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

Ans:- The name of the feature responsible for generating Regex objects was re.compile() method.

---

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

Ans:- As back slashes do not have to be escaped, So Raw strings often appear in Regex objects.

---

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

Ans:- search() method returns the index(position) of the first match and also returns -1 if no match was found.

---

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

Ans:- The actual strings that match the pattern from a match item can get by the group() method.

---

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 the r'(\d\d\d)-(\d\d\d-\d\d\d\d)', Group-0 covers the entire match, Group-1 covers the first set of parentheses and Group-2 covers the second set of parentheses.

---

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:- The regex can tell by escaping with 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:- If the regex has no groups, a list of strings is returned. If the regex has groups, a list of tuples of strings is returned.

---

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

Ans:-The | character signifies matching “either, or” between two groups.

---

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

Ans:- Each character in a regular expression is either a metacharacter, having a special meaning or a regular character that has a literal meaning.

---

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

Ans:- The + matches one or more and * matches Zero or more.

---

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

Ans:- In regular expression {4} exactly matches four instances of the preceding group and {4,5}

matches between four and five instances.

---

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

Ans:- In regular expression the shorthand characters \d, \w, and \s matches to a single digit,

Word or space character, respectively.

---

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

Ans:- In regular expression the shorthand characters \D,\W and \S matchs to a single character

that is not a digit, word, or space character, respectively.

---

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

Ans:- (.*?) matches any character (.) any number of times (*), as few times as possible to make the

Regex match(?) and (.*)? Captures a group zero or one times(?).

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

Ans:- The syntax for matching both numbers and lowercase letters with a character class is Either[0-9,a-z] or [a-z,0-9].

---

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

Ans:- The procedure for making a normal expression in regax case insensitive was by using CASE_INSENSITIVE flag.

---

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

Ans:- . character normally matches any character except the newline character. If re.DOTALL is

Passed as the second argument to re.complie(), then the dot will also match newline characters.

---

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

hen') return?

Ans:- X drummers, X pipers, five rings, X hens.

---

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

Ans:- By passing re.VERBOSE as the 2nd argument to re.compile() Ignores Whitespace and comments inside the regular expression string.

---

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:- re.complier(r'^\d{1,3}(,\d{3})*$') will create this regex, but other regex strings can produce a similar regular expression.

---

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:- re.complie(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.'

Ans:-
re.complier(r'(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.',re.IGNORECASE)