Python Arbitrary Arguments - *args and **kwargs

***args** allow you to pass multiple non-key **arguments**.

****kwargs** allow you to pass multiple **keyword arguments**

The ***** is called the "**unpacking operator**"

When you invoke a function that has args or kwargs as parameters, you will pack all of those arguments into a **tuple** if its **args**, or a **dictionary** if the parameter is **kwargs**.

```
def add(x, y):
    return x + y

print(add(1, 2))
```

The output will be 3, but this is very limiting – what if I need to add 3 numbers? If we pass 3 arguments, the function will fail. This add function only takes two positional arguments.

If we replace the parameters with ***args**, the arguments we pass into the function will be packed into a **tuple**.

```
def add(*args):
    total = 0
    for arg in args:
        total += arg
    return total

print(add(1, 2, 3, 4))
```

In the above example, there is no limit to the number of arguments that may be passed to the function.

Another Example

```
def display_name(*args):
   print(f"Hello", end=" ")
   for arg in args:
       print(arg, end=" ")

display_name("Dr.", "Spongebob", "Harold", "Squarepants","II")
```

kwargs

Kwargs use two unpacking operators followed by the work kwargs, ****kwargs**

Kwargs will be packed into a **dictionary**.

```
def print_address(**kwargs):
    for value in kwargs.values():
        print(value, end=" ")

print_address(street="9 Byron Street",
              city="Ringwood",
              state="VIC",
              postcode="3134")
```

output:

```
9 Byron Street Ringwood VIC 3134
```

We could use the key, value pair feature of a dictionary to alter the layout to include the key.

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

print_address(street="9 Byron Street",
              city="Ringwood",
              state="VIC",
              postcode="3134")
```

output:

```
street: 9 Byron Street
city: Ringwood
state: VIC
postcode: 3134
```

Exercise: Print a Shipping Label that uses *args and **kwargs to display a name and address. The function should check whether there is a PO Box or an Apartment number and only display these values if they exist.

```
Dr. Spongebob Squarepants
1/9 Byron Street
Ringwood
VIC, 3134
```

Solution:

```
def shipping_label(*args, **kwargs):
    for arg in args:
        print(arg, end=" ")
    print()

    if "apt" in kwargs:
        print(f"{kwargs.get('apt')}/{kwargs.get('street')}")
    elif "pobox" in kwargs:
        print(f"{kwargs.get('street')}")
        print(f"{kwargs.get('pobox')}")
    else:
        print(f"{kwargs.get('street')}")

    print(f"{kwargs.get('city')}")
    print(f"{kwargs.get('state')} {kwargs.get('postcode')}")

shipping_label("Dr.", "Spongebob", "Squarepants",
               apt="1",
               street="9 Byron Street",
               city="Ringwood",
               state="VIC",
               postcode="3134")
```