

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

Ans:- The feature responsible for generating Regex objects is the `re.compile()` function

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

Ans:- Raw strings often appear in Regex objects because the regular expression engine uses the backslash (`\`) character for its own escaping purpose. Using raw strings, denoted by prefixing the string with an `r`, treats the backslash as a literal character and not an escape character.

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

Ans:- The `search()` method in regex returns a Match object if there is a match anywhere in the string. If no match is found, it returns `None`

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

Ans:- From a Match item, you can get the actual strings that match the pattern using the `group()` method

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:- In the regex created from `r'(\d\d\d)-(\d\d\d-\d\d\d\d)'`, group zero covers the entire match, group 1 covers the first set of parentheses (the first three digits), and group 2 covers the second set of parentheses (the remaining digits)

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:- In standard expression syntax, to specify that you want a regex to match actual parentheses and period characters, you can use the backslash (`\`) character to escape them

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:- The `findall()` method in regex returns a list of strings if the pattern has no groups. If one or more groups are present in the pattern, it returns a list of tuples of strings

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

Ans:- In standard expressions, the vertical bar (`|`) character is used to denote logical OR

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

Ans:- In regular expressions, the `.` character is a special character known as a metacharacter

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

Ans:- In regular expressions, both `+` and `*` are quantifiers. The `+` character means "one or more" of the preceding element, while `*` means "zero or more" of the preceding element

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

Ans:- In regular expressions, `{4}` matches exactly four instances of the preceding group, while `{4,5}` matches between four and five instances

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

Ans:- The `\d`, `\w`, and `\s` shorthand character classes in regular expressions match a single digit, word, or space character, respectively

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

Ans:- The \D, \W, and \S shorthand character classes in regular expressions match a single character that is not a digit, word, or space character, respectively

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

Ans:- .* is a greedy match that will match as much as possible to make the regex match. On the other hand, .*? is a non-greedy or lazy match that will match as little as possible to make the regex match.

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

Ans:- To match both numbers and lowercase letters with a character class in regex, you can use **[0-9a-z]**

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

Ans:- To make a regular expression case-insensitive in Python, you can pass re.I or re.IGNORECASE as the second argument to re.compile()

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

Ans:- 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?

Ans:- If numReg = re.compile(r'\d+'), then numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4 hen') will return 'X drummers, X pipers, five rings, X hen'

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

Ans:- Passing re.VERBOSE as the second argument to re.compile() allows you to add whitespace and comments to the string passed to re.compile(), which can make your regular expression more readable

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:- A regex that matches a number with a comma for every three digits could be written as:

`^\d{1,3}(\,\d{3})*$`. This will match '42', '1,234', and '6,368,745', but not '12,34,567' or '1234'.

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:- A regex that matches the full name of someone whose last name is Watanabe could be written as: `[A-Z][a-z]*\sWatanabe`. This will match 'Haruto Watanabe', 'Alice Watanabe', and 'RoboCop Watanabe', but not 'haruto Watanabe', 'Mr. Watanabe', 'Watanabe', or 'Haruto watanabe'.

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:- 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 could be written as:

`^(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.$`. This should be case-insensitive so you can add `re.IGNORECASE` when compiling this regex.