### # Default arguments are only executed once when the function is created

import time

from datetime import datetime

def add\_emp(emp, emp\_list=[]):

    emp\_list.append(emp)

    print(emp\_list)

add\_emp('John Doe')

# ['John Doe']

add\_emp('Eric Mathew')

# ['John Doe', 'Eric Mathew']

add\_emp('Billy Johnson')

# ['John Doe', 'Eric Mathew', 'Billy Johnson']

# Solution

def add\_emp2(emp, emp\_list=None):

    if(emp\_list == None):

        emp\_list = []

    emp\_list.append(emp)

    print(emp\_list)

add\_emp2('John Doe')

# ['John Doe']

add\_emp2('Eric Mathew')

# ['Eric Mathew']

add\_emp2('Billy Johnson')

# ['Billy Johnson']

# Second Example

def show\_time(cur\_time=datetime.now()):

    print(cur\_time.strftime('%H:%M:%S'))

# running show\_time() three times doesn't increment

# seconds by one, because default argument is run only once  
show\_time()  # 17:09:53

time.sleep(1)

show\_time()  # 17:09:53

time.sleep(1)

show\_time()  # 17:09:53

time.sleep(1)

# solution

def show\_time2(cur\_time=None):

    if cur\_time == None:

        cur\_time = datetime.now()

    print(cur\_time.strftime('%H:%M:%S'))

# running show\_time() three times increment

# seconds by one, because there isn’t default arg nomore

show\_time2()  # 17:09:53

time.sleep(1)

show\_time2()  # 17:09:54

time.sleep(1)

show\_time2()  # 17:09:55

time.sleep(1)

### # iterators can be exhausted that means we can loop through and access their values one time and can’t access their values second time

# unlike list iterators can be exhausted

names = ['Peter Parker', 'Clark Kent', 'Wade Wilson', 'Bruce Wayne']

heroes = ['Spiderman', 'Superman', 'Deadpool', 'Batman']

# after first return iterators are exhausted and zip is an iterator as well  
identities = zip(names, heroes)

print(list(identities))

# this iterator won't print anything because iterator is already exhausted

for identity in identities:

    print(f"{identity[0]} is actually {identity[1]}.")

# the output :-

# [('Peter Parker', 'Spiderman'), ('Clark Kent', 'Superman'), ('Wade Wilson', 'Deadpool'), ('Bruce Wayne', 'Batman')]

# solution - is to cast zip iterator to a list and list can't be exhausted

identities = list(zip(names, heroes))

print(identities)

# this iterator won't print anything because iterator is already exhausted

for identity in identities:

    print(f"{identity[0]} is actually {identity[1]}.")

# [('Peter Parker', 'Spiderman'), ('Clark Kent', 'Superman'), ('Wade Wilson', 'Deadpool'), ('Bruce Wayne', 'Batman')]

# Peter Parker is actually Spiderman.

# Clark Kent is actually Superman.

# Wade Wilson is actually Deadpool.

# Bruce Wayne is actually Batman.