1. What is the name of the feature responsible for generating Regex objects?

Answer: The function re.compile() is responsible for generating Regex objects.

2. Why do raw strings often appear in Regex objects?

Answer: Raw strings are used so that backslashes donot have to be escaped.

3. What is the return value of the search() method?

Answer: Search() method returns matched objects.

4. From a Match item, how do you get the actual strings that match the pattern?

Answer : We perform the following steps to get actual strings that match the pattern:

1. import regex module as re
2. create regex object with re.compile()
3. pass the string you want to search into search() function of the object.This returns a match object.
4. Call the object’s group method to return a string of the actual matched text.

5. In the regex which created from the r'(\d\d\d)-(\d\d\d-\d\d\d\d)', what does group zero cover? Group 2? Group 1?

Answer: Group 0 returns entire string matched according to the pattern. Group 1 returns first expression matched between 1 pair of parenthesis . Group 2 returns expression matched between 2nd pair of parenthesis.

6. In standard expression syntax, parentheses and intervals have distinct meanings. How can you tell a regex that you want it to fit real parentheses and periods?

Answer: We do the same by mentioning in the expression inside re.compile() method which consists of different expressions inside different pairs of parenthesis.

7. The findall() method returns a string list or a list of string tuples. What causes it to return one of the two options?

Answer: findall() method returns a string list and returns every occurance of the expression that we got in form of list list .

8. In standard expressions, what does the | character mean?

Answer: | means either or condition which return True/False depending whether the condition is satisfied or not for 1 or multiple arguments send inside findall() method.

9. In regular expressions, what does the character stand for?

Answer: Character simply are a set of values which you want to match upto and get the result defined accordingly inside findall() method.

10.In regular expressions, what is the difference between the + and \* characters?

Answer: + considers 1 or more occurances. \* considers 0 or more occurances.

11. What is the difference between {4} and {4,5} in regular expression?

Answer: {4} means it wants to find occurance of a character 4 times exactly . For the expression {4,5}, we want to find occurance of a character 4 or 5 times exactly which is defined inside re.compile() method.

12. What do you mean by the \d, \w, and \s shorthand character classes signify in regular expressions?

Answer: \d matches a digit. \w represents character . \s is used to find a whitespace character.

13. What do means by \D, \W, and \S shorthand character classes signify in regular expressions?

Answer: \D matches to any non-decimal digit , \W matches any alpha numeric character and \S matches upto a string containing non spaces.

14. What is the difference between .\* and .\*?

Answer: .\* is a greedy mode which tried to get the longest string satisfying the condition and .\*? is a non greedy mode which tried to get the minimal string satisfying the string at the moment.

15. What is the syntax for matching both numbers and lowercase letters with a character class?

Answer: Our pattern = \d\*\w\*\S\*

16. What is the procedure for making a normal expression in regax case insensitive?

Answer: use re.IGNORECASE parameter inside every method to be used .

17. What does the . character normally match? What does it match if re.DOTALL is passed as 2nd argument in re.compile()?

Answer: . character matches with any character in the current line. Using re.DOTALL helps in expanding this functionality to new line as well.

18. If numReg = re.compile(r'\d+'), what will numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4 hen') return?

Answer: We will get X drummers, X pipers, five rings, X hen.

19. What does passing re.VERBOSE as the 2nd argument to re.compile() allow to do?

Answer: re.VERBOSE allows to write regular expressions that look nicer by making it more readable and separates it logically by adding comments as well for them.

20. How would you write a regex that match a number with comma for every three digits? It must match the given following:

'42' --

'1,234'

'6,368,745'

but not the following:

'12,34,567' (which has only two digits between the commas)

'1234' (which lacks commas)

Answer : Maybe this expression will work (^\d+(,\d{3})\*$)

21. How would you write a regex that matches the full name of someone whose last name is Watanabe? You can assume that the first name that comes before it will always be one word that begins with a capital letter. The regex must match the following:

'Haruto Watanabe'

'Alice Watanabe'

'RoboCop Watanabe'

but not the following:

'haruto Watanabe' (where the first name is not capitalized)

'Mr. Watanabe' (where the preceding word has a nonletter character)

'Watanabe' (which has no first name)

'Haruto watanabe' (where Watanabe is not capitalized)

Answer:

import re

namesearch = re.compile(r’[A-Z][a-z]+\s’Watanabe’,re.I|re.VERBOSE)

result = namesearch.search(“Haruto Watanebe”)

print(result)

22. How would you write a regex that matches a sentence where the first word is either Alice, Bob, or Carol; the second word is either eats, pets, or throws; the third word is apples, cats, or baseballs; and the sentence ends with a period? This regex should be case-insensitive. It must match the following:

'Alice eats apples.'

'Bob pets cats.'

'Carol throws baseballs.'

'Alice throws Apples.'

'BOB EATS CATS.'

but not the following:

'RoboCop eats apples.'

'ALICE THROWS FOOTBALLS.'

'Carol eats 7 cats.'

Answer:

Import re

Reg = re.compile(r’(Alice|Bob|Carol)\s( eats|pets|throws)\s(apples|cats|baseball)\.’,re.IGNORECASE)

Print(reg.search(str))

# str is our input text here.