1. What is Indexing?

Each character in a string has a position number.

This is called its index.

Python starts counting from 0 (not 1)!

Example:

```
text = "Python"
```

Character	P	y	t	h	o	n
Index	0	1	2	3	4	5

Accessing a character:

```
text = "Python"
print(text[0])   # P
print(text[3])   # h
```

2. Negative Indexing

Python also supports negative indexes.

Character	P	y	t	h	o	n
Index	0	1	2	3	4	5
Negative	-6	-5	-4	-3	-2	-1

```
text = "Python"
print(text[-1])  # n (last letter)
print(text[-3])  # h
```

3. What is Slicing?

Slicing means cutting out a part of the string.

Syntax:

```
String[start:end]
```

It includes the start, but excludes the end.

Example:

```
text = "Python"
print(text[0:4])   # Pyth (index 0 to 3)
print(text[2:6])   # thon
```

4. Slicing Tricks

Slice from beginning to index:

```
print(text[:4])   # Pyth
```

Slice from index to end:

```
print(text[2:])   # thon
```

Whole string:

```
print(text[:])    # Python
```

5. Using len() with Indexing

```
word = "Hello"
length = len(word)
print("Last character:", word[length - 1])  # o
```

6. String Reversal

```
text = "Python"
print(text[::-1])  # nohtyP (reversed)
```

Practice Questions

Print the first letter of your name

Print the last letter of "Python" using negative indexing

Slice and print only "tho" from "Python"

Print all letters of "Welcome" except the last one

Reverse the string "Coding"

Get the middle 3 letters of "Elephant"

Print each letter of "Class" one by one using a for loop

Ask the user for a word and print only its first 3 letters

1. Looping Through a String

We can use a `for` **loop** to go through each character in a string — **one by one**.

Example:

```
word = "hello"

for letter in word:
    print(letter)
```

Output:

```
h
e
l
l
o
```

Why is this useful?

You can:

- Count letters
- Check vowels
- Create custom output
- Reverse a string
- Clean or analyze text

Example: Count how many times a appears

```
text = "banana"
count = 0

for letter in text:
    if letter == 'a':
        count += 1

print("a appears", count, "times.")
```

Output:

```
a appears 3 times.
```

---

2. String Formatting – Making Output Look Better

When we want to insert values inside a sentence, we use **formatting**.

Option 1: `f-strings` (Python 3.6+)

```
name = "Amit"
age = 18

print(f"My name is {name} and I am {age} years old.")
```

Output:

```
My name is Amit and I am 18 years old.
```

Real-Life Use Case

Let's say you want to send personalized messages:

```
user = "Anjali"
course = "Python"

print(f"Hello {user}, welcome to the {course} course!")
```

Output:

```
Hello Anjali, welcome to the Python course!
```

Practice Questions

1. Loop through your name and print each letter
2. Ask the user for a word and count how many `e` are in it
3. Ask name and hobby, and print using f-string:
   `"Hello <name>, your hobby is <hobby>"`
4. Write a loop to print only vowels in `"education"`
5. Create a loop that reverses `"Python"` and prints each letter