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

Regex objects are created by the function **re.compile()**.

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

Raw strings are used to avoid having to escape backslashes.

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

The search() method returns items that have been matched or None if no match found.

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

* + import the regex module as  re
  + Using re.compile(), build a regex object
  + Enter the string to be searched into the object's search() function. This returns a match entity.
  + To get a string containing the actual matched text, use the object's group method.

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?

The entire match is covered by Group 0, the first set of parentheses is covered by Group 1, and the second set of parentheses is covered by 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?

We do the same thing by stating it in the expression within the re.compile() method, which is made up of various expressions enclosed by 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?

The findall() method returns a string list that contains any occurrence of the expression that we obtained in the form of a list list.

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

| denotes an either or condition that returns True or False depending on whether the condition is satisfied or not for one or more arguments within the findall() method.

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

Characters are simply a collection of values that you want to match up to and get the result specified within the findall() process.

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

* + - **considers 1 or more occurrences. \* considers 0 or more occurrences.**

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

The number 4 {4} indicates that it wishes to locate the occurrence of a character four times in a row. For the expression {4,5}, we want to find the exact occurrence of a character specified within the re.compile() method 4 or 5 times.

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

\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?

\D matches any non-decimal digit, \W matches any alpha numeric character, and \S matches any string that does not contain spaces.

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

.\* is a greedy mode that seeks the longest string satisfying the condition, whereas.\*? is a non greedy mode that seeks the shortest string satisfying the string at the time.

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

**\d\*\w\*\S\***

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

Use the **re.IGNORECASE** parameter within each method that will be used.

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

. character matches with every character in the current line.  The use of re.DOTALL aids in the expansion of this functionality to new lines as well.

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

**X drummers, X pipers, five rings, X hen**

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

re.VERBOSE helps you to write regular expressions that are more readable and logically separated by adding annotations to 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)

**(^\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)

#############################

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.'

#############################

import re

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

Print(reg.search(str))

#############################

Note: str is our input text here.