

args and kwargs

November 30, 2021

```
[55]: #Functions can hold regular arguments, *args and **kwargs
      #-altogether
      #*args argument collector: allows to pass any arguments to a
      #-function
      #like a list but not same
      #lets you assign as many arguments as u like
      def myStuff(*args):
          for item in args:
              print(item)
      myItems = ["Phone", "Keys", "Wallet", "Bag", "Shoes", "clothes"]
      myStuff(*myItems) # the * wildcard here is necessary as we're
      #calling items in a list
```

Phone
Keys
Wallet
Bag
Shoes
clothes

```
[56]: #if calling normal no * needed:
      myStuff("Phone", "Keys", "Wallet", "Bag", "Shoes")
      #Here one can use as many arguments as one wants
```

Phone
Keys
Wallet
Bag
Shoes

```
[59]: #if using normal parameters plus an args list, it can be done:
      def myStuff(param1, param2, *args): #Give space for required values,
                                          #then *args
          print(param1, param2,) #print them inside loop
          for item in args: #specify that u want every item printed
              print(item)
      myItems = ["Phone", " ", "Keys", " ", "Wallet", " ", "Bag", " ", "Shoes", " ", "clothes"] #this
      ↪ will be our *args
      print("First argument, ", "Second argument, ", *myItems)
```

```
#call function using arguments of choice
```

First argument, Second argument, Phone, Keys, Wallet, Bag, Shoes, clothes

```
[ ]: ***kwargs like a dictionary, holds keyword arguments  
#
```

```
[ ]:
```

```
[ ]:
```