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

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

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

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

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?

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

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

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

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

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

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

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

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

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

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

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

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

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

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)

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)

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:**1. The feature responsible for generating Regex objects is the `re.compile()` function in Python's `re` module.

2. Raw strings (prefixed with 'r') are often used in Regex objects because they treat backslashes (\) as literal characters. This is useful in regular expressions where backslashes are frequently used to escape special characters.

3. The return value of the `search()` method is a Match object if a match is found, otherwise it returns `None`.

4. From a Match object, you can get the actual strings that match the pattern using the `group()` method. Group 0 represents the entire match, while subsequent groups represent portions of the match captured by parentheses in the regular expression pattern.

5. In the regex `r'(\d\d\d)-(\d\d\d-\d\d\d\d)'`, group 0 covers the entire match, group 1 covers the first set of three digits (area code), and group 2 covers the second set of three digits followed by a hyphen and four digits (phone number).

6. To specify real parentheses and periods in a regex pattern, you need to escape them with a backslash (\). For example, to match a literal period, use `\.`.

7. The `findall()` method returns a list of string tuples when the regex pattern contains capturing groups, and a list of strings otherwise. If the pattern contains groups, each tuple in the returned list represents a match, with each element of the tuple corresponding to a capturing group.

8. In standard expressions, the | character means "or", allowing you to match either one expression or another.

9. In regular expressions, the dot character (.) matches any single character except a newline character.

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

11. {4} in a regular expression specifies exactly four occurrences of the preceding element, while {4,5} specifies between four and five occurrences of the preceding element.

12. In regular expressions, the shorthand character classes \d, \w, and \s represent:

- \d: Matches any digit (equivalent to [0-9]).

- \w: Matches any alphanumeric character (equivalent to [a-zA-Z0-9\_]).

- \s: Matches any whitespace character (space, tab, newline).

13. In regular expressions, the shorthand character classes \D, \W, and \S represent:

- \D: Matches any non-digit character (equivalent to [^0-9]).

- \W: Matches any non-alphanumeric character (equivalent to [^a-zA-Z0-9\_]).

- \S: Matches any non-whitespace character.

14. .\*? matches any character (except for newline) zero or more times, but as few times as possible. .\* matches any character (except for newline) zero or more times, but as many times as possible.

15. The syntax for matching both numbers and lowercase letters with a character class is [0-9a-z] or [a-z0-9].

16. To make a normal expression case insensitive in regex, you can pass the re.IGNORECASE or re.I flag as the second argument to the re.compile() function.

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

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

19. Passing re.VERBOSE as the 2nd argument to re.compile() allows for the use of whitespace and comments within the regular expression pattern for better readability.

20. A regex that matches a number with commas for every three digits is r'^\d{1,3}(,\d{3})\*$'.

21. A regex that matches the full name of someone whose last name is Watanabe is r'[A-Z][a-z]\*\sWatanabe'.

22. A regex that matches a sentence with specific words and ends with a period is r'^(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.$'.