Q1. What is the benefit of regular expressions?

Answer:

Regular expressions, also known as regex, are used to match text strings, such as specific characters, words, or character patterns. This means we can match and extract any string of text using regular expressions. It helps developers write less and cleaner code. It also avoids using several if/else statements. It provide an efficient way to identify patterns in a string and are thus useful for manipulation, searching, and other string operations.

Q2. Describe the difference between the effects of "(ab)c+" and "a(bc)+." Which of these, if any, is the unqualified pattern "abc+"?

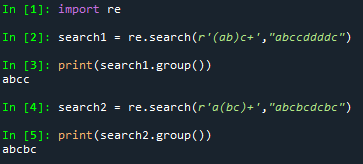
Answer:

🡪 The pattern (ab)c+ looks for only one occurrence of "ab" and one or more occurrences of "c".

🡪 The pattern a(bc)+ matches only one occurrence of "a" and one or more occurrences of "bc".

🡪 The pattern (ab)c+ is the unqualified pattern "abc+"

Example:



Q3. How much do you need to use the following sentence while using regular expressions?

import re

Answer:

Every time we need to work with regular expression, this statement needs to be imported before using regular expressions.

Q4. Which characters have special significance in square brackets when expressing a range, and under what circumstances?

Answer:

🡪 If the set consists of several characters, e.g. from a to z or from 1 to 9, it is boring to list them in the set. Instead, we can use a range of characters between square brackets. For example, [a-z] is a character in the range a-z and [0-9] is a number 0-9.

🡪 We can also use multiple ranges within the same square brackets. For example, [a-z0-9] contains two ranges that match a character that is a-z or a number 0-9.

🡪 Similarly, we can use one or more characters between square brackets, such as [\d\s] for a number or a space. Also, we can mix character with characters. For example, [\d\_] matches a number or underscore.

🡪 We use the caret (^) operator at the beginning to negate a set or range. For example, the range [^0-9] matches any character except a number. This is the same as the \D character set.

Q5. How does compiling a regular-expression object benefit you?

Answer:

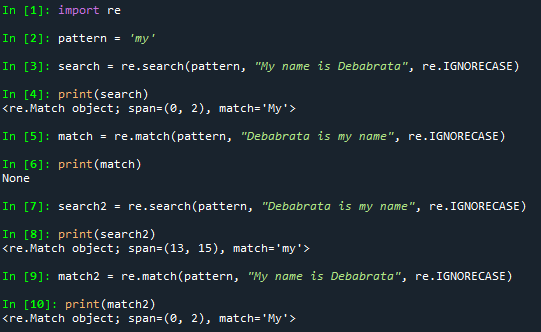
We can combine regular expressions into patterns that can be used for pattern matching. It also helps to find the pattern again without re-writting it. Also, it is faster compared to using pattern and other re functions.

Q6. What are some examples of how to use the match object returned by re.match and re.search?

Answer:

The re,match() and re.search() functions searches for a sub-string in a string and returns a matching object if found, otherwise it returns nothing. Both functions return the first match of a sub-string found in the string, however re.match() only looks for the beginning of the string and returns a matching object if found and if a sub-string match is found somewhere in the middle of the string, it returns nothing, while re.search() searches the entire string, even if the string contains multiple lines, and tries to match the substring on all lines of the string.

Example:



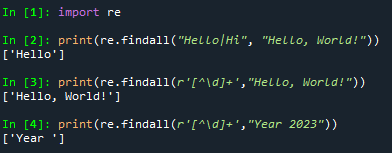
Q7. What is the difference between using a vertical bar (|) as an alteration and using square brackets as a character set?

Answer:

🡪 vertical bar (|): it checks if one of the strings exists in the search string.

🡪 square brackets ([]): it searches for any character in a string.

Example:



Q8. In regular-expression search patterns, why is it necessary to use the raw-string indicator (r)? In   replacement strings?

Answer:

Raw-string indicator denoted by “r” operator and it indicates that the leading characters are single characters, not escapes. This changes the interpretation of the string literal. Such letters are stored as they appear. Regular expression search patterns use raw strings, so there is no need to omit the backslash.