**Assignment\_12**

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

**Ans: Strings are made immutable so that programmers cannot alter the contents of object.**

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

**Ans: : += does not concatenate the strings**.  **Strings are made immutable so that programmers cannot alter the contents of object.**

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

**Ans: String Indexing is a method to index a character. Individual characters in a string can be accessed by specifying the string name followed by a number in square brackets ([])**

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?

**Ans: There is no data type for indexed character.**

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

**Ans: Strings are array of bytes representing Unicode characters. However, 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: 1) “+”**

**2) “concat()”**

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: The indexOf () method return the position of the first occurrence of specified character(s) in a string.**

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

**Ans: and, or and not produce Boolean results**