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

Ans : The name of the feature responsible for generating Regex objects is the re module in Python.

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

Ans : Raw strings (prefixed with r) are often used in Regex objects to avoid unwanted escaping of characters. Since regular expressions often involve backslashes, which are also escape characters in Python strings, using a raw string ensures that the backslashes in the regular expression are interpreted as literal backslashes.

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

Ans : The return value of the **search()** method is a **Match** object if a match is found, or **None** if no match is found.

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

Ans : To get the actual strings that match the pattern from a **Match** object, you can use the **group()** method with an argument of **0** or no argument. **group(0)** represents the entire matched string.

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 : In the regex created from **‘r’(\d\d\d)-(\d\d\d-\d\d\d\d)'**, group zero (i.e., **group(0)**) covers the entire matched string. Group 1 (i.e., **group(1)**) covers the first three digits, and Group 2 (i.e., **group(2))** covers the second set of three digits separated by a hyphen.

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 : To tell a regex that you want to match real parentheses **( ),** as well as periods **‘.’** , you need to escape them using 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 : The **findall()** method returns a list of strings when the regex pattern has no capturing groups, and it returns a list of tuples when the regex pattern has capturing groups. Each tuple contains the matched groups.

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

Ans : In standard expressions, the **|** character means **"OR,"** allowing you to specify alternatives for the pattern. It matches either the expression before it or the one after it.

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

Ans : In regular expressions, the . character (dot) is a special metacharacter that matches any character except for a newline.

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

Ans: In regular expressions, the + character means "one or more occurrences" of the preceding character or group. The \* character means "zero or more occurrences."

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

Ans : **{4}** in a regular expression matches exactly four occurrences of the preceding character or group. **{4,5}** matches between four to five occurrences of the preceding character or group.

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

Ans : In regular expressions, the shorthand character class \d signifies any digit (0-9). Similarly, \w represents any alphanumeric character (a-z, A-Z, 0-9, and underscore \_). \s stands for any whitespace character (spaces, tabs, newlines).

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

Ans : The uppercase versions of shorthand character classes signify the negation of their lowercase counterparts. So, \D represents any non-digit, \W represents any non-alphanumeric character, and \S represents any non-whitespace character.

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

Ans : The .\*? and .\* are both used for matching patterns, but they behave differently. .\*? is a non-greedy or minimal match, meaning it will match as few characters as possible, while .\* is a greedy match, meaning it will match as many characters as possible.

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

Ans : To match both numbers and lowercase letters, you can use the character class [0-9a-z] or use the shorthand character class \w (word character) which includes lowercase letters and digits.

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

Ans : To make a regular expression case-insensitive, you can pass the re.IGNORECASE flag (or re.I as a shorthand) as the second argument to re.compile() or when using other regex functions.

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

Ans : The . character in a regular expression normally matches any character except for a newline. If re.DOTALL is passed as the second argument to re.compile(), then the . character will match any character, including a newline.

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

Ans : numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4 hen') will 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?

Ans : Passing re.VERBOSE as the second argument to re.compile() allows you to write regular expressions in a more readable and well-structured format. It ignores whitespace and allows you to add comments within the pattern.

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

pattern = r'^\d{1,3}(,\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)

Ans : import re

pattern = r'^[A-Z][a-zA-Z]\* Watanabe$'

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

pattern = r'^(Alice|Bob|Carol) (eats|pets|throws) (apples|cats|baseballs)\.$'

regex = re.compile(pattern, re.IGNORECASE)