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

Ans: **re.compile() helps you create regex objects.**

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

Ans: **Sometimes in Regex we use raw strings to find pattern in a large chunk of text and by including raw string, we can ignore escape characters which may exist in the pattern matching. For example:**

Graphical user interface, text, application

Description automatically generated

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

Ans: **Search() method only returns the first occurrence of the matching pattern like below:**

Text

Description automatically generated

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

Ans: **here if we use the match method then it looks for the text at the beginning of the text only but if we use findall() then we can find all of the occurrence of the pattern in the text.**

Text

Description automatically generated

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 match, group 1 looks for the first parenthesis and group 2 is for the second 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: **escape characters like backslash can be used to include periods and parenthesis.**

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: **it returns a list of strings but if we have group in regex then it returns a list of string tuples.**

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

Ans: **it is a bitwise OR operator in standard expressions.**

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

Ans: **in regular expression, we can find characters in a large chunk of text. However, there are some special characters which hold a different meaning.**

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

Ans: **these both are quantifiers. \* is used to find 0 or more repetitions and + is used to find 1 or more repetitions like below:**



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

Ans: **{4} will match and give you 4 instance of the proceeding characters and {4,5} gives you 4 and 5 instances of the proceeding characters. For example:**

Text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

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

Ans: **\d is used for digits, \w is used for alphanumeric characters and \s is used for whitespaces.**

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

Ans:

**\D: matches any non-digit character**

**\W: matches any non-alpha numeric character**

**\S: matches any non-whitespace character**

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

Ans: **.\* is considered greedy and .\*? is considered non greedy.**

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

Ans:

**import re**

**pattern=re.compile(r'[0-9a-z]')**

**match=pattern.finditer(text)**



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

Ans: **re.IGNORECASE can be used to make the regex expression case insensitive.**

Graphical user interface, text, application, email

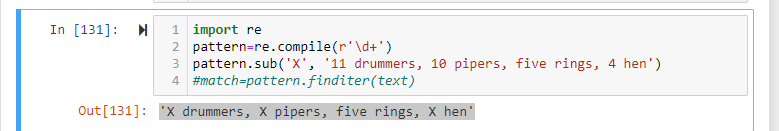
Description automatically generated

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

Ans: **This . character is used to match every character except next line character. If you pass re.DOTALL then it matches next line character as well.**

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

Ans:



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

Ans:  **this allows you to spread the regular expression over multiple lines with comments in the case of complex regular expression.**

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:

**import re**

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

**match=pattern.finditer(text)**

Graphical user interface, text, application

Description automatically generated

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:

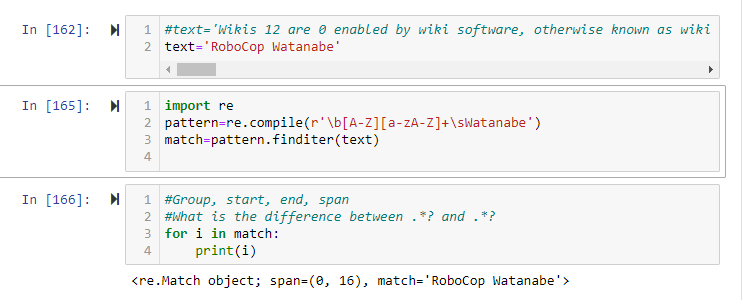
**import re**

**pattern=re.compile(r'\b[A-Z][a-zA-Z]+\sWatanabe')**

**match=pattern.finditer(text)**

**for i in match:**

**print(i)**



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**:

**import re**

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

**match=pattern.finditer(text)**

**for i in match:**

**print(i)**

