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

**Ans** : re.compile() function is used for generating regex objects.

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

**Ans :** Patterns are usually expressed in Python using raw strings because many a times we want to include special characters in patterns like \ , + , \* , ? , \n , etc. These characters have some special meaning in python. If we use them directly, it collides with python’s usage of same character. So to avoid this raw string notation is used. Eg. r“\n” ; here \n is not considered as new line, instead its considered as string of \ & n.

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

**Ans** : search () method gives indices of the matching string (starting & ending index values) & the matching string as well.

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

**Ans** : We need to use .group() function.

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** : The group zero covers whole pattern which will match total 10 digits. Group 1 i.e. first set of parentheses matches 3 digits, group 2 covers second set of parentheses which covers 7 digits in set of 3 & 4.

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** : For that purpose we need to use escape sequence. Its denoted by mentioning that special characters prefixed with \ . Eg. “\(“

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 :** It depends on the number of capturing groups in pattern. If there are no groups, return a list of strings matching the whole pattern. If there is exactly one group, return a list of strings matching that group. If multiple groups are present, return a list of tuples of strings matching the groups.

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

**Ans :** The character | means OR in standard expressions used for Boolean operations.

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

**Ans** : In regular expressions | creates a regular expression that will match either *A* or *B* for expression A|B .

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

**Ans** : + causes the resulting RE to match 1 or more repetitions of the preceding RE. Whereas \* causes the resulting RE to match 0 or more repetitions of the preceding RE

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

**Ans** : {4} specifies exactly 4 copies of the RE should be matched. {4,5} causes the RE to match 4 to 6 repetitions of the RE.

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

**Ans :** \d is used to match any digit in the range [0-9]

\w is used to match any single letter, number or underscore [a-zA-Z0-9\_]

\s stands for whitespace character. It is used to match a space, a tab \t, a new line \n etc.

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

**Ans :** \D is opposite of \d ; its used to match any single character that is not a digit.

\W is opposite of \w ; its used to match any single character which is not a word character.

\S is opposite of \s ; its used to match any character which is not a whitespace character.

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

**Ans :** .\* matches entire string whereas .\*? matches as few characters as possible.

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

**Ans** : [a-z0-9]

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

**Ans :** We need to use re.IGNORECASE flag for that purpose.

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

**Ans :**  . character matches any character except a new line. If re.DOTALL is passed as flag it will match any character including new line.

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

**Ans :** 'X drummers, X pipers, five rings, X hen'

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

**Ans :** re.VERBOSE allows us to write regexs that look nicer & are better readable by allowing us to add comments in pattern using character #. Anything starting from # is ignored till end. So comment can be added by using prefix #.

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** : import re  
 regex = re.compile(r"\d{1,3}(,\d{3})\*$")  
 result = regex.match('6,368,745')  
 print(result)

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 :**

import re  
regex = re.compile(r"[A-Z]\w+\sWatanabe")  
result = regex.match('Haruto Watanabe')  
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.'

**Ans :**

import re

string = input("Enter your string : ")

regex = re.compile(r"(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.?$",re.IGNORECASE)

result = regex.match(string)

try:

if string == result.group() :

print(" yes its valid", result)

except Exception as e :

print("its invalid string")