Function Arguments

In Python, user-defined functions can take four different types of arguments.
The argument types and their meanings, however, are pre-defined and can't be changed.
But a developer can, instead, follow these pre-defined rules to make their own custom functions.

The following are the four types of arguments and their rules.

Default arguments :
Python has a different way of representing syntax and default values for function arguments.
Default values indicate that the function argument will take that value if no argument value is passed during function call.
The default value is assigned by using assignment (=) operator.

Required arguments / Position Argument :
Required arguments are the mandatory arguments of a function.
These argument values must be passed in correct number and order during function call.

Keyword arguments :
Keyword arguments are relevant for Python function calls.
The keywords are mentioned during the function call along with their corresponding values.
These keywords are mapped with the function arguments so the function can easily identify the corresponding values even if the order is not maintained during the function call.

Variable number of arguments:
This is very useful when we do not know the exact number of arguments that will be passed to a function.
Or we can have a design where any number of arguments can be passed based on the requirement.

Consider below application which demonstrate concept of Function Arguments

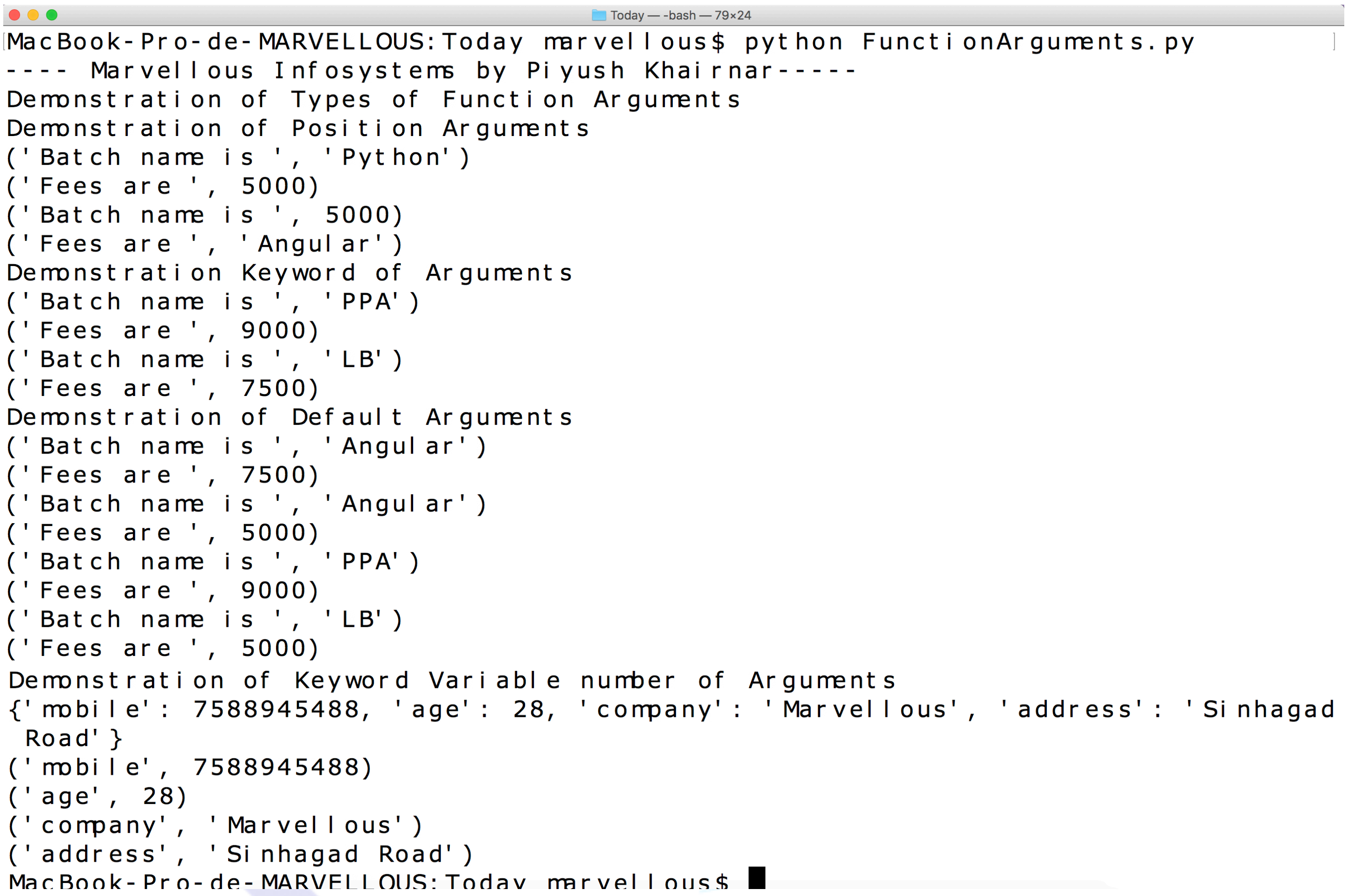
```
print("---- Marvellous Infosystems by Piyush Khairnar-----")

print("Demonstration of Types of Function Arguments")

# Position arguments

def Batches1(name,fees):
    print("Batch name is ", name)
    print("Fees are ", fees)

print("Demonstration of Position Arguments")

Batches1('Python', 5000)
```

```
Batches1(5000,'Angular')
# Keyword Arguments

def Batches2(name,fees):
    print("Batch name is ", name)
    print("Fees are ", fees)

print("Demonstration Keyword of Arguments")

Batches2(fees=9000, name='PPA')
Batches2(name='LB',fees=7500)

# Default Arguments

def Batches3(name,fees = 5000):
    print("Batch name is ", name)
    print("Fees are ", fees)

print("Demonstration of Default Arguments")

Batches3('Angular',7500)
Batches3('Angular')
Batches3(fees=9000, name='PPA')
Batches3(name='LB')

# Variable number of arguments

def Add(*no):
    ans = 0
    for i in no:
        ans = ans + i

    return ans

print("Demonstration of Variable number of Arguments")

ret = Add(10,20,30)
print("Addition is ",ret)

ret = Add(10,20,30,40,50,60)
print("Addition is ",ret)

ret = Add(10,20)
print("Addition is ",ret)

# Keyword Variable number of arguments

def StudentInfo(**other):
    print(other)
    for i,j in other.items():
        print(i,j)
```

print("Demonstration of Keyword Variable number of Arguments")

StudentInfo(age=28, address="Sinhagad Road", mobile=7588945488, company="Marvellous")

Output of Above application



```
MacBook-Pro-de-MARVELLOUS:Today marvellous$ python FunctionArguments.py
---- Marvellous Infosystems by Piyush Khairnar-----
Demonstration of Types of Function Arguments
Demonstration of Position Arguments
('Batch name is ', 'Python')
('Fees are ', 5000)
('Batch name is ', 5000)
('Fees are ', 'Angular')
Demonstration Keyword of Arguments
('Batch name is ', 'PPA')
('Fees are ', 9000)
('Batch name is ', 'LB')
('Fees are ', 7500)
Demonstration of Default Arguments
('Batch name is ', 'Angular')
('Fees are ', 7500)
('Batch name is ', 'Angular')
('Fees are ', 5000)
('Batch name is ', 'PPA')
('Fees are ', 9000)
('Batch name is ', 'LB')
('Fees are ', 5000)
Demonstration of Keyword Variable number of Arguments
{'mobile': 7588945488, 'age': 28, 'company': 'Marvellous', 'address': 'Sinhagad
 Road'}
('mobile', 7588945488)
('age', 28)
('company', 'Marvellous')
('address', 'Sinhagad Road')
MacBook-Pro-de-MARVELLOUS:Today marvellous$
```