**Expresiones Regulares y MetaCaráteres**

Una expresión regular es un patrón que nos permite buscar un texto   
formado por metacaracteres y caracteres ordinarios.   
  
Los metacaracteres son ciertos caracteres con un significado   
específico dentro de una expresión regular. Estos caracteres tienen un   
significado que va más allá del símbolo que representan y tienen un   
comportamiento especial en una expresión regular.   
  
Aquí tenéis una lista de metacaracteres que usamos en expresiones   
regulares:   
  
  
. Significa cualquier caracter.   
^Indica el principio de una línea.   
$ Indica el final de una línea.   
\* Indica cero o más repeticiones del caracter anterior.   
+ Indica una o más repeticiones del caracter anterior.   
  
\< Indica el comienzo de una palabra.   
  
\> Indica el final de una palabra.   
\ Caracter de escape. Da significado literal a un metacaracter.   
[ ] Uno cualquiera de los caracteres entre los corchetes. Ej: [A-Z]   
(desde A hasta Z).   
[^ ] Cualquier caracter distinto de los que figuran entre corchetes:   
Ej: [^A-Z].   
  
{ } Nos permiten indicar el número de repeticiones del patrón anterior   
que deben darse.   
| Nos permite indicar caracteres alternativos: Ej: (^|[?&])   
( ) Nos permiten agrupar patrones. Ej: ([0-9A-F]+:)+   
Ojo. En las expresiones regulares se distingue entre mayúsculas y   
minúsculas.   
  
Si queremos representar un caracter entre la a y la z, lo haremos de   
la siguiente manera: [a-z]   
Dentro del conjunto, podemos especificar todos los caracteres que   
queramos. Por ejemplo: [a-zA-Z] representaría los caracteres   
alfabéticos en minúsculas y mayúsculas. Eso sí. El conjunto representa   
a un sólo caracter.   
Si lo que queremos es representar identificar un número o una letra,   
podríamos hacerlo así:   
[a-zA-Z0-9]   
  
Los conjuntos pueden representarse, nombrando todos y cada uno de los   
elementos, o el intervalo. Ej: [0-9] representa lo mismo que   
[0123456789].   
  
Si queremos representar un número que se compone de cero o más   
dígitos: [0-9]\*   
  
Y si queremos representar un número que se compone de uno o más   
dígitos: [0-9]+   
  
Si ahora queremos representar cualquier caracter menos los dígitos:   
[^0-9]   
  
Ahora, imaginemos que queremos representar un número de 5 dígitos:   
[0-9]{5}   
  
Y si quisieramos representar una palabra que tiene entre dos y cuatro   
caracteres: [a-zA-Z]{2,4}   
  
Dentro de los conjuntos de caracteres individuales, se reconocen las   
siguientes categorías:   
[:alnum:] alfanuméricos   
[:alpha:] alfabéticos   
[:cntrl:] de control   
[:digit:] dígitos   
[:graph:] gráficos   
[:lower:] minúsculas   
[:print:] imprimibles   
[:punct:] de puntuación   
[:space:] espacios   
[:upper:] mayúsculas   
[:xdigit:] dígitos hexadecimales   
  
Vamos a ver algunos ejemplos de expresiones regulares:   
  
# grep '^La' fichero   
El comando anterior nos devuelve todas las líneas del fichero que   
comienzan por La.   
  
# grep '^ \*La' fichero   
El comando anterior nos devuelve todas las líneas del fichero que   
comienzan por cualquier número de espacios seguido de La.   
  
# grep '^\..\*' fichero   
El comando anterior nos devuelve todas las líneas del fichero que   
comienzan por punto y tienen cualquier número de caracteres.   
  
# ls -la | grep ' \..\*'   
El comando anterior nos devuelve la lista de ficheros que comienzan   
por un espacio seguido de un punto y cualquier número de caracteres,   
es decir, la lista de ficheros ocultos.   
  
# ls -l | grep '^d'   
El comando anterior nos devuelve la lista de ficheros que comienzan   
por d, es decir, la lista de directorios.