Q1. What is the benefit of regular expressions?

Regular expressions are used to search and replace operations.The most common uses are:

1. to look for a sub-string that matches a certain pattern and replace it with something else.
2. extract the required values or texts from a word/sentence..

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

None of them is unqualified. The difference is r"(ab)c+" matches the pattern with any string that has the substring "ab" followed by "c" getting repeated.,eg, abc, abcc,abcccc.....

Whereas for r"a(bc)+", it looks for strings that matches the pattern "a" followed by repetitions of the group string "bc". e.g., abc,abcbc, abcbcbc,....and so on.

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?

A closing square bracket has special meaning, terminating the character set; therefore, a closing bracket must be escaped with a backslash to be interpreted literally: “]”

The minus sign (-) has special meaning unless it occurs at the very beginning or end of the character set, in which case it is interpreted as a literal minus sign.

A caret (^) has special meaning at the beginning of a range but not elsewhere.

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

Ans. Regular expression pattern can be combined into pattern objects, which can be used for pattern matching. Moreoevr it can be used to search a pattern again without having to rewrite it.

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

Ans. re.search() searches for the whole string even if the string contains multi-lines and tries to find a match of the substring in all the lines of string. It returns the first occurrence or match of searched substring .

re.match() searches only from the beginning of the string and return match object if found. But if a match of substring is found somewhere in the middle of the string, it returns none. It also returns only the first occurrence or match of search substring.

**import** re

substring **=**'have'

line1 **=** "I have to have coffee "

line2 **=**"Have you ever had starbucks coffee before? If not, let's have it today."

*# Use of re.search() Method*

print(re**.**search(substring, line1, re**.**IGNORECASE))

*#Returns the first/only occurence of "have"*

*# Use of re.match() Method*

print(re**.**match(substring, line1, re**.**IGNORECASE))

*#Returns None since "have" isn't the first occurence in line1*

*# Use of re.search() Method*

print(re**.**search(substring, line2, re**.**IGNORECASE))

*#Returns the first occurence of "have"*

*# Use of re.match() Method*

print(re**.**match(substring, line2, re**.**IGNORECASE))

*#Returns the match since "have" is the first occurence in line1*

<re.Match object; span=(2, 6), match='have'>

None

<re.Match object; span=(0, 4), match='Have'>

<re.Match object; span=(0, 4), match='Have'>

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

Ans. When (|) is used the pattern searches for 'or' option i.e \|\ means it searches for as \ or \. When both \ and \ occur in the searched string, the first occurrence of matching text will be returned as the Match object

Using character set in square brackets is used, it searches for all the character set in the square bracket and returns the match if found.

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

Ans. Raw strings are used in regular expression search patterns, when the regular expression includes a backslash character