regex2.py

```
import re
str1="fry cry dry try cry Dry Cry a cry "

cry_s=re.search("cry" , str1)
print("start index" ,cry_s.start())
print("end index" ,cry_s.end())
print(cry_s.span())
print("match " ,cry_s.group())

#search
#findall
#finditer
# Finditer returns an iterator object with only one element

print(re.finditer("cry",str1))
print(list(re.finditer("cry",str1)))
for i in re.finditer("cry",str1):
print("span" ,i.span())
print("start" , i.start())
print("end" , i.end())
print("match" , i.group())
print("\n")

# . dot {} ()
# Metacharacters and special sequence characters
# [ ] : Match what is in the brackets
# [^ ] : Match anything not in the brackets
# ( ) : Return surrounded submatch
# . : Match any 1 character or space
# \b : Word boundary
# ^ : Beginning of String
# $ : End of String
# \n : Newline
# \d : Any 1 number
# \D : Anything except a number
# \w : Same as [a-zA-Z0-9_]
# \W : Same as [^a-zA-Z0-9_]
# \s : Returns a match where the string contains a white space character
# \S : Returns a match where the string DOES NOT contain a white space character
# {5} : Match 5 of what preceeds the curly brackets
# {5,7} : Match values that are between 5 and 7 in length

str2="regex is a smart way 090@."
print(re.findall(".",str2))

str1="fry cry dry try cry 1ry #ry ry Cry a cry "
print(re.findall(".ry" , str1))
```

```
str3="pin bin kin tin"
print(re.findall(".in" , str3))

str4='''regex is a smart way to search and organize
text easily and efficiently'''
print(re.findall("\s(\w{3})\s",str4))
nums="123 455 89992 9887776662 7663332272 77777777778 "
print(re.findall("\s(\d{10})\s",nums))

str5="Mr. C.B.I Dr. .1."

print(re.findall("\w\.",str5))
print(re.findall("\..\.",str5))

str4='''regex is a smart way to search and organize
text easily and efficiently'''


print(re.findall(r"\s(\w{2,6})\s",str4))

num1="0755-8989-223456"

print(re.findall("\d{4}-\d{4}-\d{6}",num1))

str1="My name is ibrahim"
emails='''miibm232005@gmail.com sww2@gmail.com
qwert.@wegmail.com'''

print(re.findall("[\w.@*&+-]{1,20}@\w{1,8}\.\w{1,5}", emails))
```

```
####-------------REGULAR EXPRESSIONS---------------##

'''
Regex, or regular expressions, is like a search tool
for text. It helps find specific patterns or
characters in words, sentences, or data.

It’s widely used for:

Data Validation: Ensuring inputs like emails or phone
numbers are correctly formatted.

Text Search & Extraction: Pulling specific info
(e.g., URLs, emails) from text or data files.

Data Cleaning: Removing unwanted characters or
standardizing formats in data.

Web Scraping: Extracting details (prices, product names)
from websites.

Editing Text: Searching and replacing specific
words or patterns across files.

regex is a smart way to search and organize
text easily and efficiently.

'''
import re
#search return the first occurance as a match object
str1="fry cry dry try cry Dry Cry a "
##print(re.search("cry" , str1))

#findall returns list of matches
##print(re.findall("cry",str1))

#[] accepts only one character
##print(re.findall("[aA-zZ]ry",str1))

# Use ^ to denote any character except
##whatever characters are between the brackets
str1="fry cry dry try cry Dry Cry a "
##print(re.findall("[^cC]ry",str1))
##print(re.findall("[^aA-gG]ry",str1))

#Replace/edit
#sub replaces one or many matches with a string
str1="fry cry dry try cry Dry Cry f "
print(re.sub("f","k", str1))
print(re.sub("ry","at", str1))
```

```
##compile method creates a regex pattern object
pat1=re.compile("cry")
##print(re.search(pat1,str1))



##The backslash (\) in regex is used to define
##special characters or escape characters that
##have a special meaning in regex.

str2="ABC Tr 23 Y6 8u #@&*()+-_"

##\d any digits (numbers from 0-9)
print(re.search("\d",str2))
print(re.findall("\d",str2))

##[^\d]=\D
##\D any string which DOES NOT contain digits
print(re.findall("\D",str2))

##\s any string which contains a white space
## character
print(re.search("\s",str2))
print(re.findall("\s",str2))

##[^\s]=\S
##\S any string which DOES NOT contain a white
## space character
print(re.findall("\S",str2))

##\w any string contains word characters
## (characters from a to Z, digits from 0-9,
## and the underscore _ character)
##w=[a-z 0-9 _]
print(re.findall("\w",str2))

##\W any string which DOES NOT contain any word characters

print(re.findall("\W",str2))
```