ADITYA-AMIN ASSIGN : 07

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

re.compile is the feature which we use by importing re library for generating regex objects.

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

Raw strings i.e. backslash (“\”) has always be used in regex objects so that we can use it without invoking its proper meaning, there is a risk of collision between regex objects and strings, to avoid this collision we use raw strings in regex.

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

the search() method is used to search for a regular expression pattern within a string. If the pattern is found in the string, the method returns a match object, which contains information about the first occurrence of the pattern, such as its starting and ending positions in the string.

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

you can get the actual strings that match the pattern from a match object by calling the group() method on the match object. This method returns the string that was matched by the regular expression pattern.

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?

Group 0 (or group() with no argument) covers the entire matched string.

Group 1 covers the first capturing group (\d\d\d), which matches a sequence of three digits.

Group 2 covers the second capturing group (\d\d\d-\d\d\d\d), which matches a sequence of three digits followed by a hyphen and another sequence of four digits.

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?

In standard regular expression syntax, parentheses and intervals have special meanings. To match literal parentheses and periods, you need to escape them with a backslash \ character. For example, to match a literal left parenthesis (, you can use the regex \(. Similarly, to match a literal right parenthesis ), you can use the regex \).

1. The findall() method returns a string list or a list of string tuples. What causes it to return one of the two options?

The findall() method in Python's re module returns a list of all non-overlapping matches of a regular expression pattern in a string.

The type of the returned list depends on the presence of capturing groups in the regular expression pattern.

If the regular expression pattern contains no capturing groups, then findall() returns a list of strings. Each element of the list corresponds to a complete match of the pattern in the input string. If the regular expression pattern contains one or more capturing groups, then findall() returns a list of tuples. Each tuple contains the captured groups of a single match of the pattern in the input string.

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

The "|" character is often used as a logical operator in regular expressions (also known as regex). It is known as the "pipe" or "alternation" operator, and it allows you to match one of several possible patterns.

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

"." (dot): Matches any single character except for a newline.

"^" (caret): Matches the start of a string or the start of a line.

"$" (dollar sign): Matches the end of a string or the end of a line.

"\*" (asterisk): Matches zero or more occurrences of the preceding character or group.

"+" (plus): Matches one or more occurrences of the preceding character or group.

"?" (question mark): Makes the preceding character or group optional, matching it zero or one times.

"\d": Matches any digit character (0-9).

"\w": Matches any word character (letters, digits, or underscore).

"\s": Matches any whitespace character (space, tab, newline, etc.).

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

In regular expressions, the "+" and "\*" characters are quantifiers that specify how many times the preceding character or group should be matched.

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

{4} specifies that the preceding character or group should be matched exactly 4 times.

{4,5} specifies that the preceding character or group should be matched between 4 and 5 times.

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

"\d" matches any digit character (0-9).

"\w" matches any "word" character, which includes letters (a-z, A-Z), digits (0-9), and underscore (\_).

"\s" matches any whitespace character, such as a space, tab, or newline.

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

In regular expressions, the uppercase versions of the shorthand character classes "\D", "\W", and "\S" are used to match the opposite of their lowercase counterparts. Here's what they signify:

• "\D" matches any character that is not a digit (0-9).

• "\W" matches any character that is not a "word" character, which includes any character that is not a letter (a-z, A-Z), digit (0-9), or underscore (\_).

• "\S" matches any character that is not whitespace, such as a non-space character, a tab, or a newline.

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

Expressions are identical, there is no difference between them.

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

To match both numbers and lowercase letters with a character class in regular expressions, you can use the following syntax:

[0-9a-z]

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

In Python, you can add the "re.IGNORECASE" or "re.I" flag to the regular expression.

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

In regular expressions, the "." character (dot) normally matches any character except for a newline character (\n).

If you pass the re.DOTALL (or re.S) flag as the second argument to re.compile() in Python, the dot character will match any character, including newline characters. This flag changes the behavior of the dot character so that it matches everything.

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

If we use the regular expression numReg = re.compile(r'\d+') and apply the sub() method to the string '11 drummers, 10 pipers, five rings, 4 hen' with the replacement string 'X', the resulting string will have all sequences of one or more digits replaced with the letter 'X'.

So the expression numReg.sub('X', '11 drummers, 10 pipers, five rings, 4 hen') will return the following string:

'X drummers, X pipers, five rings, X hen'

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

In Python's re module, passing re.VERBOSE as the second argument to re.compile() allows you to use a verbose regular expression syntax, which makes it easier to write and read complex regular expressions.

1. 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)

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

This regular expression matches numbers that start with 1 to 3 digits, followed by any number of comma-separated groups of exactly 3 digits. This allows for numbers like '42', '1,234', and '6,368,745' to match.

However, it will not match '12,34,567' (which has only two digits between the commas) or '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)

^[A-Z][a-z]\*\sWatanabe$

Explanation:

• ^ - matches the start of the string

• [A-Z][a-z]\* - matches a single uppercase letter followed by zero or more lowercase letters, representing the first name

• \s - matches a single whitespace character (space, tab, etc.)

• Watanabe - matches the literal string "Watanabe"

• $ - matches the end of the string

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.'

The following regex should match 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 either apples, cats, or baseballs; and the sentence ends with a period. This regex is case-insensitive:

^(Alice|Bob|Carol)\s+(eats|pets|throws)\s+(apples|cats|baseballs)\.$

• ^ - matches the start of the string

• (Alice|Bob|Carol) - matches one of the three given names (case-insensitive)

• \s+ - matches one or more whitespace characters

• (eats|pets|throws) - matches one of the three given verbs (case-insensitive)

• \s+ - matches one or more whitespace characters

• (apples|cats|baseballs) - matches one of the three given nouns (case-insensitive)

• \. - matches a period (escaped with a backslash)

• $ - matches the end of the string