Cheatsheets / **Learn Intermediate Python 3**

codecademy

Function Arguments

Mutable Default Arguments

Python's default arguments are evaluated only once when the function is defined, not each time the function is called. This means that if a mutable default argument is used and is mutated, it is mutated for all future calls to the function as well. This leads to buggy behaviour as the programmer expects the default value of that argument in each function call.

```
# Here, an empty list is used as a
default argument of the function.
def append(number, number_list=[]):
  number_list.append(number)
  print(number_list)
  return number_list

# Below are 3 calls to the `append`
function and their expected and actual
outputs:
append(5) # expecting: [5], actual: [5]
append(7) # expecting: [7], actual: [5,
7]
append(2) # expecting: [2], actual: [5,
7, 2]
```

Function Keyword Arguments

Python functions can be defined with named arguments which may have default values provided. When function arguments are passed using their names, they are referred to as keyword arguments. The use of keyword arguments when calling a function allows the arguments to be passed in any order — *not* just the order that they were defined in the function. If the function is invoked without a value for a specific argument, the default value will be used.

```
def findvolume(length=1, width=1,
depth=1):
  print("Length = " + str(length))
  print("Width = " + str(width))
  print("Depth = " + str(depth))
  return length * width * depth;

findvolume(1, 2, 3)
findvolume(length=5, depth=2, width=4)
findvolume(2, depth=3, width=4)
```

codecademy

Default argument is fallback value

In Python, a default parameter is defined with a fallback value as a default argument. Such parameters are optional during a function call. If no argument is provided, the default value is used, and if an argument is provided, it will overwrite the default value.

```
def greet(name, msg="How do you do?"):
  print("Hello ", name + ', ' + msg)

greet("Ankit")
greet("Ankit", "How do you do?")

"""
this code will print the following for
both the calls -
`Hello  Ankit, How do you do?`
"""
```

Print Share