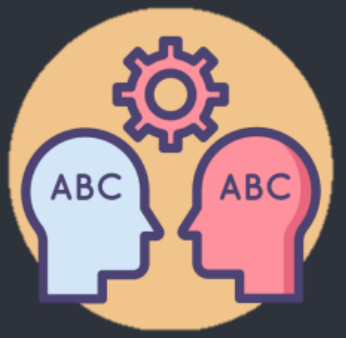
Procesamiento de Lenguaje Natural{



[NLP]

<Regular Expressions>

}

Ivan Rojas Gonzalez – Universidad Panamericana

La Agenda de hoy {

01 Regular Expressions

<Que son, como se usan>

02 Creando expresiones regulares

<El poder de Regex101>

03 El modulo Re

<Patrones en el texto>

}

Que son las Regular expressions?{

}

Que son las Regular expressions?{

<Que representa este patrón?>

Margarito@soniderolachanga.com

}

```
Que son las Regular expressions?{

    <Que representa este patrón?>

    Margarito@soniderolachanga.com

}
```

Que son las Regular expressions?{

<Que representa este patrón?>

+525599887766

}

Que son las Regular expressions?{

<Que representa este patrón?>

+525599887766

}

Que son las Regular expressions?{

<Que representa este patrón?>

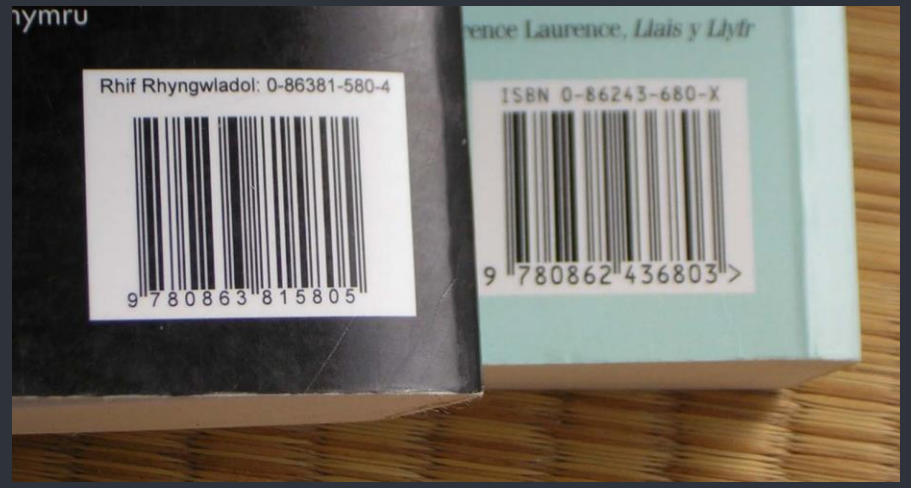
ROGI950424HDFJNV04

}

```
Que son las Regular expressions?{

    <Que representa este patrón?>

                ROGI950424HDFJNV04

}
```

Que son las Regular expressions?{

<Existen patrones en muchos elementos a nuestro alrededor>



}

{Como hacer código que pueda descubrir estos patrones?}

Ivan Rojas Gonzalez – Universidad Panamericana

```
Regular expressions{

    <Caracteres y simbolos que conforman un
    patrón que puede ser usado para hacer búsquedas
    en un cumulo de texto>

}
```

Cheat Sheet de Regex{

Caracter	Significado	Ejemplo
*	Match de cero, uno o mas del caracter previo	Ah* hace match con "Ahhhh" O "Ah"
?	Match de cero, o uno del caracter previo	Ah? hace match con "Ah" o "Al"
+	Match de uno o mas del caracter previo	Ah+ hace match con "Ah" pero no hace match con "Al"
\	Caracter de escape para caracteres especiales	Hola\? Hace match con "Hola?"
.	Comodin, hace match a cualquier caracter	Do. hace match con "dog", "door", "dot"
()	Agrupa caracteres	Este ejemplo lo veremos mas adelante
[]	Match a un rango de caracteres	[0-9] hace match con cualquier digito entre el 0 y el 9. [A-Z] hace match con cualquier letra mayuscula de la A a la Z

}

Cheat Sheet de Regex{

Caracter	Significado	Ejemplo
|	Hace match con el caracter (o grupo de caracteres) anterior o posterior	(Mon)|(Tues)day hace match con Monday o con Tuesday
{}	Hace match con un numero especifico de apariciones del caracter anterior	[0-9]{3} hace match con 3 digitos que vayan del 0 al 9. Como 123, 911
^	Match con string que comience con el patron indicado a la derecha. Al ser usado en rangos [] implica negacion	^http hace match con cualquier string que comience con http. [0-9] hace match con string que no comience con algun numero entre el 0 y el 9
$	Match con string que termine con el caracter especificado a la izquierda	ing$ hace match con palabras como "interesting" o "amusing"

}

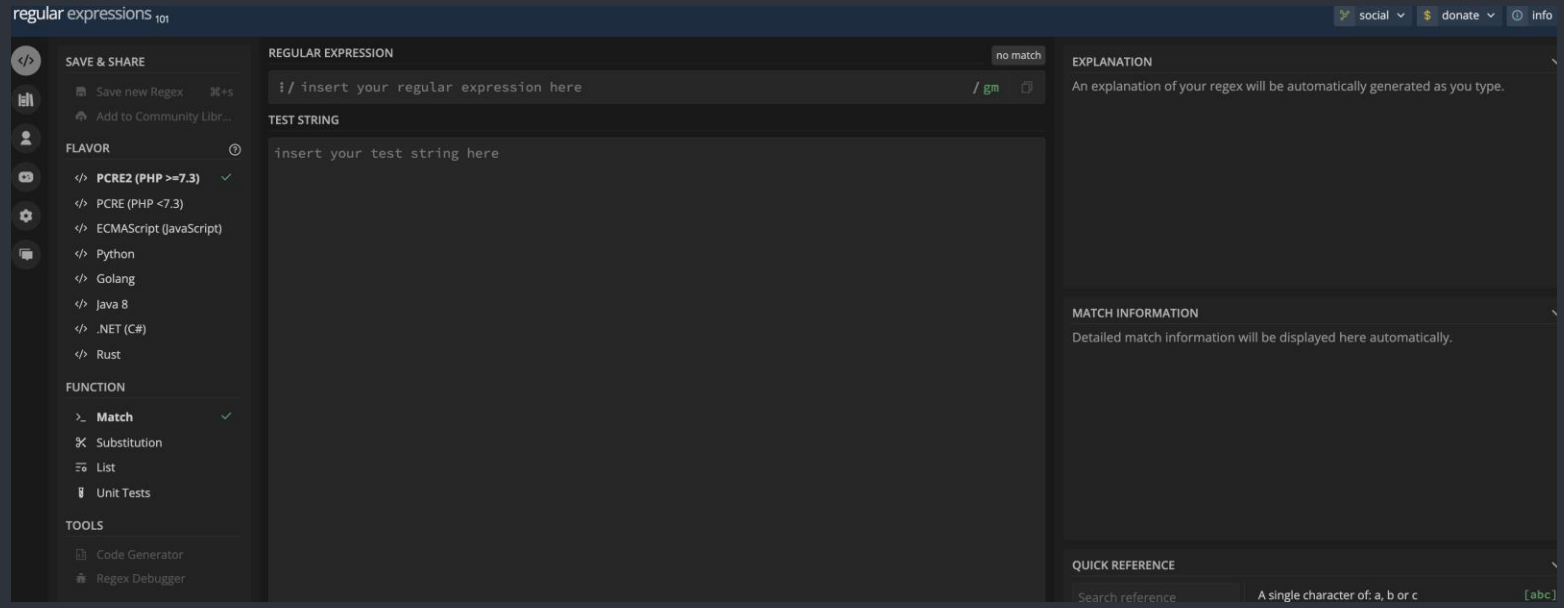
Caracteres especiales de Regex{

Caracter	Significado
\d	Hace match a cualquier digito. Es lo mismo que poner [0-9]
\D	Hace match a cualquier NO digito. Es lo mismo que poner [^0-9]
\s	Hace match a cualquier espacio en blanco
\S	Hace match a cualquier NO espacio en blanco
\w	Hace match a un "word character". Es lo mismo que poner [A-Za-z0-9]
\W	Hace match a un NO "word character". Es lo mismo que poner [^A-Za-z0-9]

}

Regular expressions{

<Antes de implementar Regex en codigo. Podemos validar nuestras expresiones en la pagina de Regex101>



}

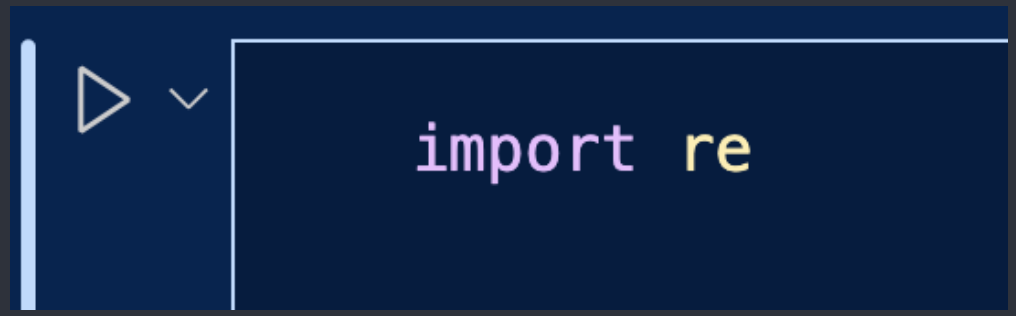
{Actividad}

{Implementando regex en codigo}

Ivan Rojas Gonzalez – Universidad Panamericana

Regex en Python{

<Python cuenta con un módulo llamado "re" que encapsula todas las operaciones relacionadas con expresiones regulares>



}

Regex en Python{

<re cuenta con las siguientes funciones. Aunque durante el curso estaremos haciendo uso más que nada de la funcion "findall" y "sub">

Nombre	Funcionamiento
findall	Regresa una lista que contiene todos los matches para una regex
search	Regresa un objeto de tipo "match" en caso de encontrar un match
split	Regresa una lista en la que cada elemento es separado por una coma en el lugar donde hace match la regex
sub	Reemplaza uno o varios matches por otro string

}

Regex en Python{

<Por ejemplo, si quisiéramos encontrar todos los teléfonos en un grupo de texto sabiendo que todos siguen el patrón +52 5500000000>

```
texto = '''
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Sed sed lacus a erat placerat facilisis. Nullam ac massa pharetra, posuere justo quis,
efficitur mauris. Phasellus blandit nunc ut sodales dignissim.
+52 5512233445
Nullam sed viverra nibh, cursus volutpat nisl. Mauris nec neque
eu nisl tempus dignissim. Integer sodales nisi vel nisl vol+525599778866utpat pretium.
Duis vitae augue sit amet elit posuere pretium at sed massa. Integer non nisl
ac purus consectetur ornare eget v+525599118822itae ex. Nulla facilisi. Vestibulum quis lobortis
nulla, id faucibus arcu. Ph+525510102222asellus tristique ipsum nec commodo luctus. Curabitur a ante varius,
facilisis nibh id, hendrerit arcu. Pellentesque non mollis tellus, ut laoreet odio.
'''
```

}

Regex en Python{

<Por ejemplo, si quisiéramos encontrar todos los teléfonos en un grupo de texto sabiendo que todos siguen el patrón +52 5500000000>

```
import re

re.findall(r'\+5255\d{8}',texto)
```

Regex a buscar (siempre incluida dentro las comillas de r' ')

Bloque de texto donde se buscará la regex

```
['+525599778866', '+525599118822', '+525510102222']
```

}

{Actividad}

Q&A