Python Strings & Indexing — Questions & Answers (Set 12)

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

Yes. Strings are immutable. Attempting s[i] = 'x' raises TypeError: 'str' object does not support item assignment.

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

No. s += t creates a new string object and rebinds s to it. The old string is discarded. CPython may optimize under the hood, but immutability remains intact.

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

Two main directions:
- Positive (0-based): s[i] from the left.
- Negative: s[-i] from the right (-1 is the last char).

A one-character slice s[i:i+1] is another way but technically slicing, not direct indexing.

Q4. What is the relationship between indexing and slicing?

- Indexing s[i] returns a single character (length-1 str).
- Slicing s[i:j:k] returns a substring (str) for the half-open interval [i, j) with optional step k.

Indexing can be emulated by a slice of length 1 (s[i] == s[i:i+1]).

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

- Indexed character: str of length 1.
- Slice result: str (possibly empty).

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

There is only one text type: str. A 'character' is simply a str of 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.

- Operators: + (concatenate), += (rebind to concatenation), * (repeat).
- Method: str.join(iterable) is preferred for joining many pieces.

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?

.index() raises ValueError if the substring is not found. Checking with in/not in avoids exceptions and allows safe branching. Alternatively, .find() can be used, which returns -1 instead of raising.

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

- Operators: ==, !=, <, <=, >, >= (lexicographic comparisons), in, not in.
- Methods: startswith(), endswith(), isalpha(), isdigit(), isalnum(), isdecimal(), isnumeric(), isspace(), islower(), isupper(), istitle(), isidentifier().