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

import re

re.compile("string")

#The re.compile() function returns Regex objects.

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

Hence, Raw strings are used (e.g. r"\n") so that backslashes do not have to be escaped.

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

The search() method searches a string for a specified value, and returns the position of the match. The search value can be string or a regular expression. This method returns -1 if no match is found

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

# For Matched items group() methods returns actual strings that match the pattern

import re

match = re.search('ineuron','Ineuron Full Stack Data Science Program', flags=re.IGNORECASE)

print('Output:',match.group()) Output: Ineuron

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?

# Example Program

import re

phoneNumRegex = re.compile(r'(\d\d\d)-(\d\d\d-\d\d\d\d)')

mo = phoneNumRegex.search('My number is 415-555-4242.')

print(mo.groups()) # Prints all groups in a tuple format ('415', '555-4242')

print(mo.group()) # Always returns the fully matched string 415-555-4242

print(mo.group(1)) # Returns the first group 415

print(mo.group(2)) # Returns the second group 555-4242

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?

The \. \( and \) escape characters in the raw string passed to re.compile() will match actual parenthesis characters.

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 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?

In Standard Expressions | means OR operator.

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

In regular Expressions, ? characters represents zero or one match of the preceeding group.

# Example Program

import re

match\_1 = re.search("Bat(wo)?man","Batman returns")

print(match\_1)

match\_2 = re.search("Bat(wo)?man","Batwoman returns")

print(match\_2)

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

In Regular Expressions, \* Represents Zero ore more occurances of the preceeding group, whereas + represents one or more occurances of the preceeding group.

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

{4} means that its preceeding group should repeat 4 times. where as {4,5} means that its preceeding group should repeat mininum 4 times and maximum 5 times inclusively

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

\w – Matches a word character equivalent to [a-zA-Z0-9\_]

\d – Matches digit character equivalent to [0-9]

\s – Matches whitespace character (space, tab, newline, etc.)

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

\D - > Any character that is not a numeric digit from 0 to 9.

\W - > Any character that is not a letter, numeric digit, or the underscore character.

\S - > Any character that is not a space, tab, or newline.

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

.\* is a Greedy mode, which returns the longest string that meets the condition. Whereas .\*? is a non greedy mode which returns the shortest string that meets the condition.

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

The Synatax is Either [a-z0-9] or [0-9a-z]

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

We can pass re.IGNORECASE as a flag to make a noraml expression case insensitive

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 the newline character. If re.DOTALL is passed as the second argument to re.compile(), 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?

The Ouput will be 'X drummers, X pipers, five rings, X hen'

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

The re.VERBOSE argument allows you to add whitespace 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)

import re

pattern = r'^\d{1,3}(,\d{3})\*$'

pagex = re.compile(pattern)

for ele in ['42','1,234', '6,368,745','12,34,567','1234']:

print('Output:',ele, '->', pagex.search(ele))

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 = r'[A-Z]{1}[a-z]\*\sWatanabe'

namex = re.compile(pattern)

for name in ['Haruto Watanabe','Alice Watanabe','RoboCop Watanabe','haruto Watanabe','Mr. Watanabe','Watanabe','Haruto watanabe']:

print('Output: ',name,'->',namex.search(name))

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

pattern = r'(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.'

casex = re.compile(pattern,re.IGNORECASE)

for ele in ['Alice eats apples.','Bob pets cats.','Carol throws baseballs.','Alice throws Apples.','BOB EATS CATS.','RoboCop eats apples.'

,'ALICE THROWS FOOTBALLS.','Carol eats 7 cats.']:

print('Output: ',ele,'->',casex.search(ele))