**ADITYA AMIN**

**ASSIGN :16**

Q1. What is the benefit of regular expressions?

Pattern Matching: Regular expressions allow you to search for patterns within text. You can define complex patterns using a combination of characters, metacharacters, and quantifiers to match specific strings or patterns of characters.

Text Validation and Extraction: Regular expressions enable you to validate and extract specific information from text inputs. For example, you can use regular expressions to validate email addresses, phone numbers, URLs, or any other patterns you define.

Text Manipulation and Transformation: Regular expressions offer powerful string manipulation capabilities.

Efficient and Compact Code: Regular expressions allow you to express complex matching patterns using a compact syntax. Instead of manually writing custom parsing or matching algorithms, you can leverage the power of regular expressions to achieve the same result with fewer lines of code.

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

"(ab)c+" matches a sequence starting with "ab" followed by one or more occurrences of "c".

"a(bc)+" matches a sequence starting with "a" followed by one or more occurrences of "bc".

"abc+" matches a sequence starting with "a" followed by one or more occurrences of "b", with an optional "c" at the end.

Each of these regular expressions has its own specific pattern matching behavior, and the unqualified pattern "abc+" matches a specific sequence consisting of "a" followed by one or more "b"s, optionally followed by "c".

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

import re

The sentence "import re" is used in Python when you want to import the re module, which provides support for regular expressions. This statement is required if you want to use regular expressions in your Python code.

Therefore, whenever you need to work with regular expressions in Python, it is necessary to include the "import re" statement at the beginning of your code to import the re module.

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

the dash (-) is used to define a range of characters within square brackets, and the caret (^) is used to negate a character class, indicating that the pattern should match any character not in the specified range.

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

Improved Performance: When you compile a regular expression pattern into a regex object using the re.compile() function, the pattern is pre-compiled into an internal representation that can be executed more efficiently.

Code Readability and Reusability: By compiling a regular expression into an object, you can assign it to a variable with a meaningful name. This enhances code readability and makes it easier to understand the purpose of the pattern.

Access to Additional Methods: Regular expression objects provide additional methods beyond the ones available in the re module. For example, you can use methods such as search(), match(), findall(), and sub() directly on the compiled object without explicitly specifying the pattern each time.

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

Extracting the Matched String:

import re

pattern = r'\d+'

text = 'There are 123 apples.'

match = re.search(pattern, text)

if match:

matched\_str = match.group()

print(matched\_str) # Output: 123

Accessing Matched Groups:

import re

pattern = r'(\d+)-(\w+)'

text = 'Order ID: 123-ABC'

match = re.search(pattern, text)

if match:

order\_id = match.group(1)

product\_code = match.group(2)

print(order\_id, product\_code) # Output: 123 ABC

Accessing Start and End Positions of Matched String:

import re

pattern = r'apple'

text = 'I have an apple.'

match = re.search(pattern, text)

if match:

start\_pos = match.start()

end\_pos = match.end()

print(start\_pos, end\_pos) # Output: 10 15

Using Matched String for Substitution:

import re

pattern = r'apple'

text = 'I have an apple.'

replaced\_text = re.sub(pattern, 'orange', text)

print(replaced\_text) # Output: I have an orange.

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

Vertical Bar (|): The vertical bar is used as an alteration operator, also known as the "pipe" operator. It allows you to specify multiple alternative patterns, and the regular expression engine will match any of the patterns. It behaves like an "OR" operator.

Square Brackets ([]): Square brackets are used to define a character set or character class. Inside the square brackets, you can specify a set of characters, and the regular expression engine will match any single character that matches any of the characters within the set. It behaves like an "OR" operator for individual characters.

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

In regular expression search patterns, it is common to use the raw-string indicator (r) to define the pattern. The raw-string indicator is not strictly necessary, but it is recommended and widely used to avoid potential issues with backslashes and escape sequences.