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

Answer: Compile() method in ‘re’ module package is responsible for generating Regex objects. The patterns we provide to search using regex methods such as re.match() or re.search() will generate the objects.

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

Answer: Raw strings are used to have proper meaning of the search pattern and that backslashes do not escape or need another backslash to support.

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

Answer: Search() method will return the match object and only first match is returned. In case of no match None is returned.

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

Answer: You can use .group() after the search to get the part of the string where there was a match.

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 is the entire match. First set of parenthesis is covered in group 1. Second set of parenthesis is covered in group 2.

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: Using a backslash we can tell regex that we want real parentheses and period: \. \( \)

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: If the search object has no groups, a list of strings is returned. If the regex has groups, a list of tuples of strings is returned.

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

Answer: | means ‘or’

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

Answer: The character stands for a specific search pattern in text. It is predefined in the class to search as per the pattern set for each character.

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

Answer: \* mean 0 or more occurrence and + means one or more occurrence

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

Answer: {4} means exactly 4 occurrences between the required pattern

{4,5} means exactly 4 or 5 occurrences between the required pattern

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

Answer: \d means the search will return a match if it contains digits (0 to 9)

\w means the search will return a match if the string contains any word characters

\s means the search will return a match if the string contains whitespace.

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

Answer: \D means the search will return a match if string does not contains digits (0 to 9)

\W means the search will return a match if the string does not contains any word characters

\S means the search will return a match if the string does not contains whitespace.

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

Answer: .\* means the maximum characters in between start and end search parameter. It starts the end character search from the end to have a match. Only one match will be provided will all the max characters inside.

.\*? means it first searches the first character and then searches the end character after the first one and if match is found then repeats the same process. More than one match will be provided if present.

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

Answer: [0-9a-z]

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

Answer: Ignorecase is an optional parameter can be used for case insensitive. [re.IGNORECASE]

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

Answer: the . character matchs any character except the newline character. If re.DOTALL is passed as 2nd argument in re.compile() then it will also match newline characters.

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

Answer: X drummers, X pipers, five rings, X hen [P.N. syntax should be numRege.sub()]

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

Answer: VERBOSE allows to add the whitespace and comments to the string passed to re search.

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: import re

x = re.compile('^\d{1,3}(,\d{3})\*$')

print (re.search(x, '6,368,745'))

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

String='Haruto Watanabe'

x = re.search("^[A-Z].\*[^.]\sWatanabe$", string)

print(x)

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

text='Alice eats apples.'

x=re.compile(r'(?:Alice|Bob|Carol)\s(?:eats|pets|throws)\s(?:apples|cats|baseballs)\.',re.IGNORECASE)

y=re.search(x, text)

print(y)