A picture containing graphical user interface

Description automatically generated

Positional arguments are shown below:

def person(name, age):

      print(name)

      print(age)

person('uday', 31)

what will happen if we swap the values for person(31, ‘uday’) – check it out

* What if we want to do something like this in function print(age-5) and this will give the error as we had passed uday in the first place and we can’t substract 5 from it.
* Now if we don’t know the sequence we can call function as person(age=31, name =’uday’) and this will work ***# here we are using keyword arguments***

If we don’t know the value of age or we want to give a default value for age we can pass the default value as shown below.

def person(name, age=18):

      print(name)

      print(age)

person('uday')

3) variable parameter:

When you want to allow multiple values to be added, first parameter a and then the second argument \*b will allow to pass as many values as needed. Check the below program with star and without it.

Graphical user interface, text, application

Description automatically generated

The above program will give error and we see what’s happening with the help of below program

Graphical user interface, application

Description automatically generated

The above program should give a tuple for b

Q: now we will write a function to add the values which are part of sum function.

Q: just work with \*b in sum(\*b) to add multiple numbers.

def sum(\*b):

      c = 0

      for i in b:

            c = c + i

      print(c)

sum(5,5,1,2,3,4)

Now we will try keyworded variable length arguments in python

Try below code and print it and observe the output. It will

**Graphical user interface, application, website

Description automatically generated**

Now if you want to

**Timeline

Description automatically generated with medium confidence**

In the below code it is I,j which is a key value pair(I,j) and we are using data.items() where items is a function and we are trying to print the key value pair(i,j)

def person(name, \*\*data):

      print(name)

      for i,j in data.items():

            print(i,j)

person('uday', age=31,city='Leeds', mobile=123456789)

**Graphical user interface

Description automatically generated with medium confidence**

Now observe the output for the above program and understand how it is working with \*\* and \* (know the differences)

def person(\*\*data):

      for i,j in data.items():

            print(i,j)

person(name='uday', age=31,city='Leeds', workmobile=123456789, homemobile=123123123)

**Global vs Local variables:**

Let’s know scope first

**Graphical user interface, application

Description automatically generated**

a = 10

def local():

      a = 15

      print(a, 'local variable')

local()

print(a, 'global variable')

Now in the below program we are saying that local a will be called as global which means it will be considered as global variable as we had used the keyword ***global***. Run the below program and check the output.

**Graphical user interface, text, application, chat or text message

Description automatically generated**

a = 10

def local():

      global a

      a = 15

      print(a, 'local variable')

local()

print(a, 'global variable')

4/4/2022

Check the below code by running it.

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

# Program make a simple calculator

# This function adds two numbers

def add(x, y):

    return x + y

# This function subtracts two numbers

def subtract(x, y):

    return x - y

# This function multiplies two numbers

def multiply(x, y):

    return x \* y

# This function divides two numbers

def divide(x, y):

    return x / y

print("Select operation.")

print("1.Add")

print("2.Subtract")

print("3.Multiply")

print("4.Divide")

while True:

    # take input from the user

    choice = input("Enter choice(1/2/3/4): ")

    # check if choice is one of the four options

    if choice in ('1', '2', '3', '4'):

        num1 = float(input("Enter first number: "))

        num2 = float(input("Enter second number: "))

        if choice == '1':

            print(num1, "+", num2, "=", add(num1, num2))

        elif choice == '2':

            print(num1, "-", num2, "=", subtract(num1, num2))

        elif choice == '3':

            print(num1, "\*", num2, "=", multiply(num1, num2))

        elif choice == '4':

            print(num1, "/", num2, "=", divide(num1, num2))

        # check if user wants another calculation

        # break the while loop if answer is no

        next\_calculation = input("Let's do next calculation? (yes/no): ")

        if next\_calculation == "no":

          break

    else:

        print("Invalid Input")