In Python, **variable keyword arguments** are used to allow a function to accept an arbitrary number of keyword arguments. This is done using the \*\*kwargs syntax.

**How \*\*kwargs Works**

The \*\*kwargs parameter in a function gathers all keyword arguments into a dictionary, where the keys are the argument names and the values are the corresponding argument values.

**Example Usage**

def greet(\*\*kwargs):

for key, value in kwargs.items():

print(f"{key}: {value}")

greet(name="Alice", age=25, city="New York")

**Output**

name: Alice

age: 25

city: New York

**Use Cases**

**1. Flexible Function Arguments**

Using \*\*kwargs allows functions to accept any number of keyword arguments.

def user\_profile(\*\*details):

return details

print(user\_profile(name="John", age=30, profession="Engineer"))

**Output:**

{'name': 'John', 'age': 30, 'profession': 'Engineer'}

**2. Default Arguments with Overrides**

It enables functions to define default values and override them if necessary.

def settings(\*\*options):

default\_settings = {

"theme": "light",

"language": "English",

"notifications": True

}

default\_settings.update(options) # Update defaults with user options

return default\_settings

print(settings(language="Spanish", notifications=False))

**Output:**

{'theme': 'light', 'language': 'Spanish', 'notifications': False}

**3. Passing \*\*kwargs to Another Function**

You can pass \*\*kwargs to another function that accepts keyword arguments.

def print\_info(name, age):

print(f"Name: {name}, Age: {age}")

data = {"name": "Emma", "age": 22}

print\_info(\*\*data) # Unpacking dictionary

**Output:**

Name: Emma, Age: 22

**4. Combining \*args and \*\*kwargs**

You can use both \*args (for positional arguments) and \*\*kwargs (for keyword arguments) in the same function.

def mix\_args(\*args, \*\*kwargs):

print("Positional arguments:", args)

print("Keyword arguments:", kwargs)

mix\_args(1, 2, 3, name="Alice", age=25)

**Output:**

Positional arguments: (1, 2, 3)

Keyword arguments: {'name': 'Alice', 'age': 25}

**Key Points**

1. \*\*kwargs collects all extra keyword arguments into a dictionary.
2. It makes functions more flexible by allowing additional parameters.
3. Can be used alongside \*args for both positional and keyword arguments.
4. Allows easy passing and modification of dictionary-like configurations.