
```
import time

# Decorator - Measure the time taken by the function
def time_it(func_name):

  def wrapper(*args, **kwargs):
    print ("I am in wrapper")
    start = time.time()

    # Call the appropriate function
    f_result = func_name(*args, **kwargs)

    time.sleep(3)
    end = time.time()
    print (func_name.__name__ + " took " + str((end - start) * 1000) + "
milliseconds")

    return f_result

  return wrapper

@time_it
def calc_sqr(max_num):
  print ("I am in calc_sqr")
  sqr_val = [num ** 2 for num in range(1, max_num + 1)]
  return (sqr_val)

@time_it
def calc_cube(max_num):
  print ("I am in calc_cube")
  cube_val = [num ** 3 for num in range(1, max_num + 1)]
  return (cube_val)

# Call calc_sqr
print ("Square ->", calc_sqr(5))

print ("-----------------------------------------------------------------")

# Call calc_cube
print ("Cube ->", calc_cube(5))

print ("-----------------------------------------------------------------")

def displayDetails(*args, **kwargs):
  for arg in args:
    print ("Arg ->", arg)

  for key,value in kwargs.items():
    print (key, '->', value)
```


```
displayDetails("name", "location", "salary", name="davies",
location="bengaluru", salary="15000")

print ("----------------------------------------------------------------")
```