

# Expresiones Regulares

Python

18/05/18

# Ejemplos

**Formulario de registro** ×

**Email**

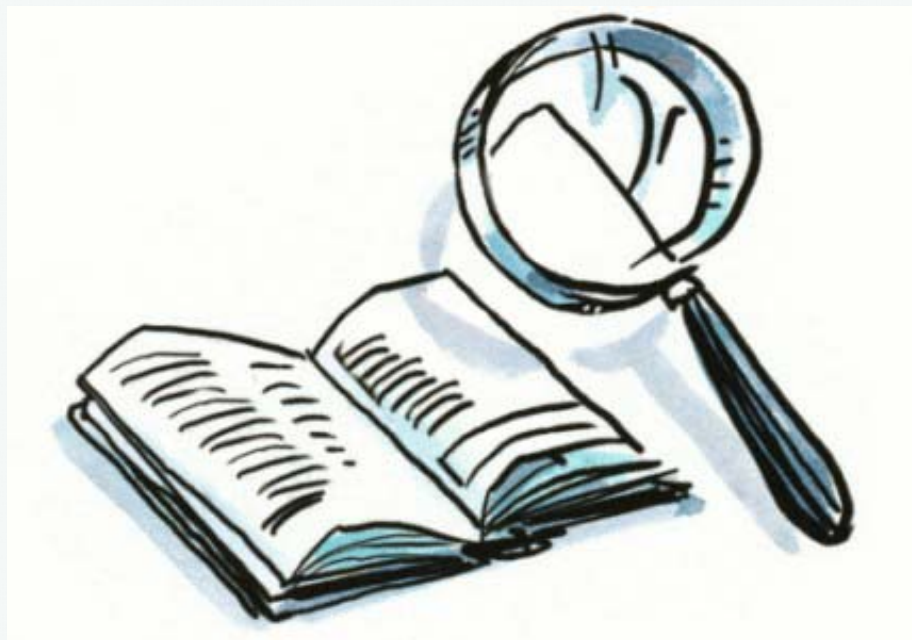
**Contraseña**

**Confirmar contraseña**

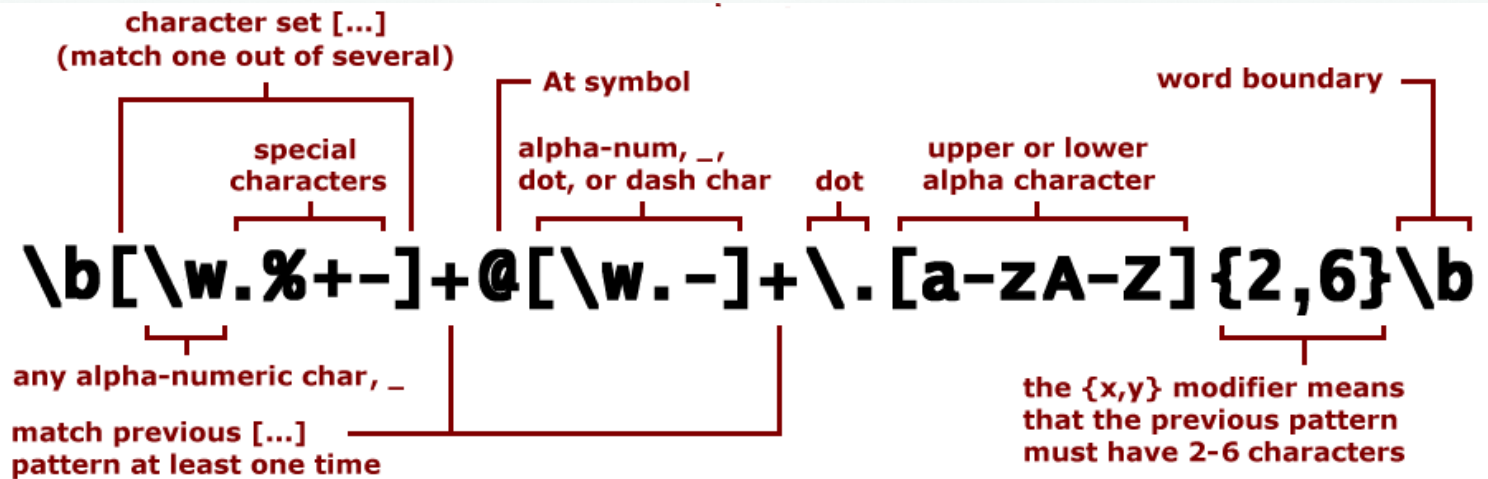
☒ Deseo recibir noticias, actualizaciones de software, y la última información relativa a productos y servicios.

**Registrarse**

# Ejemplos



# Ejemplos



**Parse: username@domain.TLD (top level domain)**





# ¿Que son?

- Son secuencias especiales de caracteres que ayudan a encontrar y comparar otras cadenas de texto.
- Son útiles para:
  - -Verificar que un String concuerde con un patrón
  - -Realizable substituciones en Strings
  -



# Funciones

- **re.match(patron, string)**
  - Se utiliza para ver si hay “match” al principio del texto
- **re.search(patron, string)**
  - Se utiliza para ver si hay match en cualquier parte del texto
- **re.findall(patron, String)**
  - Regresa una lista de todas las subcadenas que hayan hecho match
- **re.sub(patron, remplazo, rexta)**
  - Regresa el texto con el remplazo en vez del patron
  - 
  -



# Metacaracteres

- .
- \*
- +
- ?
- {n}
- {n,m}
- |
- ^
- \$
- []
- ( )
- \

Carácter	Significado
^	Principio de la cadena
\$	Final de la cadena
.	Cualquier carácter excepto salto de línea
*	Operador de repetición 0 o más veces
+	Operador de repetición 1 o más veces
?	Operador alternativo: una vez o ninguna
	Alternativa
( )	Agrupar expresiones
[ ]	Conjunto de caracteres
{ }	Modificador de repetición
\	Permite presentar un metacarácter como un carácter ordinario



# Especiales

## Caracteres especiales

Símbolo	Significado
.	Cualquier caracter, excepto \n
\d	Cualquier dígito
\D	Cualquier caracter que NO sea dígito
\w	Cualquier caracter alfanumérico
\W	Cualquier caracter que NO sea alfanumérico
\s	Espacio en blanco
\S	Cualquier caracter que NO sea espacio
\t	Tabulador
\n	Salto de línea (Unix)

