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

yes

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

Yes.

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

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?

“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?

String, string[begin: end: step]

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

Python does not have a character data type, a single character is simply a string with a length of 1. Square brackets can be used to access elements of the string.

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

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?

True , False , not , and , and or are the only built-in Python Boolean operators.