

# ASSIGNMENT-2

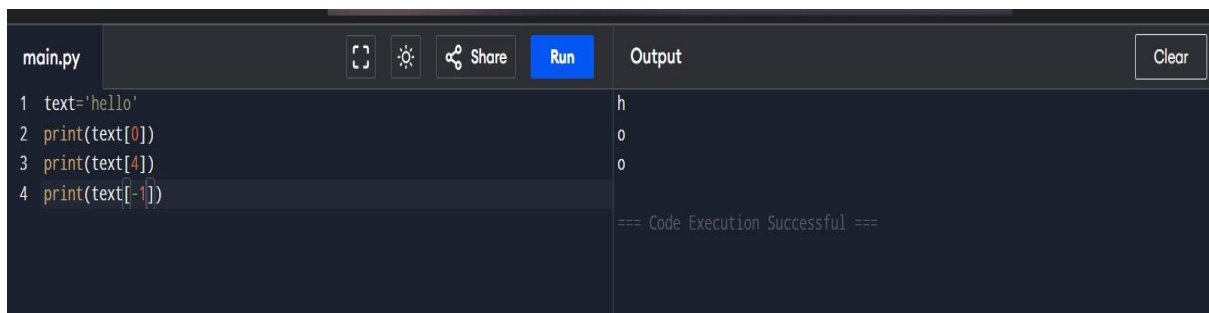
## Python string indexing & Slicing practice

1. In Python, a string is a sequence of characters enclosed in quotes. It's used to represent textual data.

2. Indexing in a string means accessing individual characters using their position (index) in the string

- Python uses zero-based indexing, so the first character is at position 0, the second at 1, and so on.
- You can also use negative indexing to count from the end of the string.

3.



The screenshot shows a Python IDE with a file named 'main.py'. The code in the editor is:

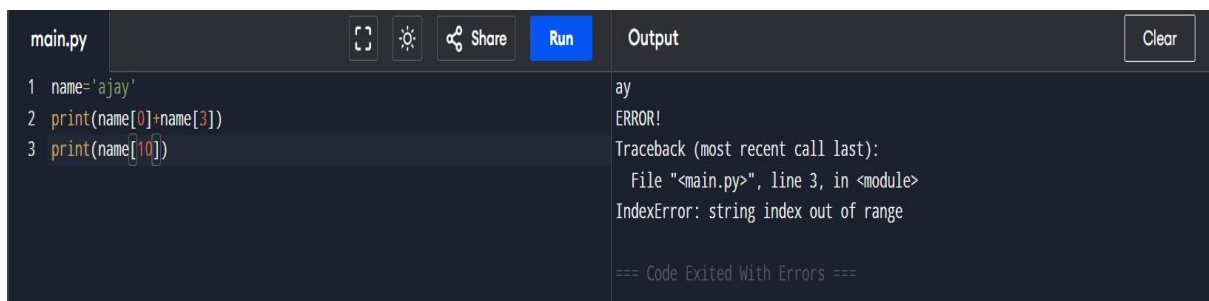
```
1 text='hello'
2 print(text[0])
3 print(text[4])
4 print(text[-1])
```

The 'Output' pane on the right shows the results of the execution:

```
h
o
o
```

Below the output, it says '=== Code Execution Successful ==='. The IDE interface includes icons for file operations, a 'Share' button, and a 'Run' button.

4.



The screenshot shows a Python IDE with a file named 'main.py'. The code in the editor is:

```
1 name='ajay'
2 print(name[0]+name[3])
3 print(name[10])
```

The 'Output' pane on the right shows the results of the execution:

```
ay
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 3, in <module>
IndexError: string index out of range
```

Below the error message, it says '=== Code Exited With Errors ==='. The IDE interface includes icons for file operations, a 'Share' button, and a 'Run' button.

5.

```
main.py  [ ] [ ] [ ] Share Run Output Clear
```

```
1 s='python'
```

```
2 print(s[0:2])
```

```
3 print(s[5:1])
```

```
py
```

```
=== Code Execution Successful ===
```

6.

```
main.py  [ ] [ ] [ ] Share Run Output Clear
1 s='python'
2 print(s[2:-1:2])
3
```

to

=== Code Execution Successful ===

7.

```
main.py [ ] [ ] [ ] Share Run Output Clear
```

```
1 str='elephant'
```

```
2 print(str[5:8])
```

```
ant
```

```
=== Code Execution Successful ===
```

8.

```
main.py  [ ] [ ] [ ] Share Run Output Clear
1 str='science'
2 print(str[2:5])
```

ien

=== Code Execution Successful ===

9. Both `s[2:5]` and `s[2:5:1]` are string slicing operations the difference lies in the third value, called the step.

`s[start : stop]`  $\rightarrow$  uses default `step = 1`

`s[start : stop :step]` → gives full control over how to slice