**Q1. What is the benefit of regular expressions?**

The main benefit of regular expressions is, it is very effective and faster in finding substrings and patterns in a large string than using any other string functions with lot of if-else statements.

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

All three expression will match different patterns

* “(ab)c+” – will match all(“+” – matches one or more) the pattern where abc has come together but it will return only the group “ab” (“()” – groups the pattern).
* “a(bc)+” – will match all(“+” – matches one or more) the pattern where abc has come together but it will return only the group “bc” (“()” – groups the pattern).
* “abc+” – will match all(“+” – matches one or more) the pattern where abc has come together with multiple “c”s

None of them is unqualified as it may be used in different situations

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

**import re**

we need to import re only one time while using regular expressions.

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

+ and ^ have special significance in square brackets.

+ in Square brackets means any character while outside it searches for one or more matches

^ not inside sqare brackets while outside it is used if string startswith a character.

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

Compiling regular – expressions allow us to reuse the pattern without writing every time.

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

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.

While 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.

Following are the example

import re

Substring ='string'

String1 ='''We are learning regex. regex is very useful for string matching and It is fast too.'''

String2 ='''string We are learning regex. regex is very useful for string matching and It is fast too.'''

# Use of re.search() Method

print(re.search(Substring, String1, re.IGNORECASE))

output - <re.Match object; span=(75, 81), match='string'>

# Use of re.match() Method

print(re.match(Substring, String1, re.IGNORECASE))

output - None

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

Vertical bar acts as “or” means It means it will return the string if any of the string separated by vertical line matches. Whereas “[ ]” searches for individual characters or a range of characters.

Ex – “Apple | Banana”, [A - Z]

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

When we use “ \ ” in the regular expression but we don’t want it to consider as the escape character we have to use raw strings.