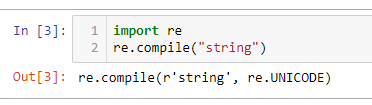
1. What is the name of the feature responsible for generating Regex objects?

A). import re

re.compile(pattern)🡪 returns the regex objects



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

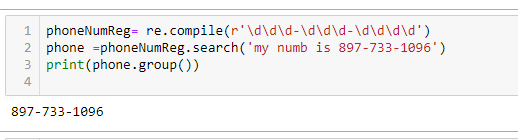
A). Raw strings are used so that backslash do not have be escaped

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

A). Seacrch() will searches a specified string for specified values and returns the position of the match, if no match found -1 will return. String or regular expressions can be searched.

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

A) By matchingobject.group() will get the matched string.



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?

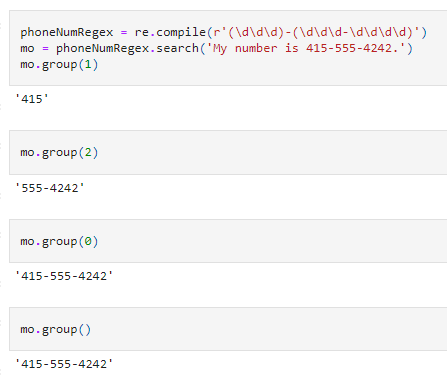
A) regex: (\d\d\d)-(\d\d\d-\d\d\d\d)

The first set of parentheses in a regex string will be group 1.

The second set will be group 2.

By passing the integer 1 or 2 to the group() match object method, you can grab different parts of the matched text.

Passing 0 or nothing to the group() method will return the entire matched text



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?

A) Periods and parentheses can be escaped 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?

A) 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?

A). The ‘/’ character identifies ‘either’ or ‘OR’ matching between two groups

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

A). The ‘?’ character matches either “ ‘0’ or one of the preceding group” or used to preceding non greedy matching.

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

A). the ‘+’ matches one or more where as ‘\*’ matches zero or more

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

A). the {4} matches exactly 4 instances of the preceding group. {4,5} matches between 4 and 5 instances.

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

A). The \d, \w, \ s are shorthand characters used to match single digit, word and space.

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

A). The \D, \W, \S are used to match which are not single digit, word and space respectively.

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

A). (.\*?) matches any character (.) any number of times (\*), as few times as possible to make the regex match (?). You'll get a match on any string, but you'll only capture a blank string because of the question mark. This feature is much more useful when you have a more complicated regex. Here, the parser doesn't have to capture anything at all to get a match: the asterisk allows any number of characters in the capturing group, while the question mark makes the parser save as many as possible from the input text for later, resulting in nothing being captured.

(.\*)? captures a group zero or one times (?). That group consists of a run of any length (\*) of any character (.). This also will match anything, but it will capture the first line, since the dot matches anything except a newline.

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

A). either [0-9a-z] or [a-z0-9]

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

A). Passing re.I or re.IGNORECASE as the second argument to re.compile() will make the matching case insensitive.

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

A). 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?

A). 'X drummers, X pipers, five rings, X hens'

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

A). he 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)

A). re.compile(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)

A). 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.'

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