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

Ans: re.compile()

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

Ans: in order to avoid escaping backslashes.

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

Ans: re.search returns the matched objects if exists otherwise None.

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

Ans: We use group() method to get the actual strings that match a 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?

Ans: Group 0 is the entire matched expression, Group 1 is the match for expression within the first set of parenthesis and Group 2 coverts the match for expression within the second set of parenthesis.

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: As parenthesis and intervals are regex metacharacters hence we need to escape with a backslash if we want to fit a real parenthesis or period.

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 there is no group in the regex, a list of strings is returned otherwise a list of tuples of strings is returned.

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

Ans: “|” means “Or

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

Ans: Either match a zero or one character

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

Ans: “+” matches one or more characters while “\*” matches zero or more characters

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

Ans: {4} means match exactly four characters while {4,5} means match either 4 or 5 characters

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

Ans: \d are for digits, \w are for word characters a-zA-Z0-9 and \s is for spaces

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

Ans: \D for all characters which are not digits, \W is for all characters except the word characters and \S is for all characters except the space character.

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

Ans: “.\*” Matches all the characters between the enclosed characters as shown below:

Say the string: S = “AxyzBccg5tApqrB” then A.\*B would return the entire string S however, A.\*?B would match the two smaller chunks as AxyzB and ApqrB. That is first one is a greedy expression while the later on is the reluctant expression.

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

Ans: [a-z0-9]

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

Ans: re.IGNORECASE

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

Ans: Dot character in general matches all but newline character. If re. DOTALL is passed as the second argument to re. compile(), then the dot will also match newline characters as well.

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 hen'

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

Ans: The re. VERBOSE argument allows us to add whitespaces and comments to the string passed to re.compile()

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:

txt = """

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)

"""

pattern = re.compile(r"\b\d{1,3}(,\d{3})\*\n")

matches = pattern.finditer(txt)

for match in matches:

print(match)

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:

pattern = re.compile(r'[A-Z][A-Za-z]+ Watanabe')

matches = pattern.finditer(txt)

for match in matches:

print(match)

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

pattern = re.compile(r'(Alice|Bob|Carol) (eats|pets|throws) (apples|cats|baseballs).', re.IGNORECASE )

matches = pattern.finditer(txt)

for match in matches:

print(match)