Q1. Does assigning a value to a string's indexed character violate Python's string immutability?

Ans = The string itself is immutable but the label can change. Assigning a new value to an existing variable is perfectly valid. Python does not have constants. This is independent from data type mutability.

Q2. Does using the += operator to concatenate strings violate Python's string immutability? Why or why not?

Ans = It violates the rules of how ID values and += are supposed to work - the ID values produced with the optimization in place would be not only impossible, but prohibited, with the unoptimized semantics - but the developers care more about people who would see bad concatenation performance and assume Python sucks

Q3. In Python, how many different ways are there to index a character?

Ans = We can access characters in a String in Two ways : Accessing Characters by Positive Index Number. Accessing Characters by Negative Index Number

Q4. What is the relationship between indexing and slicing?

Ans = “Indexing” means referring to an element of an iterable by its position within the iterable. “Slicing” means getting a subset of elements from an iterable based on their indices.

Q5. What is an indexed character's exact data type? What is the data form of a slicing-generated substring?

Explore string operators, index values, length of strings, string slicing ... Types that are comprised of smaller pieces are called collection data types.

Q6. What is the relationship between string and character "types" in Python?

Ans = Like many other popular programming languages, strings in Python are arrays of bytes representing unicode characters. However, Python does not have a character data type, a single character is simply a string with a length of 1.

Q7. Identify at least two operators and one method that allow you to combine one or more smaller strings to create a larger string.

Ans = You can combine both string variables and string literals using the “+” operator. However, there's another method that allows an easy way of concatenating multiple strings. It is using the in-place (+=) operator.

Q8. What is the benefit of first checking the target string with in or not in before using the index method to find a substring?

Q9. Which operators and built-in string methods produce simple Boolean (true/false) results?

Ans = The logical operators and, or and not are also referred to as boolean operators. While and as well as or operator needs two operands, which may evaluate to true or false, not operator needs one operand evaluating to true or false. Boolean and operator returns true if both operands return true.