

Python String Data Type - Notes

1. Variable Creation

- **Definition:** A string in Python is a sequence of characters enclosed in quotes.
- **Example:**

```
my_string = "Hello, World!"
```

- `my_string` is a variable that stores the string `"Hello, World!"`.

2. Memory Allocation After Variable Declaration

- **Immutable Nature:** Strings in Python are immutable, meaning once a string is created, its value cannot be changed.
- **Memory Allocation:**
 - When a string is assigned to a variable, Python allocates memory to store the characters.
 - If another variable is assigned the same string, Python points both variables to the same memory location to save space.
- **Example:**

```
str1 = "Hello"  
str2 = "Hello"
```

- Both `str1` and `str2` point to the same memory location.

3. Accessing Strings Using Index

- **Indexing:** Strings are indexed with integers, starting from `0` for the first character.
- **Example:**

```
my_string = "Hello, World!"  
first_char = my_string[0] # 'H'
```

- `my_string[0]` accesses the first character, `H`.

4. Modifying Strings Using Index

- **Immutability:** Since strings are immutable, you cannot modify a string directly using an index.
- **Workaround:**
 - To "modify" a string, you need to create a new string.
 - **Example:**

```
my_string = "Hello"  
new_string = 'h' + my_string[1:] # 'hello'
```

- Here, a new string is created with the first character changed.

5. Slicing in Strings

- **Slicing:** You can extract a substring by specifying a range of indices.
- **Syntax:** `string[start:end]` (end is exclusive)
- **Example:**

```
my_string = "Hello, World!"
substring = my_string[7:12] # 'World'
```

- `my_string[7:12]` extracts the substring `"World"`.

6. Negative Index in Strings

- **Negative Indexing:** Python allows indexing from the end of the string using negative indices.
- **Example:**

```
my_string = "Hello, World!"
last_char = my_string[-1] # '!'
```

- `my_string[-1]` accesses the last character, `!`.

- **Negative Slicing:**

- **Example:**

```
substring = my_string[-6:-1] # 'World'
```

- `my_string[-6:-1]` extracts `"World"` by counting backward.

Interview Questions

1. Is a string immutable or mutable, and why?

- Strings in Python are **immutable**. Once a string is created, its contents cannot be changed. This immutability allows Python to optimize memory usage by sharing the same memory location for identical strings.

2. How does memory allocation happen when declaring a string?

- When a string is declared, Python allocates memory to store the string's characters. If another string variable is assigned the same value, Python points to the same memory location to optimize memory usage.

3. What is slicing in a string?

- Slicing is the process of extracting a portion of a string by specifying a range of indices. The syntax is `string[start:end]`, where `start` is inclusive and `end` is exclusive.

4. Is negative indexing supported in Python strings?

- Yes, negative indexing is supported. It allows you to access characters from the end of the string. For example, `string[-1]` gives the last character.

5. How do you access the last element of a string?

- You can access the last element of a string using negative indexing:
`string[-1]` .

Exercise Questions

1. Create a string variable and access the first and last characters using indexing.
2. Given a string `s = "PythonProgramming"` , extract the substring `"Python"` using slicing.
3. Modify a string to replace the first character with another character by creating a new string.
4. Reverse a string using slicing.
5. Access the third character from the end of the string using negative indexing.