Regex:-

It is used to search a word in a string.Seach a text in a really advanced way.

/at/ search 1st occurance of at only.

/at/g searches all occurances of at .

/e+/g match e or more than one e the result will be e , ee , eee etc.

/ea?/g matchs e , ea means e is necessary and a is optional

/re\*/ matches r , re,reee etc.

/.at/ matches cat , bat , .at etc.except new line

\. Matches . in the string backslash ignores just after the other except the special character such as (.)

/.\./ matches t. g. etc.

Originally the pattern not includes/fsfsf/ mean two forward slashes.

\w match any word character.

\s any form of white space

\S which is not a white space

\s and \S , \w and \W work as opposites respectively.

\w{4} any four char in a row.

\w{4,} four or more.

\w{4,5} four or five characters

[fc]at match any character which starts with f or c . It will give fat,cat etc.

[a-zA-Z]at matches aat, eat,Eat etc

[0-9]at means 0at 7at

[a-f]at aat ,fat, dat

(t|T) matches t or T | pipecharacter stands for or.

(t|T)he matches the , The

T|The will search only for t and The , it will not search the

(t|e|r){2,3} any two or three of(t,e,r,) example tre,et, eet,eee

(t|e|r){2,3}. Matches ter. , ret. , ert. eee.

() is used to have grouping.

re{2,3} gives ree,reee only

(re){2,3} gives rere,rerere only

(r|e){2,3} gives rr,ee,re,er,ere,rrr,eee etc

^T matches the beginning of a entire chuck , it will not work with separate lines.

^[Tt]he paragraph starting with the, The.

\.$ matches . at the end of whole paragraph

\ backslash is important and used in regex

\d{10} 10 digits in a row 1234567890

-? Means – is optional

\d{3}-?\d{3}-?\d{4} means 123-456-7890

(\d{3})[ -]?(\d{3})[ -]?(\d{4}) means 123 456 7890

$1$2$3 will give 1234567890

$ gives the group , if you don’t name the group

(?<areacode>\d{3})[ -]?(\d{3})[ -]?(\d{4})

‘areacode’ corresponds to 123

Optional paranthesis in (123)will be by

\)?(?<areacode>\d{3})\)?

\)? Will mention optional parenthesis

(\+1[ -])? if number begins with +1 , and here +1 is optional

import re

# by using compile function pattern not need to be mentioned everytime

# pattern = "\\\\@[w./]+\.[w]+@[w..]\/"

# text = "\\@vaibhav.sachdeva/@elililly.com@21../"

pattern = "\\\\[@][\w./]+@[\w.]+[\w@]+[\w.]+"

text = "\\@vaibhav.sachdeva/@elililly.com@21.."

x = re.findall(pattern, text)

print(x)