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

Ans 1 - re.compile() function returns regex objects.

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

Ans 2 - Raw strings are used so that backslashes do not have to be escaped.

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

Ans 3 - Python re.search() is an inbuilt regex function that searches the string for a match and returns the match object if there is a match. The re.search() function takes two parameters and returns a match object if there is a match

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

Ans 4 - The group() method returns strings of the 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?

Ans 5 - Group 0 is the entire match, group 1 covers the first set of paranthesis, and group 2 covers the second set of paranthesis

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?

Ans 6 - Period and Paranthesis can be escaped with a backslash : \., \(, and \).

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

Ans 7 - If the regex 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?

Ans 8 - REs separated by '|' are tried from left to right. When one pattern completely matches, that branch is accepted. This means that once A matches, B will not be tested further, even if it would produce a longer overall match

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

Ans 9 - The Python module re provides full support for Perl-like regular expressions in Python. A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern.

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

Ans 10 - Each of them are quantifiers, the star quantifier(\*) means that the preceding expression can match zero or more times it is like {0,} while the plus quantifier(+) indicate that the preceding expression MUST match at least one time or multiple times and it is the same as {1,} .

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

Ans 11 - {4} will match exactly four 'a' characters, but not three.

{4,5} will match 4 to 5 'a' characters. Omitting m specifies a lower bound of zero, and omitting n specifies an infinite upper bound

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

Ans 12 - \d is short for [0-9].

\w stands for “word character”. It always matches the ASCII characters [A-Za-z0-9\_].

\s stands for “whitespace character”. Again, which characters this actually includes, depends on the regex flavor. In all flavors discussed in this tutorial, it includes [ \t\r\n\f].

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

Ans 13 - In this case, [\d\w\s] matches any digit, word, or whitespace character

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

Ans 14 - (Dot.) - In the default mode, this matches any character except a newline.

(?) - Causes the resulting RE to match 0 or 1 repetitions of the preceding RE. ab? will match either ‘a’ or ‘ab’.

(\*) - Causes the resulting RE to match 0 or more repetitions of the preceding RE, as many repetitions as are possible. ab\* will match ‘a’, ‘ab’, or ‘a’ followed by any number of ‘b’s.

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

Ans 15 - 1) Import the regex module with import re.

2) Create a Regex object with the re.compile() function. (Remember to use a raw string.)

3) Pass the string you want to search into the Regex object’s search() method. This returns a Match object.

4) Call the Match object’s group() method to return a string of the actual matched text.

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

Ans 16 - re.IGNORECASE : This flag allows for case-insensitive matching of the Regular Expression with the given string

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

Ans 17 - The . character normally matches any character except the newline character. If re.DOTALL is passed as the second argument to re.compile(), then the dot 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?

Ans 18 - 'X drummers, X pipers, five rings, X hens'

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

Ans 19 - The re.VERBOSE argument allows you to add whitespace and comments to the string passed to re.compile().

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)

Ans 20 - re.compile(r'^\d{1,3}(,\d{3})\*$') will create this regex, but other regex strings can produce a similar regular expression.

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)

Ans 21 - re.compile(r'[A-Z][a-z]\*\sWatanabe')

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.'

Ans 22 - re.compile(r'(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.', re.IGNORECASE).