xdigit:] : Representa a díxitos hexadecimais. Equivalente a 0-9A-Fa-f.

Exemplos máis complexos de expresións regulares:

- Buscar palabras nun texto: \b[a-zA-Z]+\b
- A expresión regular (^|[?&])parametrox=([^&])+ significa que buscamos algo que:

^[?&] __ O seguinte que aparece está a principio de liña (^) OU (|) que o buscado comeza por un carácter ? ou un &
 parametrox= __ sería o principio do buscado ou o que segue aos caracteres antes vistos
 [^&]+ __ e o buscado remata por un ou máis caracteres (+) que non sexan & (^&)

- Buscar dúas palabras cercanas nun párrafo, é dicir, que están separadas entre 1 e 8 palabras:

\bpalabra1\W+(\w+\W+){1,8}palabra2\b
 \bpalabra1 __ Indica que se busca un comezo de palabra coa sucesión de caracteres **palabra1**
 \W+ __ seguido por un ou varios caracteres non alfanuméricos \W
 (\w+\W+){1,8} __ seguido de entre 1 e 8 grupos {1,8} de un ou máis caracteres alfanuméricos \w e un ou máis caracteres non alfanuméricos \W
 palabra2\b __ acabado na sucesión de caracteres **palabra2**.

Está claro que, por suposto, pode ser que non importe que **palabra1** vai antes que **palabra2**, así que podemos especificar que unha cousa ou | a outra:

\b(palabra1\W+(\w+\W+){1,8}palabra2|palabra2\W+(\w+\W+){1,8}palabra1)\b

Enlace interesante

Archivos para exercicios:

◇ [historia.txt](#)