

Práctica de expresiones regulares

Se facilitan dos herramientas on line y un pdf con explicación breve de cómo usar regex101.

<https://www.debuggex.com/>

<https://regex101.com/>

- En las tablas que siguen, cada fila de la tabla corresponde a una expresión regular; cada columna, a una cadena de caracteres. Completar esas tabla escribiendo SI o NO en cada una de sus celdas, según si la cadena que encabeza la columna pertenece o no al lenguaje representado por la expresión regular que encabeza la fila. Puede utilizarse cualquiera de las herramientas en línea que se brindaron en este pdf y puede completarse el resultado con notas en el mismo.

	abc	bca	xxx	x	(xxx)	(z+	1a2b3	b34c	123*	pan
<code>[^abc]?</code>										
<code>([a][b][c])*</code>										
<code>^(abc)+</code>										
<code>\([^abc]\)*</code>										

	pa	Pan	PanN	PpAa	Pan*	Xan*	A	\.\.\	.\.*	.\an*
<code>(Pp)(Aa)(Nn)*</code>										
<code>[P\p][A\a][N\N]*</code>										
<code>\.\.\.*</code>										
<code>[^Aa](an)*</code>										

	(ab)	b7	b52s	[^ab]	(x	b2s	(pe	bas?	((xxx
<code>\([^ab]</code>										
<code>b[3-72]s</code>										
<code>(([a][b])*)+</code>										
<code>b[^0-59]s?</code>										

- Dado el siguiente código:

```
import re
string = 'bat, lat, mat, bet, let, met, bit, lit, mit, bot, lot, mot'
result = re.findall( )
print(result)
```

Encuentra la expresión para que se emita: ['bat', 'bot']

- Dado el siguiente texto:

""Maria tiene 5 años, y su hermana Valeria tiene 2.
Rita y Pedro, sus primos, tienen 3.""

Encuentra la expresión regular que extraiga y emita sólo los nombres.

4. Dado el siguiente texto:

```
texto = """Office of Research Administration: (734) 647-6333 | 4325 North Quad
Office of Budget and Financial Administration: (734) 647-8044 | 309 Maynard, Suite 205
Health Informatics Program: (734) 763-2285 | 333 Maynard, Suite 500
Office of the Dean: (734) 647-3576 | 4322 North Quad
UMSI Engagement Center: (734) 763-1251 | 777 North University
Faculty Administrative Support Staff: (734) 764-9376 | 4322 North Quad"""
```

Encuentra la expresión para que se emita: ['(734) 647-6333', '(734) 647-8044', '(734) 763-2285', '(734) 647-3576', '(734) 763-1251', '(734) 764-9376']

5. Dado el archivo pedidos.txt. Escribe un nuevo archivo 'pedidos_agrupados.txt', en el que los datos se reagrupen de la siguiente manera:

```
1289,T83456
1289,Z22334
1205,T10032
1205,B77301
1205,T10786
1205,C77502
1410,K34001
1410,T98987
```

Sugerencia para abrir el archivo:

```
txt = open("<ruta>/pedidos.txt").read()
```

Sugerencia para escribir el archivo:

```
open("pedidos_agrupados.txt", "w")
```

6. Dado dos archivos provincias.txt, que contiene un listado de las provincias de la república Argentina y localidades.txt, que contiene un listado de todas las localidades de la RA con su correspondiente código postal. En relación al ejemplo dado responder:

- Existe la posibilidad de encontrar todos los códigos postales que le corresponden a cada provincia utilizando expresiones regulares?.
- En el caso de respuesta afirmativa al punto anterior: La misma expresión regular del ejemplo dado es útil? O hay que cambiarla?.

Al construir el programa hay que tener en cuenta que lo que une a cada provincia con sus ciudades y pueblos es el id. Se facilitan dos archivos en formato csv y dos en formato .txt. Si desea utilizarlos como csv, en Python se debe incluir: import csv (referencia: <https://docs.python.org/3/library/csv.html>).