Priyanshu Kumar

*args and **kwags in Python

@priyanshukumar69

02

What are *args and **kwargs?

- In Python, *args and **kwargs are special syntax used in function definitions.
- *args allows passing a variable number of non-keyword arguments to a function.
- **kwargs allows passing a variable number of keyword arguments to a function.

03

Working with *args

- *args collects all non-keyword arguments passed to a function.
- It allows flexibility in the number of arguments.
- Access the arguments within the function using the *args parameter.

```python
def my_function(*args):
    for arg in args:
        print(arg)

my_function(1, 2, 3)
```

04

Working with **kwargs

- **kwargs collects all keyword arguments passed to a function.
- It allows flexibility in the number of keyword arguments.
- Access the arguments within the function using the **kwargs parameter.

```python
def my_function(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}: {value}")

my_function(name="John", age=25, city="New York")
```

Combining *args and **kwargs

05

- You can combine *args and **kwargs in a function definition.
- *args collects non-keyword arguments, and **kwargs collects keyword arguments.
- This provides flexibility in accepting different types of arguments.

```python
def my_function(*args, **kwargs):
    for arg in args:
        print(arg)
    for key, value in kwargs.items():
        print(f"{key}: {value}")

my_function(1, 2, name="John", age=25)
```

DONT FORGET TO LIKE, SHARE AND SAVE IF YOU LIKE THIS POST

@priyanshukumar69