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

Ans:

The `re.compile()` function is responsible for generating Regex object.

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

Ans:

Raw strings are often apperin Regex objects because they allow backslashes to be treated as normal characters instead of escape characters.

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

Ans:

The `search()` method in Regex module returns a match object if the pattern is found in the searched string, and returns “None” if the pattern is not found.

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

Ans:

We can use the group() method of the match object to extract the actual strings that match the pattern in Python.

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?

Ans:

In the regex `r'(\d\d\d)-(\d\d\d-\d\d\d\d)'`, group 0 covers the entire matched string, group 1 covers the first capturing group `(\d\d\d)`, and group 2 covers the second capturing group `(\d\d\d-\d\d\d\d)`.

1. 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:

To match real parentheses and periods in a regex, you need to escape them with a backslash (\). For example, to match the string "(hello)", we can use the regex "\(\hello\)".

1. 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 returns a string list when there are no capturing groups in the regular expression passed to it. If there are one or more capturing groups, then the method returns a list of string tuples.

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

Ans:

The | character in Python is used as a bitwise OR operator.

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

Each characters has a specific function in regular expressions such as matching any character, matching the beginning or end of a line, creating character classes, or specifying a range of occurrences for a character or group.

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

Ans:

The difference between “+” and ”\*” characters is the “+” matches one or more occurrences of the preceding character, while the “\*” character matches zero or more occurrences of the preceding character.

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

Ans:

The difference between {4} and {4,5} in regular expression is {4} matches exactly four occurrences of the preceding character and {4,5} matches between four and five occurrences of the preceding character.

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

Ans:

* The “\d” matches any digits (0-9).
* The “\w” matches any word character, which includes uppercase and lowercase letters, digits, and the underscore characters.
* The “\s” matches any whitespace character, including spaces, tabs, and newlines.

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

Ans:

* The “\D” matches any character that is not a digit (0-9).
* The “\W” matches any character that is not a word character, which includes uppercase and lowercase letters, digits, and the underscore characters.
* The ”\S” matches any character that is not a whitespace character, including spaces, tabs, and newlines.

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

Ans:

There is no difference between “.\*?” and “.\*?”. Both are used to match any character (except for a newline) zero or more times by considering the pattern.

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

Ans:

The syntax to match both numbers and lowercase letters with a character class is “r[0-9a-z]+”

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

Ans:

The procedure for making a normal expression in regax case insensitive is

re.findall(pattern,string, flags=re.IGNORECASE)

1. 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 for a newline character.

The “re.DOTALL” flag as the second argument to re.compile() then the “.” character will match any character including a newline character as well.

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

Ans:

output:”X drummers, X pipers, five rings, X hen”

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

Ans:

The re.VERBOSE can be passed as the second argument to re.compile(),it allows you to write regular expressions that are easier to read and understand by adding comments and whitespace to the pattern.

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 = r'^\d{1,3}(,\d{3})\*$'**

print(re.match(pattern, '42,323,456'))

print(re.match(pattern, '1,234'))

print(re.match(pattern, '6,368,745'))

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

names = [“Haruto Watanabe”,”Alice Watanabe”, “RoboCop Watanabe”,”haruto Watanabe”, “Mr. Watanabe”, “Watanabe”, “Haruto watanabe”]

**regex =r"^[A-Z][a-z][A-Za-z]\* Watanabe$"**

for name in names:

if re.match(regex, name):

print(f"{name} matches the regex")

else:

print(f"{name} does not match the regex")

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

sentence = input("Enter a sentence: ")

**regex= r'^(Alice|Bob|Carol)\s+(eats|pets|throws)\s+(apples|cats|baseballs)\.$'**

if re.search(regex, sentence, re.IGNORECASE):

print("The sentence matches the pattern.")

else:

print("The sentence does not match the pattern.")