

Tecnológico Nacional de México
Campus Pachuca



A c t i v i d a d :

Expresión Regular

L e n g u a j e s A u t ó m a t a s

Catedrático: Rodolfo Baume Lazcano

Alumna:

- Morales Mateos Johanna N.C.: 21200619

Carrera: Ingeniería en Sistemas Computacionales.

19 m a r z o 2024

¿QUÉ ES?

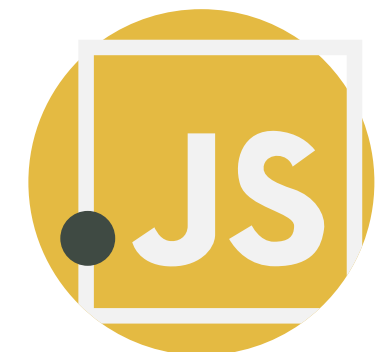
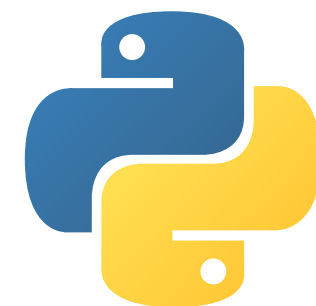
La definición exacta de una expresión regular se encuentra como: Una expresión regular, también conocida como regex es una secuencia de caracteres que actúa como modelo para la coincidencia y manipulación de series



¿CÓMO FUNCIONAN LAS EXPRESIONES REGULARES?

Una expresión regular puede estar formada, o bien exclusivamente por caracteres normales (como abc), o bien por una combinación de caracteres normales y metacaracteres (como ab*c). Los metacaracteres describen ciertas construcciones o disposiciones de caracteres: por ejemplo, si un carácter debe estar en el inicio de la línea o si un carácter solo debe o puede aparecer exactamente una vez, más veces o menos.

Las expresiones regulares se pueden aplicar en diversos lenguajes, tales como Perl, Python, Ruby, JavaScript, XML o HTML, por lo que los usos o funciones pueden llegar a ser muy diferentes.



DE OTRA FORMA...

Las expresiones regulares son como si fueran trucos especiales que te ayudan a buscar palabras en un montón de letras. Si quieres encontrar palabras que tengan una letra específica al principio o al final, o incluso palabras que sigan un cierto patrón, las expresiones regulares te ayudarán a hacerlo más fácilmente.

¿De qué forma logra esto?

Por ejemplo, si quieres encontrar todas las palabras que terminan en "ar", podrías usar una expresión regular para decirle a un programa que busque todas las letras que terminan en "ar". Entonces, el programa te mostrará todas las palabras que terminan en "ar", como "hablar", "saltar", y muchas otras.



IMPORTANCIA



Son importantes debido a que, con ayuda de estas expresiones regulares, podemos establecer filtros, para buscar ciertos grupos con ciertas combinaciones de caracteres específicas y fácilmente ignorar el resto de información que podamos tener, con esto puedes crear un filtro para validar números telefónicos, emails, URL's, incluso combinándolas con algún lenguaje de programación se podrían crear muchas cosas más. Como podemos observar, estas expresiones regulares permiten realizar tareas relacionadas con el procesamiento de cadenas.

CASOS DE USO

A continuación ejemplos de caso de uso, es importante decir que los ejemplos que se muestran a continuación pueden ser útiles como punto de partida para crear otras expresiones regulares más complejas.

Ejemplo de caso de uso	Búsqueda de números de teléfono
Aplicación	Encontrar números de teléfono en un texto que sigan el formato XXX-XXX-XXXX.
Ejemplo regex	<code>^\d{3}-\d{3}-\d{4}\$</code>
Descripción	<div>1. <code>^</code>: Este símbolo indica que la cadena debe comenzar con el patrón que sigue a continuación.</div> <div>2. <code>\d{3}</code>: Aquí, <code>\d</code> es un atajo que coincide con cualquier dígito del 0 al 9. Entonces, <code>\d{3}</code> significa que esperamos exactamente tres dígitos.</div> <div>3. <code>-</code>: Este es un guion literal que debe aparecer después de los primeros tres dígitos.</div> <div>4. <code>\d{3}</code>: Igual que antes, esta parte busca tres dígitos adicionales después del guion.</div> <div>5. <code>-</code>: Otro guion literal que sigue después de los siguientes tres dígitos.</div> <div>6. <code>\d{4}</code>: Similar a los anteriores, este fragmento busca exactamente cuatro dígitos al final del patrón.</div> <div>7. <code>\$</code>: Este símbolo indica que la cadena debe terminar justo después de los últimos cuatro dígitos.</div>

Ejemplo de caso de uso	Validación de correos electrónicos
Aplicación	Verificar si una cadena de texto es una dirección de correo electrónico válida.
Ejemplo regex	<code>^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$</code>
Descripción	<p>1. <code>^</code>: Este símbolo indica que la cadena debe comenzar</p> <p>2. <code>[a-zA-Z0-9._%+-]+</code>: Esta parte de la expresión regular corresponde al nombre de usuario en la dirección de correo electrónico. Aquí, <code>[a-zA-Z0-9._%+-]</code> significa que se pueden usar letras mayúsculas y minúsculas, dígitos del 0 al 9, y ciertos caracteres especiales como ".", "_", "%", "+" y "-". El "+" al final significa que al menos uno de estos caracteres debe estar presente, y el conjunto completo puede repetirse una o más veces.</p> <p>3. <code>@</code>: Este símbolo, que separa el nombre de usuario del dominio en una dirección de correo electrónico.</p> <p>4. <code>[a-zA-Z0-9.-]+</code>: Aquí se define el dominio de la dirección de correo electrónico. Este patrón permite letras mayúsculas y minúsculas, dígitos del 0 al 9, y los caracteres "." y "-", que son comunes en los nombres de dominio. El "+" al final indica que al menos uno de estos caracteres debe estar presente, y el conjunto completo puede repetirse una o más veces.</p> <p>5. <code>\.</code>: Este es un escape que se utiliza para indicar que se espera un punto literal (".") en la dirección de correo electrónico.</p> <p>6. <code>[a-zA-Z]{2,}</code>: Aquí se especifica la extensión del dominio, como ".com", ".org", etc. <code>[a-zA-Z]</code> permite solo letras mayúsculas y minúsculas, y <code>{2,}</code> significa que al menos dos letras deben estar presentes, pero puede haber más.</p> <p>7. <code>\$</code>: Este símbolo indica que la cadena debe terminar justo después de la extensión del dominio.</p>

Ejemplo de caso de uso	Validación de códigos postales
Aplicación	Verificar si una cadena de texto representa un código postal válido, con o sin extensión de cuatro dígitos.
Ejemplo regex	<code>\b\d{5}(?:-\d{4})?\b</code>
Descripción	<p>1. <code>\b</code>: Este símbolo representa un límite de palabra, lo que significa que la coincidencia debe ocurrir al comienzo o al final de una palabra. Ayuda a asegurar que el patrón coincida con un código postal completo y no solo con una parte de él.</p> <p>2. <code>\d{5}</code>: Aquí, <code>\d</code> es un atajo para cualquier dígito del 0 al 9, y <code>{5}</code> indica que esperamos exactamente cinco de estos dígitos. Por lo tanto, este segmento busca un bloque de cinco dígitos consecutivos.</p> <p>3. <code>(?:-\d{4})?</code>: Este es un grupo de no captura (indica que no se va a capturar la coincidencia), que puede aparecer cero o una vez. Dentro de este grupo, <code>-\d{4}</code> busca un guion seguido de exactamente cuatro dígitos. Esto permite que el código postal tenga un guion seguido de un bloque de cuatro dígitos adicionales, pero el guion y los cuatro dígitos son opcionales debido al <code>?</code>.</p> <p>4. <code>\b</code>: Similar al primer <code>\b</code>, este símbolo representa otro límite de palabra al final del patrón.</p>

Ejemplo de caso de uso	Búsqueda de URLs
Aplicación	Encontrar enlaces web en un texto, tanto con http como con https.
Ejemplo regex	<code>https?://(?:[-\w.] (?:%[\da-fA-F]{2}))+</code>
Descripción	<p>1. https?: Esto coincide con "http" o "https". El s? indica que la "s" es opcional, lo que significa que la expresión regular coincidirá con "http" o "https".</p> <p>2. ://: Esto simplemente coincide con los dos puntos y las dos barras que siguen a "http" o "https".</p> <p>3. (?: ...)+: Este es un grupo de no captura que se repite una o más veces. El + al final significa que el grupo debe aparecer al menos una vez. Lo que está dentro del grupo (?: ...) se repite.</p> <p>4. [-\w.]: Dentro del grupo de no captura, esto coincide con un guion, un carácter de palabra o un punto. El - y el . están fuera del rango \w y por lo tanto no necesitan ser escapados.</p> <p>5. : Esto indica una alternativa, lo que significa que coincidirá con lo que está a la izquierda o a la derecha del operador.</p> <p>6. (?:%[\da-fA-F]{2}): Dentro del grupo de no captura, esto coincide con un porcentaje seguido de dos caracteres hexadecimales. Los \d coinciden con cualquier dígito decimal, y [a-fA-F] coincide con cualquier carácter hexadecimal (tanto en minúsculas como en mayúsculas).</p>

Ejemplo de caso de uso	Validación de direcciones IP
Aplicación	Verificar si una cadena de texto representa una dirección IP válida, como por ejemplo "192.168.1.1".
Ejemplo regex	<code>\b(?:\d{1,3}\.){3}\d{1,3}\b</code>
Descripción	<ol style="list-style-type: none">1. <code>\b</code>: Representa un límite de palabra, lo que significa que la coincidencia debe ocurrir al principio o al final de una palabra. Ayuda a garantizar que la dirección IP esté separada de otras palabras por espacios u otros caracteres no alfanuméricos.2. <code>(?: ...)</code>: Esto es un grupo de no captura, que agrupa un conjunto de elementos sin capturar la coincidencia. En este caso, se utiliza para agrupar un bloque específico de patrones que se repite.3. <code>\d{1,3}\.</code>: Aquí, <code>\d</code> coincide con cualquier dígito del 0 al 9, y <code>{1,3}</code> indica que esperamos entre 1 y 3 dígitos consecutivos. Después de esos dígitos, el <code>\.</code> coincide con un punto literal.4. <code>{3}</code>: Este cuantificador indica que el grupo anterior (que consiste en entre 1 y 3 dígitos seguidos de un punto) debe repetirse exactamente tres veces.5. <code>\d{1,3}</code>: Este patrón coincide con entre 1 y 3 dígitos consecutivos.6. <code>\b</code>: Otro límite de palabra, que asegura que la dirección IP esté separada de otras palabras por espacios u otros caracteres no alfanuméricos.

REFERENCIAS:

- Equipo editorial de IONOS. (2019, 30 diciembre). Regex o expresiones regulares: la manera más sencilla de describir secuencias de caracteres. IONOS Digital Guide. <https://www.ionos.mx/digitalguide/paginas-web/creacion-de-paginas-web/regex/>
- IBM documentation. (s. f.). <https://www.ibm.com/docs/es/i/7.3?topic=expressions-regular>
- Ricardocelis. (2018, 20 junio). ¿Qué son las expresiones regulares? Explicadas en video. Platzi. <https://platzi.com/blog/que-expresion-regular/>
- Ejemplos de expresiones regulares - Ayuda de Administrador de Google Workspace. (s. f.). <https://support.google.com/a/answer/1371417?hl=es>
- Goyvaerts, J. (s. f.). Regular-Expressions.info - Regex Tutorial, Examples and Reference - Regexp Patterns. <https://www.regular-expressions.info/>
- google. (s. f.). Syntax. GitHub. <https://github.com/google/re2/wiki/Syntax>