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.

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

Ans: The plus equal operator (+=) appends to strings and creates a new string while not changing the value of the original string.

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

Ans: specifying the string name followed by a number in square brackets ( [] ).

Q4. What is the relationship between indexing and slicing?

Ans: They help you access specific elements in a sequence, such as a list, tuple or string.

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

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.

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

Ans: Python does not have a character data type.

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: Using + operator.

Using join() method.

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?

Ans: find()

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

Ans: Logical operators and Comparison operators.