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

Import re , **re.compile(r‘maching expression’)**

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

**To consider all as raw string so that backslashes will be included.**

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

**It returns the matched data / Objects if there is no match it returns none , Datatype is : re.Match**

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

**group() method is used to get the exact matched values**

1. 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?

**import re**

**s = re.compile(r'(\d\d\d)-(\d\d\d-\d\d\d\d)')**

**m = s.search(r'111-222-3333')**

**m.group(0) - '111-222-3333' Displays all the matched data**

**m.group(1) - '111' displays the first matched pattern mentioned in first ()**

**m.group(2) - '222-3333' displays the second matched pattern mentioned in first ()**

**m.group(3) – No such group**

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? **Using Backslash**

**import re**

**s = re.compile(r'\(\)\.')**

**m = s.search("().")**

**m <re.Match object; span=(0, 3), match='().'>**

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

**import re**

**St = """matched1 asdsjg kjdshkjfdh matched2 skjldfj kjsf matched3 sjkdfhshkl**

**jkshfd matched 4 kjlfdjdkjlkd matched5"""**

**SS = 'matched\d'**

**match = re.findall(SS, St)**

**print(match)**

**['matched1', 'matched2', 'matched3', 'matched5']**

**# List of string values matched and has no groups**

**import re**

**St = """ asdsjg kjdshkjfdh matched2 skjldfj kjsf matched3 sjkdfhshkl**

**jkshfd matched matched"""**

**match = re.findall( r'(([matched]+3)|[matched]+2)', St )**

**print(match)**

**[('matched2', ''), ('matched3', 'matched3')]**

**#As the regex has groups, a list of tuples of strings is returned.**

8. In standard expressions, what does the | character mean? **It is like OR operator between two groups**

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

**This is to check the optional pattern which has the probability to occur once or not occur in that string.**

**import re**

**St = re.compile(r'india(bharath)?country')**

**res = St.search('indiabharathcountry')**

**print(res.group())**

**indiabharathcountry**

**import re**

**St = re.compile(r'india(bharath)?country')**

**res = St.search('indiacountry')**

**print(res.group())**

**indiacountry**

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

If the search pattern occurs more than once then we need to use \*

**import re**

**St = re.compile(r'india(bharath)?country')**

**res = St.search('indiabharathbharathcountry')**

**print(res.group()) 🡪 Error as Bharath occurs twice here so need to use \* to get the pattern .**

**import re**

**St = re.compile(r'india(bharath)\*country')**

**res = St.search('indiabharathbharathcountry')**

**print(res.group())**

**indiabharathbharathcountry**

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

**import re**

**St = re.compile(r'(india){4}')**

**res = St.search('indiaindiaindiaindiaindiaindiaindia')**

**print(res.group(0)) – indiaindiaindiaindia // there are seven india are there in string but it group by 4india matches only.**

**import re**

**St = re.compile(r'(india){4,5}')**

**res = St.search('indiaindiaindiaindiaindiaindiaindia')**

**print(res.group(0)) - indiaindiaindiaindiaindia**

**// create a group with max 5 india as it is mentioned to pick 4 to 5 india**

12. What do you mean by the \d, \w, and \s shorthand character classes signify in regular expressions? **To match only a single digit, word, or space character**.

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

**To match the single character which is not single digit, word, or space .**

14. What is the difference between .\*? and .+? **Starts with single char as Dot is mentioned and preceding value may contain n no of matching pattern.** .+? **Starts with single char as Dot is mentioned and preceding value may contain 0 or one match.**

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

**[a-z0-9] or [0-9a-z]**

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

**import re**

**St = re.compile(r'(india){4}'),re.IGNORECASE)**

**nm = "INDIAindiaindiaindia"**

**res = St.search(nm)**

**print(res.group(0))**

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

**import re**

**St = re.compile(r'(.)')**

**nm = """**

**#**

**"""**

**res = St.search(nm)**

**print(res.group(0)) # Matches with any char except newline , New line not accepted.**

**St = re.compile(r'(.),re.DOTALL') // New line will be included**

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

**numReg = re.compile(r'\d+')**

**numReg.sub('X', '11 drummers, 10 pipers, five rings, 4 hen')**

**'X drummers, X pipers, five rings, X hen'**

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

**This re.VERBOSE argument in re.compile allows to add space and comments to the 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)

**re.compile(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)

**re.compile(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.'

**re.compile(r'(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.', re.IGNORECASE)**