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

The feature responsible for generating Regex objects in Python is the “re” module.

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

Raw string are commonly used for regular expressions to avoid backslash escaping issues.

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

The search() method is used to search for a pattern within a string. It scans the string from left to right and returns the first occurrence of the pattern. If the pattern is found a match object is returned, which provides information about the matched substring including its position in the original string.

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

The search() method is used to find the first occurrence of the pattern within the string. The group() method is then called on the Match object to obtain the matched substring.

**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?**

Group 0: This represents the entire match

Group 1: This refers to the first set of parentheses (\d\d\d).

Group 2: This refers to the second set of parentheses (\d\d\d-\d\d\d\d)

**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?**

The parentheses and period can be escaped using a backslash (\) which indicates that they should be treated as literal characters and not as special metacharacters.

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

If the regular expression pattern contains no capturing groups (i.e. no parentheses) findall() returns a list of strings. Each string in the list represents a complete match of the pattern in the input string.

If the regular expression pattern contains one or more capturing groups findall() returns a list of string tuples. Each tuple in the list corresponds to a complete match and each element of the tuple represents the captured content from each capturing group.

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

In standard regular expressions, the | character known as the pipe or alternation operator is used to specify alternative patterns.

pattern1|pattern2 matches either pattern1 or pattern2. It tries to match pattern1 first and if it fails it tries to match pattern2.

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

**-------**

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

The + matches one or more. The \* matches zero or more.

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

The {4} matches exactly three instances of the preceding pattern. The {4,5} specifies that the preceding pattern must occur between 4 and 5 times.

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

\d shorthand character class represents any digit character.

\w shorthand character class represents any word character.

\s shorthand character class represents any whitespace character.

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

\D shorthand character class represents any character that is not a digit.

\W shorthand character class represents any character that is not a word character.

\S shorthand character class represents any character that is not a whitespace character.

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

The .\* performs a greedy match, and the .\*? performs a nongreedy match.

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

[0-9]: Matches any digit character from 0 to 9.

a-z: Matches any lowercase letter from a to z.

When combined within the same character class [0-9a-z], it matches any character that is either a digit or a lowercase letter.

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

To make a regular expression case-insensitive in Python, you can use the re.IGNORECASE flag or the re.I flag. These flags instruct the regular expression engine to perform a case-insensitive matching.

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

In regular expressions, the . (dot) character normally matches any character except a newline character (\n). However, if the re.DOTALL flag is passed as the second argument to re.compile(), the . character will match any character including newline characters.

**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?**

Passing re.VERBOSE as the second argument to re.compile() allows you to create regular expressions that are more readable and maintainable by adding whitespace and comments to 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)**

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)**

re.compile(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.'**

re.compile(r'^(Alice|Bob|Carol) (eats|pets|throws) (apples|cats|baseballs)\.$', re.IGNORECASE)