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

re.compile() function in Python's ‘re’ module.

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

Raw strings (r'...') are often used in Regex objects to prevent Python's string literal interpretation from modifying backslashes. This is useful because backslashes are commonly used in regular expressions, and using raw strings ensures that they are interpreted as literal backslashes.

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

The return value of the search() method is a Match object if the pattern is found in the string, or None if no match is found.

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

From a Match object, you can get the actual strings that match the pattern using the group() method with argument 0 for the entire match.

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?

In the regex r'(\d\d\d)-(\d\d\d-\d\d\d\d)', group 0 covers the entire matched string, group 1 covers the first three digits, and group 2 covers the remaining digits.

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 fit real parentheses and periods in a regex, you need to escape them with a backslash I.e. \(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?

The findall() method returns a list of strings when there are no capturing groups in the pattern. If there are one or more capturing groups, it returns a list of tuples containing the strings matched by each group.

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

‘or’

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

In regular expressions as well, the character | stands for "or". It is used to specify alternatives within a regular expression pattern. For example, the pattern cat|dog would match either "cat" or "dog".

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

In regular expressions, the + character matches one or more occurrences of the preceding element, while the * character matches zero or more occurrences.

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

{4} in regular expression matches exactly four occurrences of the preceding element, while {4,5} matches between four and five occurrences.

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

In regular expressions, \d matches any digit (equivalent to [0-9]), \w matches any alphanumeric character (equivalent to [a-zA-Z0-9_]), and \s matches any white space character.

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

In regular expressions, \D matches any non-digit character, \W matches any non-alphanumeric character, and \S matches any non-whitespace character.

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

.*? is a non-greedy match that matches as few characters as possible, while .* is a greedy match that matches as many characters as possible.

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

[0-9a-z]

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

Pass the re.IGNORECASE or re.I flag as the second argument to re.compile()

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

The . character normally matches any character except newline. If re.DOTALL is passed as the second argument in re.compile(), the . character will match any character, including newline.

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?

It allows to add whitespace and comments to the regular expression pattern to make it more readable.

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)

A regex that matches a number with comma for every three digits is 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)

A regex that matches the full name of someone whose last name is Watanabe is

r'^[A-Z][a-zA-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.'

A regex that matches a sentence as described is
r'^(Alice|Bob|Carol)\s+(eats|pets|throws)\s+(apples|cats|baseballs)\.$' with re.IGNORECASE flag.