

-----  
-----QUESTIONS-----  
-----  
-----

1. What is the name of the feature responsible for generating Regex objects?
2. Why do raw strings often appear in Regex objects?
3. What is the return value of the search() method?
4. From a Match item, how do you get the actual strings that match the 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?
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?
7. The findall() method returns a string list or a list of string tuples. What causes it to return one of the two options?
8. In standard expressions, what does the | character mean?
9. In regular expressions, what does the character stand for?
10. In regular expressions, what is the difference between the + and \* characters?
11. What is the difference between {4} and {4,5} in regular expression?
12. What do you mean by the \d, \w, and \s shorthand character classes signify in regular expressions?
13. What do means by \D, \W, and \S shorthand character classes signify in regular expressions?
14. What is the difference between .\*? and .\*?
15. What is the syntax for matching both numbers and lowercase letters with a character class?
16. What is the procedure for making a normal expression in regex case insensitive?
17. What does the . character normally match? What does it match if re.DOTALL is passed as 2nd argument in re.compile()?
18. If numReg = re.compile(r'\d+'), what will numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4 hen') return?
19. What does passing re.VERBOSE as the 2nd argument to re.compile() allow to do?
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)
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)

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.'

-----  
-----ANSWERS-----  
-----  
-----

- 1- The `re.compile()` function returns Regex objects.
- 2- Raw strings are used so that backslashes do not have to be escaped
- 3- The `search()` method returns Match objects.
- 4- The `group()` method returns strings of the matched text.
- 5- Group 0 is the entire match, group 1 covers the first set of parentheses, and group 2 covers the second set of parentheses.
- 6- Periods and parentheses can be escaped with a backslash: `\.`, `\(`, and `\)`.
- 7- 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- The `|` character signifies matching "either, or" between two groups.
- 9- The `?` character can either mean "match zero or one of the preceding group" or be used to signify nongreedy matching.
- 10- The `+` matches one or more. The `*` matches zero or more.
- 11- `{4}` matches exactly four instances of the preceding group. `{4,5}` matches between four and five instances.
- 12- The `\d`, `\w`, and `\s` shorthand character classes match a single digit, word, or space character, respectively.
- 13- The `\D`, `\W`, and `\S` shorthand character classes match a single character that is not a digit, word or space character, respectively.
- 14- `*?` is used instead of the greedy `.*` to prevent overmatching.
- 15- Either `[0-9a-z]` or `[a-z0-9]`

16- Passing `re.I` or `re.IGNORECASE` as the second argument to `re.compile()` will

make the matching case insensitive.

17- 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- `'X drummers, X pipers, five rings, X hens'`

19- The `re.VERBOSE` argument allows you to add whitespace and comments to the string passed to `re.compile()`.

20- `re.compile(r'^\d{1,3}(\,\d{3})*$')` will create this regex, but other regex strings can produce a similar regular expression

21- `. re.compile(r'[A-Z][a-z]*\sWatanabe')`

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