**Section 6: Methods and functions**

**11.02.**

**41. Methods and the Python Documentation**

help(object.method) can be used to get info about the method, but method w/0 ()

Also, can be used SHIFT+TAB after writing object\_name.

**46. Tuple Unpacking with Python Functions**

We can return multiple values in Python and use the tuple unpacking for the returned results

**14.02.**

**49. \*args and \*\*kwargs in Python**

args – non keyword arguments

\*args will embed a tuple of arbitrary arguments

kwargs – key word arguments

\*\*kwargs will embed a dictionary of key-value pairs

Args and kwargs naming are a convention of PEP8

Both args and kwargs will let us use as many arguments as wanted

If we set in a function definition the args first and then kwargs, we cannot send position arguments after key word arguments

\*kwargs in a function give us the keys

We cannot use \*args after \*\*kwargs in a function

After \*args we can use keyword arguments before or after \*\*kwargs

[**https://www.programiz.com/python-programming/function**](https://www.programiz.com/python-programming/function)

! In a function we can access the doctring in the following way: function\_name.\_\_doc\_\_

[**https://levelup.gitconnected.com/5-types-of-arguments-in-python-function-definition-e0e2a2cafd29**](https://levelup.gitconnected.com/5-types-of-arguments-in-python-function-definition-e0e2a2cafd29)

A function has formal parameters that are in the definition of the function

A function has actual parameters or arguments that are passed when the function is called

In Python there are multiple types of arguments:

Default arguments

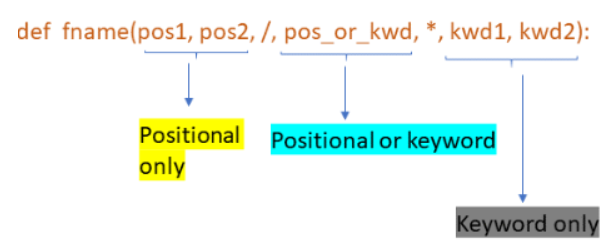
Positional arguments

Keyword arguments

Arbitrary positional arguments

Arbitrary keyword arguments

There is a separation of formal parameters in the function definition to increase readability:



In this way the position and keyword only arguments are enforced

**15.02.**

**50. Function Practice Exercises – Overview**

.join(iterable) returns a string in which between every 2 elements a specific string is positioned

For example ‘#’.join([‘a’, ‘b’, ‘c’]) will give ‘a#b#c’