Python training - lab 5

Functions

```
def func():
   print('Hello World')

def func2(p):
   print('Hello ' + p)

def func3(p):
   return 'Hello ' + p

func()
# None, function doesn't return anything
v = func2('Dolly')

v = func3('Dolly') # Hello Dolly
```

Passing params by name

```
def media(a, b, c):
   return (a + b + c) / 3

media(5, 2, 3) # ok
media(b=5, c=2, a=3) # ok
media(c=2, a=3) # NOT ok
media(4, c=2, a=3) # NOT ok
media(4, c=2, b=3) # ok
```

Parameter default values

```
# ok; some default values and some not
def media(a, b=2, c=3):
   return (a + b + c) / 3

media(4, 5, 6) # ok
media(4, 5) # ok
media(4) # ok
media() # NOT ok

# not ok
def media(a=2, b, c=3):
   return (a + b + c) / 3
```

Variable number of parameters

```
# accepts variable number of positional arguments
def media(*args):
    print(args)
    print(type(args)

media() # ok
media(3, 'ab') # ok
media([34, 9], True, 'abcd') # ok

def media(*args):
    if len(args) == 0:
        return 0
    return sum(args) / len(args)
```

Variable number of parameters

```
# accepts variable number of keyword arguments
def func(**kwargs):
    print(kwargs)
    print(type(kwargs)

func() # ok
func(34, 'a') # NOT ok
func(b=34, d='a') # ok
```

Variable number of parameters

```
# accepts variable number of
# positional arguments and keyword arguments
def func(*args, **kwargs):
   print(args)
   print(kwargs)


func() # ok
func(34, 'a', b=123) # ok
func(b=34, d='a') # ok
func(b=34, 55, d='a') # NOT ok
```

Variable number of parameters

```
# a and b parameters are mandatory
# the others are optional
def func(a, b, *args, **kwargs):
   print(args)
   print(kwargs)


func(5) # NOT ok, missing value for b
func(34, 'QWE', b=123) # NOT ok
func(34, 'QWE', 22, 33, b=123) # ok
func(4, 5, a=7) # NOT ok, multiple value for a
```

Variable number of parameters

```
# a and b parameters are mandatory
# c is a mandatory keyword argument
def func(a, b, *args, c=8, **kwargs):
    print(args)
    print(kwargs)


func(2, 3, 'a', True, c=9, ab=12) // OK

# b must be passed as a keyword only
# every param that follows * must be passed as kw
def func(a, *, b):

func(3, 4) // not OK
func(3, b=4) // OK
func(b=3, a=4) // OK

def func(a=7, *, b): // definition OK
```