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

Yes . String is immutable datatype . Slicing and indexing is possible in string but not assigning

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

NO += operator to concatenate has nothing to do with string’s immutability . Immutability means assigning a value to any indexed character cannot be done.

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

**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?

In both of the above cases , the data type will be string

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

String is more of a general datatype. There is nothing like char in python. In python string with length one is char which is also called string only with length 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..

“+” and “+=”

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?

Before using the index method , if we check the target string with in or not in , it will help find if the substring exists in the target string and then the operation of indexing will happen .

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

Some of the operators are isspace, isalpha, isnumeric etc