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

**ANS:-**

**Yes. It is beacause of string immutability that means we can not change the characters in string by assigning new values to indexes.**

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

**ANS:-**

**No. When we concatenate string, we assign new value to the same string variable. We are not altering string based on the indexes, so it won't violate Python's string immutability**

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

**ANS:-**

**There are two methods for this, find() and index() both will return the lowest index at which char is found.**

**Q4. What is the relationship between indexing and slicing?**

**ANS:-**

**Indexing is a way to access an element of a string by its position or index. Slicing is accessing only part od a string and not a full string. String slicing is done by using index.**

**s = 'ineuron'**

**print(s[1:4])**

**print(s[-1:2:-1])**

**neu**

**noru**

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

**ANS:-**

**for both cases, it is 'str' type.**

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

**ANS:-**

**In Python, There is no char data type, even a single character enclosed in doubble quotes is considered as str.**

**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:-**

**"+" will concatenate two strings and "\*" will repeat the string number of times mentioned. We can also use join() method for getting a larger string.**

**print("Hello!" + 'How are you?')**

**print("Rain" \* 4)**

**l = ['How', 'are', 'you?']**

**s = ' '.join(l)**

**print(s)**

**Hello!How are you?**

**RainRainRainRain**

**How are you?**

**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:-**

**Normally, when we use index() method it will return the starting index of a substring, if substring is present in a string. But the major concern is if a substring is not present in a string, index() will return a ValueError. To avoid this, we can first check the target string with in or not in before using the index method to find a substring.**

**s = 'abcdefghijk'**

**s1 = 'xyz'**

**if s1 in s:**

**print(s.index(s1))**

**s.index(s1)**

**--------------------------------------------------------------**

**ValueError Traceback (most recent call last)**

**<ipython-input-3-e04554b5c0f8> in <module>**

**6 print(s.index(s1))**

**7**

**----> 8 s.index(s1)**

**ValueError: substring not found**

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

**ANS:-**

**Operators:**

* **Comparison operators : (== ,!= ,< ,> ,<=, >=)**
* **Logical Operators : (and, or, not)**
* **Identity Operators : (is , is not)**
* **Membership Operators : (in, not in)**

**String methods:**

**1. endswith() Returns true if the string ends with the specified value**

**2. isalnum() Returns True if all characters in the string are alphanumeric**

**3. isalpha() Returns True if all characters in the string are in the alphabet**

**4. isdecimal() Returns True if all characters in the string are decimals**

**5. isdigit() Returns True if all characters in the string are digits**

**6. isidentifier() Returns True if the string is an identifier**

**7. islower() Returns True if all characters in the string are lower case**

**8. isnumeric() Returns True if all characters in the string are numeric**

**9. isprintable() Returns True if all characters in the string are printable**

**10. isspace() Returns True if all characters in the string are whitespaces**

**11. istitle() Returns True if the string follows the rules of a title**

**12. isupper() Returns True if all characters in the string are upper case**