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

2. Why do raw strings often appear in Regex objects? **Regular expressions use special characters (such as \ or ^) that have special meaning in Python strings, and using a raw string prevents these characters from being interpreted as special characters**

3. What is the return value of the search() method? **Returns a match object if there is a match anywhere in the string. If there is no match, it returns None**

4. From a Match item, how do you get the actual strings that match the pattern? **Using the group() method we can get the actual string that match the pattern**

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 refers to the entire match \d\d\d-\d\d\d-\d\d\d\d**

**group 1 refers to the first set of capturing parentheses \d\d\d**

**group 2 refers to the second set of capturing parentheses, i.e., the next three digits followed by a hyphen \d\d\d- and the last four digits \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? **To match them literally in the text, you need to escape them with a backslash \**

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 pattern contains no capturing groups (i.e., no sets of parentheses), findall() returns a list of strings, where each element in the list is a substring of the input string that matches the pattern.**

**If the pattern contains one or more capturing groups, findall() returns a list of tuples, where each tuple contains the strings that match the individual capturing groups. In this case, each tuple represents a separate match of the pattern in the input string.**

8. In standard expressions, what does the | character mean? **| (pipe) character is used to specify alternation, meaning that a match can occur either before or after the pipe.**

9. In regular expressions, what does the character stand for? **Which character? This question is incomplete**

10.In regular expressions, what is the difference between the + and \* characters? **In regular expressions, + matches one or more occurrences of the preceding character or group, while \* matches zero or more occurrences of the preceding character or group. The difference is that + requires at least one match, while \* allows for zero matches.**

11. What is the difference between {4} and {4,5} in regular expression? **Curly braces {} with a number specify a precise number of occurrences of the preceding character or group.**

**{4} specifies exactly 4 occurrences of the preceding character or group**

**{4,5} specifies between 4 and 5 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? **\d represents a digit character (0-9).**

**\w represents a word character, which includes letters (A-Z, a-z), digits (0-9), and underscore (\_).**

**\s represents a white-space character, which includes space, tab, newline, and other characters that cause whitespace.**

13. What do means by \D, \W, and \S shorthand character classes signify in regular expressions? **\D represents any character that is not a digit character (0-9).**

**\W represents any character that is not a word character, which includes letters (A-Z, a-z), digits (0-9), and underscore (\_).**

**\S represents any character that is not a white-space character, which includes space, tab, newline, and other characters that cause whitespace.**

14. What is the difference between .\*? and .\*? **In regex, .\*? is a non-greedy version of .\* and matches as little characters as possible, while .\* is a greedy version and matches as many characters as possible.**

**For example, in the string "1234567890", the pattern .\*?5 would match "1234", while the pattern .\*5 would match "1234567890".**

15. What is the syntax for matching both numbers and lowercase letters with a character class? **This can be done using the [0-9a-z] syntax**

16. What is the procedure for making a normal expression in regax case insensitive? **In regular expressions, you can make a pattern case-insensitive by using the re.IGNORECASE flag or the (?i) inline flag.**

17. What does the . character normally match? What does it match if re.DOTALL is passed as 2nd argument in re.compile()? **. character normally matches any character except a newline character (\n).**

**if the re.DOTALL flag is passed as the second argument in the re.compile() function, the . character will match any character, including the newline character.**

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? **Allows to include extra whitespace and comments in your regular expression pattern. This makes your pattern more readable and easier to understand, especially for complex or multi-line patterns.**

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)

**import re**

**pattern = re.compile(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)

**import re**

**pattern = 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.'

**import re**

**sentence\_regex = re.compile(r'(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.', re.IGNORECASE)**