

Expresiones Regulares



¿Qué son las expresiones regulares?

- A menudo llamadas **regex** con secuencias de caracteres que forman un patrón de búsqueda.
- Se formalizan con una sintaxis específica.
- Se interpretan como conjunto de instrucciones que luego se ejecuta sobre le texto de entrada.
- Es un mini lenguaje por si mismo.



¿Para qué expresiones regulares?

- Procesamiento de texto complejo
- Poderoso lenguaje utilizado para reconocer patrones
- Facilita operaciones de extracción, validación, búsqueda y sustitución
- Direcciones de correo electrónico, números de teléfono, validar campos de entrada, o una letra mayúscula seguida de dos minúsculas y de 5 dígitos entre 1 y 3



Componentes

- **Literales:** Cualquier carácter que se encuentra a si mismo. Ejemplo.- "raul" encontrara todas las pariciones de "raul" en el texto
- **Secuencias de escape:** Secuencias de escape que ya conocemos de otros lenguajes
- **Clases de caracteres:** Se pueden especificar encerrando una lista entre corchetes []. Si seguido tenemos el símbolo ^, se encontrara cualquier carácter que **no** este en la lista.

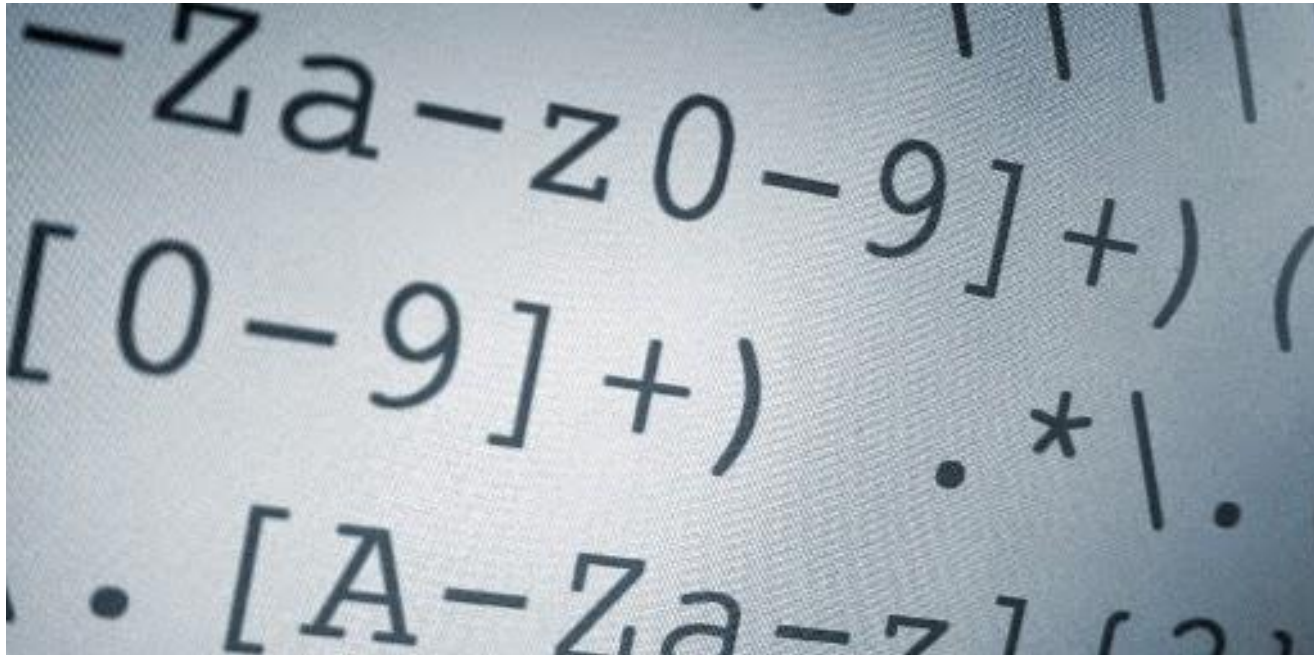


Secuencia de escape	Significado
<code>\n</code>	Nueva línea (new line). El cursor pasa a la primera posición de la línea siguiente.
<code>\t</code>	Tabulador. El cursor pasa a la siguiente posición de tabulación.
<code>\\</code>	Barra diagonal inversa
<code>\v</code>	Tabulación vertical.
<code>\ooo</code>	Carácter ASCII en notación octal.
<code>\xhh</code>	Carácter ASCII en notación hexadecimal.
<code>\xhhhh</code>	Carácter Unicode en notación hexadecimal.



Metacaracteres

- Caracteres especiales de suma importancia. Existen diferentes tipos ya que nos ayudan a entender la sintaxis de las **regex**.





Meta - Delimitadores

Metacaracter	Descripción
^	inicio de línea.
\$	fin de línea.
\A	inicio de texto.
\Z	fin de texto.
.	cualquier caracter en la línea.
\b	encuentra límite de palabra.
\B	encuentra distinto a límite de palabra.



Meta – Clases predefinidas

Metacaracter	Descripción
<code>\w</code>	un caracter alfanumérico (incluye "_").
<code>\W</code>	un caracter no alfanumérico.
<code>\d</code>	un caracter numérico.
<code>\D</code>	un caracter no numérico.
<code>\s</code>	cualquier espacio (lo mismo que [<code>\t\n\r\f</code>]).
<code>\S</code>	un no espacio.



Meta - Iteradores

Metacaracter	Descripción
*	cero o más, similar a {0,}.
+	una o más, similar a {1,}.
?	cero o una, similar a {0,1}.
{n}	exactamente n veces.
{n,}	por lo menos n veces.
{n,m}	por lo menos n pero no más de m veces.
*?	cero o más, similar a {0,}?
+?	una o más, similar a {1,}?
??	cero o una, similar a {0,1}?
{n}?	exactamente n veces.
{n,}?	por lo menos n veces.
{n,m}?	por lo menos n pero no más de m veces.



Meta - Alternativas

- Marca una serie de alternativas para una plantilla utilizando “|” para separarlas.

Ejemplo:

`foo(bar|foo) --> encuentra las cadenas 'foobar' o 'foofoo'.`



Meta - Subexpresiones

- La construcción (...) puede ser utilizada para definir sub-expresiones de las expresiones regulares.

Ejemplos:

`(foobar){10}` --> encuentra cadenas que contienen 8, 9 o 10 instancias de 'foobar'

`foob([0-9]|a+)r` --> encuentra 'foob0r', 'foob1r' , 'foobar', 'foobaar', 'foobaar' etc.



En python

import re