Q1. What is the benefit of regular expressions?

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

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

import re

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

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

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

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

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

A1. Regular expressions are a powerful tool for pattern matching and text manipulation. They allow you to search for specific patterns in a text string and perform various operations such as replacing, extracting, or validating text based on those patterns.

A2. "(ab)c+" matches a string that has one or more occurrences of the sequence "abc", where "abc" is a single unit. For example, "abc", "abcbc", and "abcbcabcbc" would all match. "a(bc)+" matches a string that has one or more occurrences of the sequence "bc", where "bc" is a single unit preceded by an "a". For example, "abc", "abcbc", and "abcbcabcbc" would all match. "abc+" is an unqualified pattern that matches a string that has one or more occurrences of the letter "c" immediately following the letter "a" and "b".

A3. The statement "import re" is used at the beginning of a Python script or module to import the Python's built-in regular expression module, "re". It needs to be used whenever regular expressions are used in a Python program.

A4. In square brackets, the characters "-", "^", and "]" have special significance when expressing a range. The dash "-" is used to specify a range of characters, "^" is used to negate a character set, and "]" is used to terminate a character set. For example, "[a-z]" matches any lowercase letter, "[^abc]" matches any character that is not "a", "b", or "c", and "[]]" matches a single "]" character.

A5. Compiling a regular-expression object benefits you by improving the performance of your regular-expression operations. When you compile a regular expression, Python converts it into an internal representation that can be executed more quickly than the original pattern string. This is particularly useful if you need to perform the same regular-expression operation multiple times.

A6. The match object returned by re.match and re.search can be used to extract information from a matched string. Some examples of methods you can use on a match object include group(), groups(), start(), and end(). For example, match.group() returns the entire matched string, match.groups() returns a tuple of all the matched groups, and match.start() and match.end() return the starting and ending positions of the matched string.

A7. A vertical bar (|) is used to specify alternatives, meaning that the regular expression can match any one of the patterns separated by the vertical bar. Square brackets, on the other hand, are used to specify a set of characters that can match any one of the characters inside the brackets. For example, the regular expression "[abc]" matches any one of the characters "a", "b", or "c".

A8. In regular-expression search patterns, it is necessary to use the raw-string indicator (r) to prevent escape sequences from being interpreted by the Python interpreter before they are passed to the regular expression engine. In replacement strings, it is necessary to use the raw-string indicator to prevent the backslash () character from being interpreted as an escape character by the Python interpreter.