Q1. What is the benefit of regular expressions?

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

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

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

import re

in every case

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

Virtually all regular expression metacharacters lose their special meaning and are treated as regular characters when used within square brackets.

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

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?

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

*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?

Raw strings **help you get the "source code" of a RegEx safely to the RegEx parser**, which will then assign meaning to character sequences like \d , \w , \n , etc.