# **Expresiones regulares.**

import re

Empezaremos importando este módulo, que utilizamos para trabajar con las expresiones regulares, hay mas pero este es el más común.

import re

texto = """Hola que tal estas el dia de hoy, esta es la cadena 1

Esta es la segunda linea de texto

Esta es la final

"""

resultado = re.search()

Lo que haremos con “re.search” es buscar coincidencias.

resultado = re.search("Hola", texto)

Aquí lo que decimos es que busque “Hola” en la variable “texto”

Y nos devolverá esto, que dice que encontró el primer Hola.

resultado = re.findall("Esta", texto)

print(resultado)

Esto lo que hare es mostrarnos todos los elementos, en este caso los “Esta” en la variable “texto”



resultado = re.findall("Esta", texto,*flags*=re.**IGNORECASE**)

A su vez podemos agragar un segundo parámetro que es el “flags”, que es para usamos ciertos parámetros, en este caso usamos “re.IGNORECASE” que dice que se ignoren todas las mayúsculas y minúsculas.

Al ejecutarlo nos muestra todas estas coincidencias.

import re

texto = """Hola que tal estas el dia de hoy.

Esta es la cadena 1.

Esta es la linea 2.

Esta es la linea 3 y final.

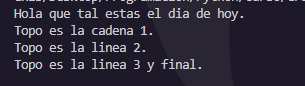
"""

reemplazo = "Topo"

resultado = re.sub(r"^Esta",reemplazo,texto,*flags*=re.**M**)

print(resultado)

Tenemos el re.sub que lo que hace es reemplazar un valor, en este caso estamos diciendo que en donde se encuentre “Esta” al inicio de una linea la reemplace por “Topo”.



La primera expresión regular que tenemos es:

* \d

Busca dígitos numéricos del 0 al 9

import re

texto = """Hola que tal estas el dia de hoy, esta es la cadena 1

Esta es la linea 2

Esta es la linea 3 y final

"""

resultado = re.findall(r"\d", texto)

print(resultado)

Para decir que vamos a usar expresiones regulares debemos poner la “r” como prefijo, como si fuera un f-string.



* \D

Busca todo menos dígitos del 0 al 9

import re

texto = """Hola que tal estas el dia de hoy, esta es la cadena 1

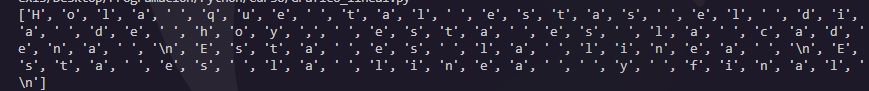
Esta es la linea 2

Esta es la linea 3 y final

"""

resultado = re.findall(r"\D", texto)

print(resultado)



* \w

Busca caracteres alfanuméricos [a-z, A-Z, 0-9, \_]

import re

texto = """Hola que tal estas el dia de hoy, esta es la cadena 1

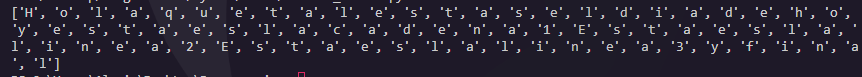
Esta es la linea 2

Esta es la linea 3 y final

"""

resultado = re.findall(r"\w", texto)

print(resultado)



* \W

Busca todo menos caracteres alfanuméricos

import re

texto = """Hola que tal estas el dia de hoy, esta es la cadena 1

Esta es la linea 2

Esta es la linea 3 y final

"""

resultado = re.findall(r"\W", texto)

print(resultado)



* \s

Busca espacios en blanco (espacios, tabulación, saltos de linea)

import re

texto = """Hola que tal estas el dia de hoy, esta es la cadena 1

Esta es la linea 2

Esta es la linea 3 y final

"""

resultado = re.findall(r"\s", texto)

print(resultado)



* \S

Busca todo menos espacios en blanco

import re

texto = """Hola que tal estas el dia de hoy, esta es la cadena 1

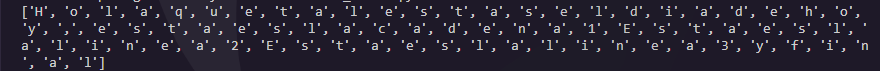
Esta es la linea 2

Esta es la linea 3 y final

"""

resultado = re.findall(r"\S", texto)

print(resultado)



* .

Busca todo menos saltos en linea

import re

texto = """Hola que tal estas el dia de hoy, esta es la cadena 1

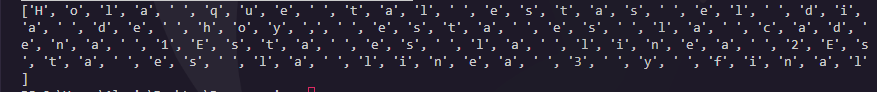
Esta es la linea 2

Esta es la linea 3 y final

"""

resultado = re.findall(r".", texto)

print(resultado)



* \n

Buscamos saltos en linea.

import re

texto = """Hola que tal estas el dia de hoy, esta es la cadena 1

Esta es la linea 2

Esta es la linea 3 y final

"""

resultado = re.findall(r"\n", texto)

print(resultado)



* \

Cancelamos caracteres especiales.

import re

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.

Esta es la linea 2.

Esta es la linea 3 y final.

"""

resultado = re.findall(r"\.", texto)

print(resultado)

Aquí cancelamos todo y solo buscamos los puntos.



Podemos combinar estas expresiones regulares:

Vamos a buscar un numero, seguido de un punto y un espacio:

import re

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.

Esta es la linea 2.

Esta es la linea 3 y final.

"""

resultado = re.findall(r'\d\.\s',texto)

print(resultado)



Sigue un patrón para darnos estos valores, si hiciese falta uno digamos el \n

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.Esta es la linea 2.Esta es la linea 3 y final.

"""



Solo nos devolvería esto, ya que nada sigue el patrón.

* ^

Busca el comienzo de una linea

import re

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.

Esta es la linea 2.

Esta es la linea 3 y final.

"""

resultado = re.findall(r'^', texto)

print(resultado)



Pero es mas para usar en conjunto solo no tiene mucho sentido.

resultado = re.findall(r'^Hola', texto)



Y digamos queremos ver la segunda linea el inicio, usaremos otro parámetro.

resultado = re.findall(r'^Esta', texto,*flags*=re.**M**)

Hace que se interprete como si fuera una nueva linea. En pocas palabras activa la Multilinea.



* $

Busca el final de una linea.

import re

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.

Esta es la linea 2.

Esta es la linea 3 y final.

"""

resultado = re.findall(r'.$', texto,*flags*=re.**M**)

print(resultado)



* {n}

Busca n-cantidad de veces el valor de la izquierda

import re

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.

Esta es la linea 223.

Esta es la linea 3 y final.

"""

resultado = re.findall(r'\d{3}', texto)

print(resultado)



Entonces aquí lo que decimos es que busque “\d” que son dígitos del 0 al 9, pero que haya 3 seguidos.

resultado = re.findall(r'\d{3}\.\s', texto)

Y aquí pedimos que busque 3 numeros juntos, un punto y un espacio.



* {n,m}

Al menos n-cantidad de veces, y máximo m-cantidad de veces.

import re

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.

Esta es la linea 2 223.

Esta es la linea 3 y final.

"""

resultado = re.findall(r'\d{1,4}', texto)

print(resultado)

Esto nos esta buscando que como minimo haya un numero, y como máximo 4 juntos.



import re

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.

Esta es la linea 2 223.

Esta es la linea 3 y final.

abbb abbbbb a b

"""

resultado = re.findall(r'ab{1,4}', texto)

print(resultado)

Aquí lo que estamos haciendo es decir que busque los donde se encuentre una a y por lo minimo de 2 a 4 veces la b.



import re

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.

Esta es la linea 2 223.

Esta es la linea 3 y final.

ababab ababababab

"""

resultado = re.findall(r'(ab){1,4}', texto)

print(resultado)

Aquí encerramos ab en un grupo y estamos diciendo que busque los elementos en donde haya como minimo 2 a 4 veces una ab juntas



import re

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.

Esta es la linea 2 223.

Esta es la linea 3 y final.

abababab ababababab

"""

resultado = re.findall(r'(ab){2}', texto)

print(resultado)

Aquí lo que estamos diciendo es que si hay dos ab nos devuelva una sola ab.

* |

Busca una cosa o la otra

import re

texto = """Hola que tal estas el dia de hoy. Esta es la cadena 1.

Esta es la linea 2 223.

Esta es la linea 3 y final.

abababab ababababab

"""

resultado = re.findall(r'\d{2,4}|Esta', texto)

print(resultado)

Aquí lo que decimos es que si no encuentra un numero que tenga de 2 a 4 digitos mas juntos busque “Esta”.



En este caso devuelve ambas ya que encontró ambas.

import re

email = "ejemplo@ejemplo.com"

pattern = "[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]\.[a-zA-Z]{2,}"

resultado = re.match(pattern, email)

if resultado:

    print("Direccion de correo valida")

else:

    print("Direccion de correo invalida")

El guion medio sirve para hacer separaciones, entonces estamos diciendo, que pueden ser validos las comparaciones que tengan de la “a” a la “z”, “A” a la “Z”,del 0 al 9, todo lo que no sea un espacio en linea, permitimos los guiones bajos, el signo de porcentaje es como un comodín osea todo lo que encuentre antes o despues de el lo va a validar, y tenemos el mas que lo que hace es buscarnos 1 o mas coincidencias pero si no encuentra coincidencias no dejara pasar.

Despues tenemos otra mas fuera de los corchetes que nos dice que al menos una coincidencia con todo el primer patron.

Despues encuentra una arroba, y despues el mismo patron solo que sin el porcentaje, despues ponemos un punto que indica que solo estamos buscando el punto, y despues que me devuelva valores de a la “a” a la “z”, “A” a la “Z”, al menos dos veces. La coma despues del dos indica que no tiene limite.

Y con “match” lo que intentamos es buscar que coincida el patron en email, si coincide es un correo valido, si no es uno invalido.