Q1. What is the benefit of regular expressions?

**Ans:** Regular expressions are useful in search and replace operations. The typical use case is to look for a sub-string that matches a pattern and replace it with something else. Most APIs using regular expressions allow you to reference capture groups from the search pattern in the replacement string.

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

**Ans:** Causes the regular-expression evaluator to look at all of expr as a single group. There are two major purposes for doing so. First, a quantifier applies to the expression immediately preceding it; but if that expression is a group, the entire group is referred to. For example, (ab)+ matches “ab”, “abab”, “ababab”, and so on.

Python regular-expression evaluator interprets the meaning of the pattern differently because of the parentheses; specifically, it is the group “ab” that is repeated.  
a(bc)+ Match “a” exactly. This forms a unit that matches one or more instances of “bc”.

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

import re

**Ans:** Regular Expression Syntax. A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).

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

**Ans:** Square brackets match something that your kind of don't know about a string you're looking for. If you are searching for a name in a string but you are not sure of the exact name you could use instead of that letter a square bracket. Everything you put inside these brackets are alternatives in place of one character. Square brackets ([]) designate a character class and match a single character in the string. Inside a character class, only the character class metacharacters (backslash, circumflex anchor, and hyphen) have special meaning.

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

**Ans:** compile (pattern, repl, string): We can combine a regular expression pattern into pattern objects, which can be used for pattern matching. It also helps to search a pattern again without rewriting it.

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

**Ans:** Both return the first match of a substring found in the string, but re. match () searches only from the beginning of the string and return match object if found. ... 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.

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

**Ans:** The vertical bar is a regex "or" means "a or b". Square brackets are a character class meaning "any character from a or b.

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 so that backslashes do not have to be escaped. The search() method returns Match objects.