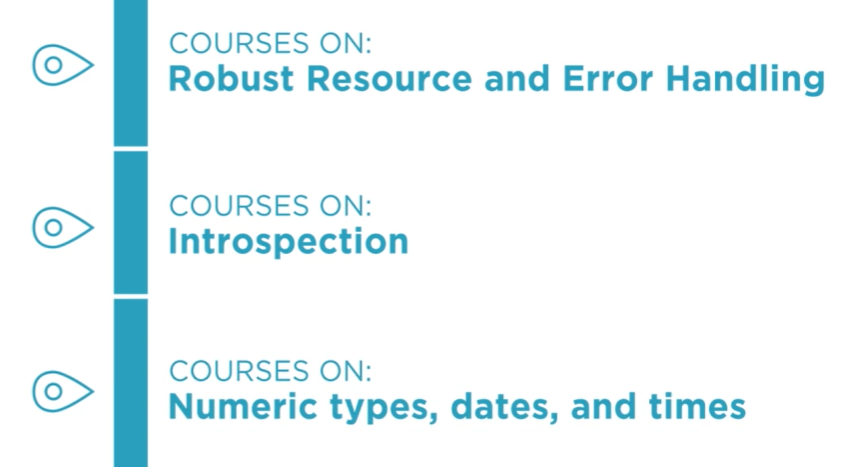
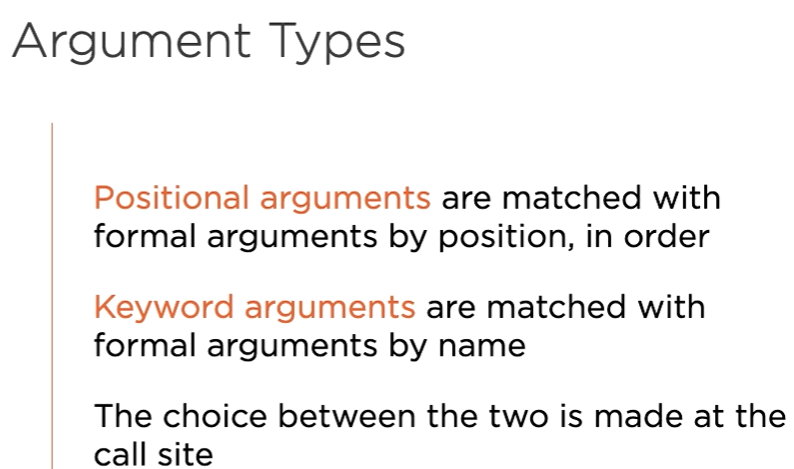
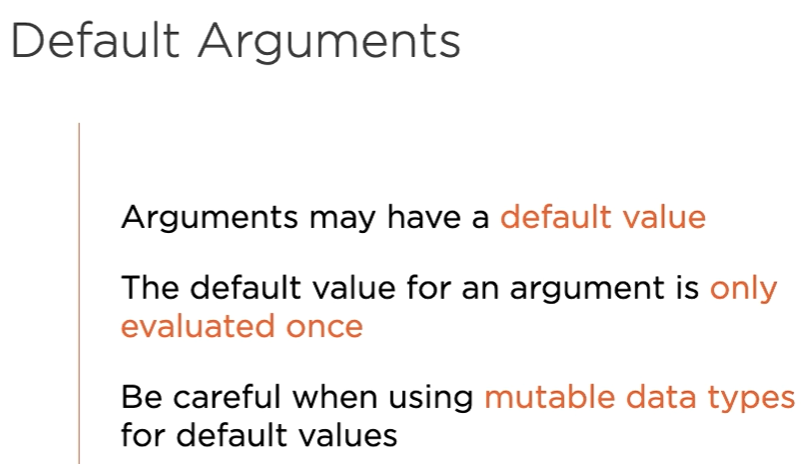
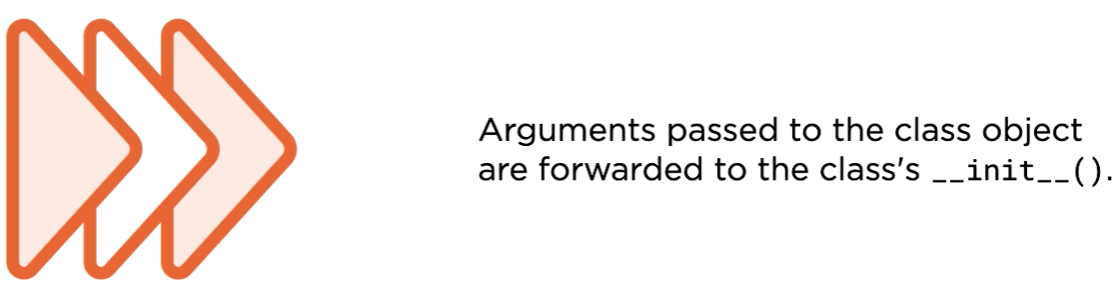
**Functions and Functional Programming**

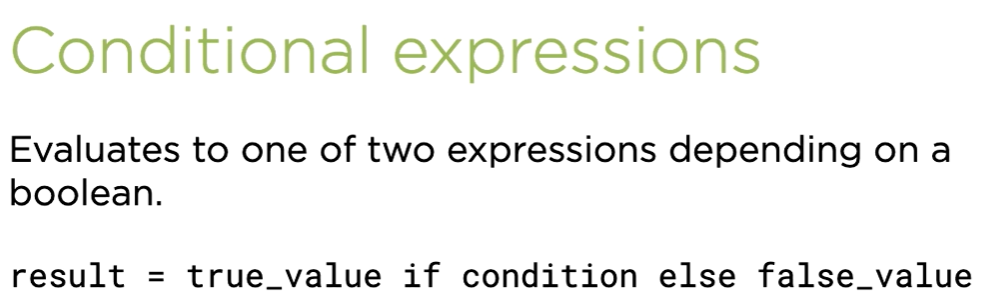
****

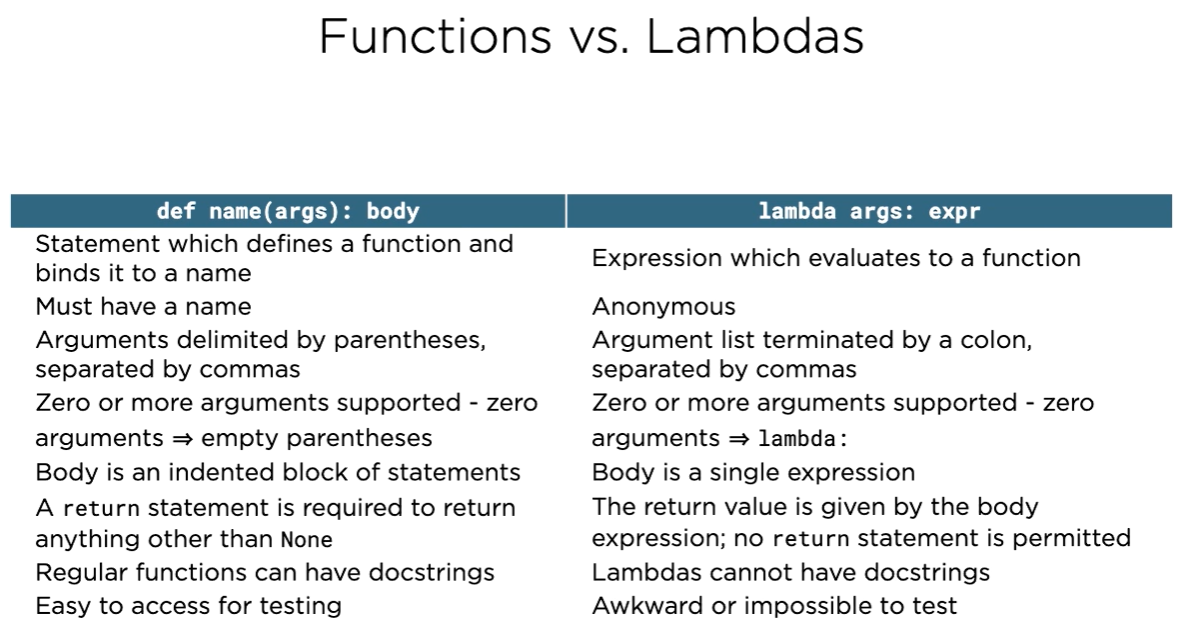
**Function and callables**

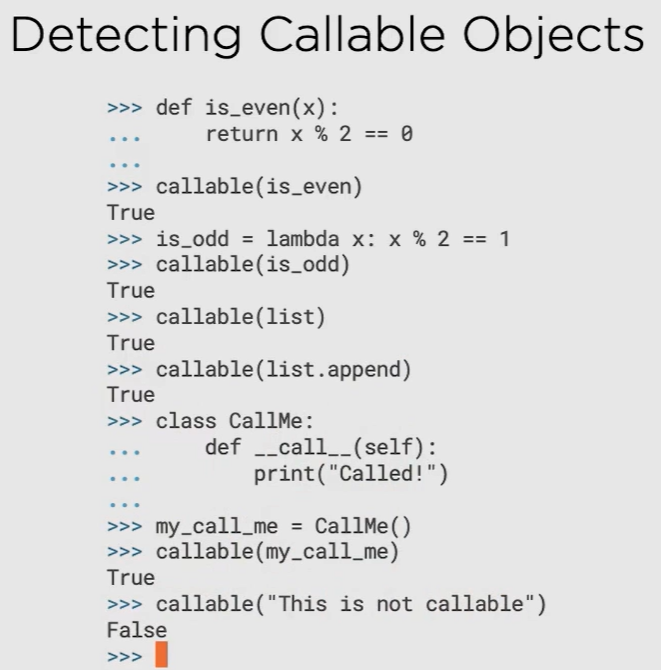
****

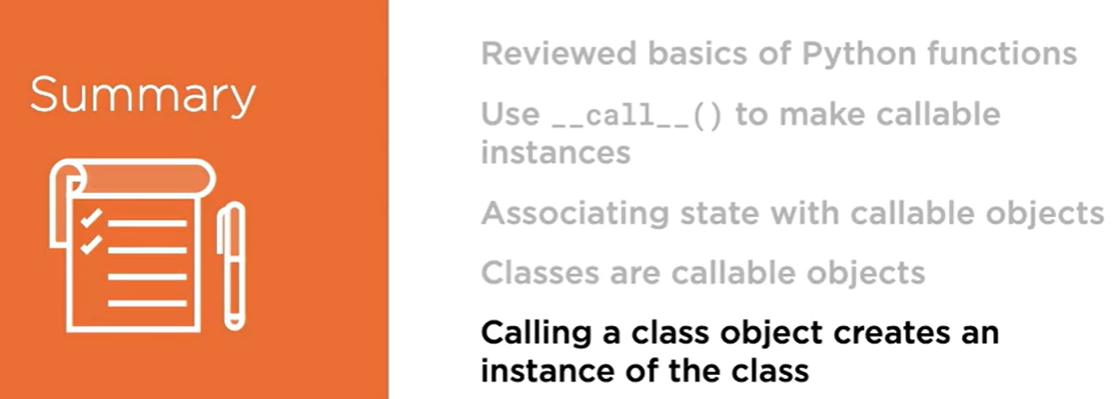
****

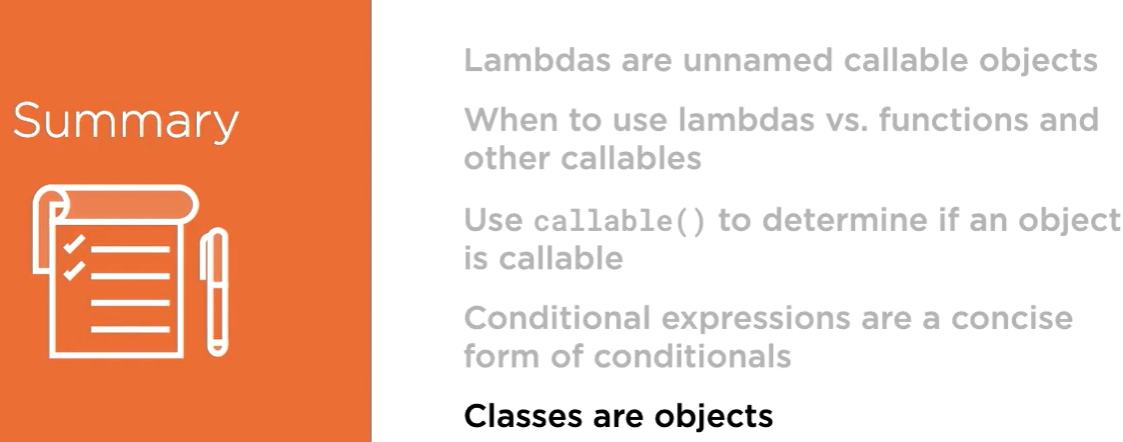
****

****

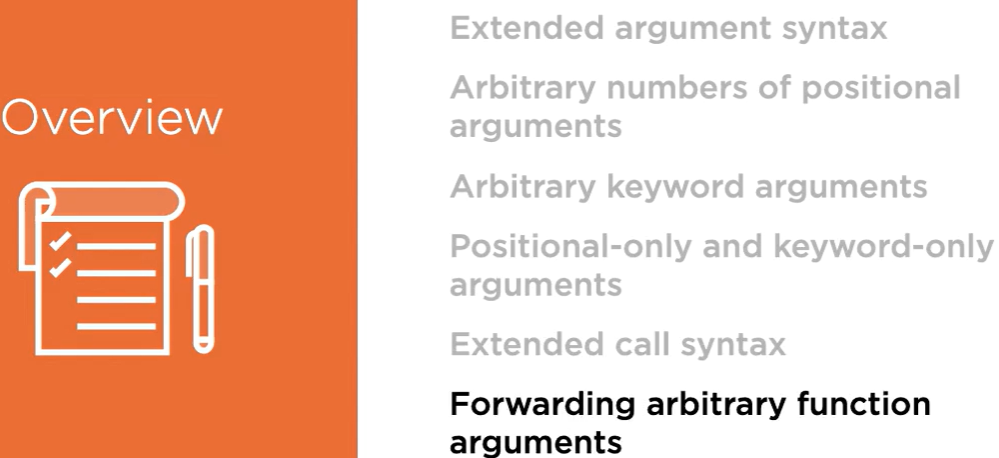
****

****

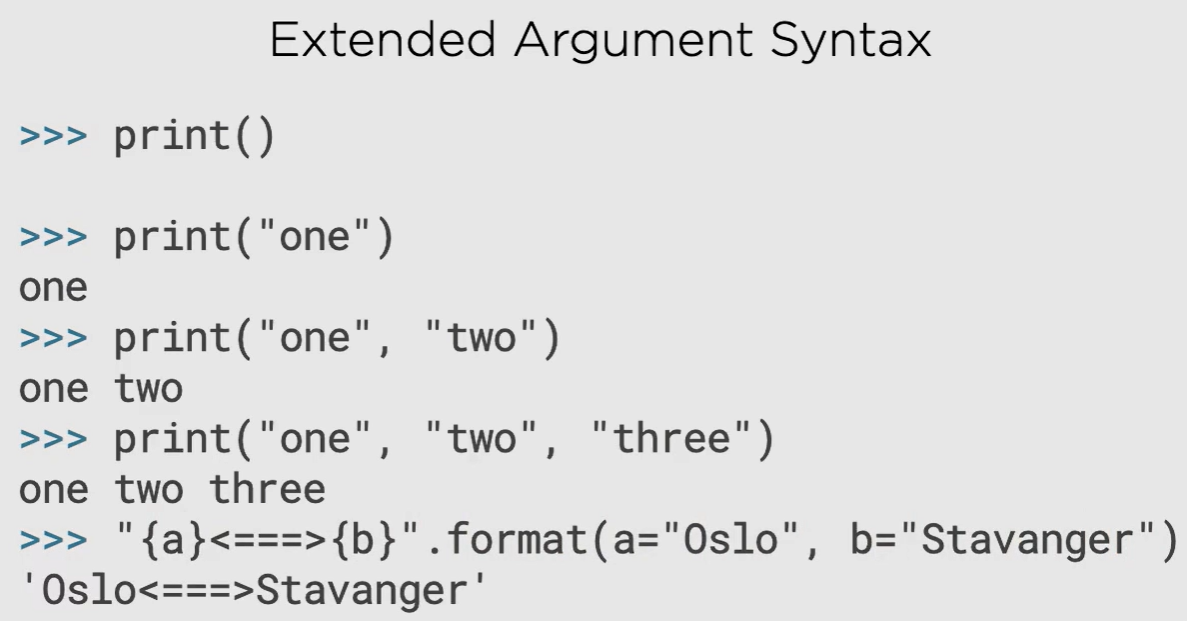
****

****

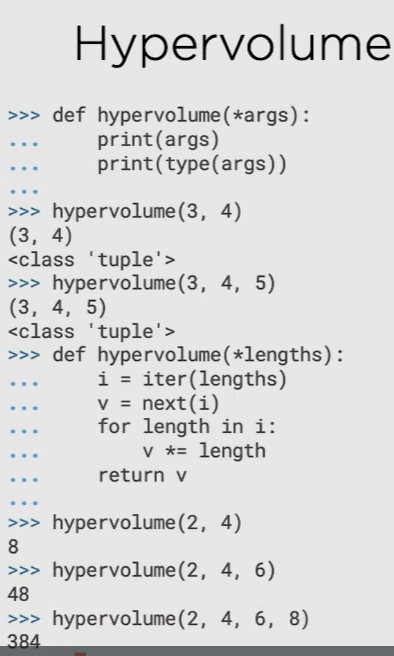
**Extended Argument and Call Syntax**

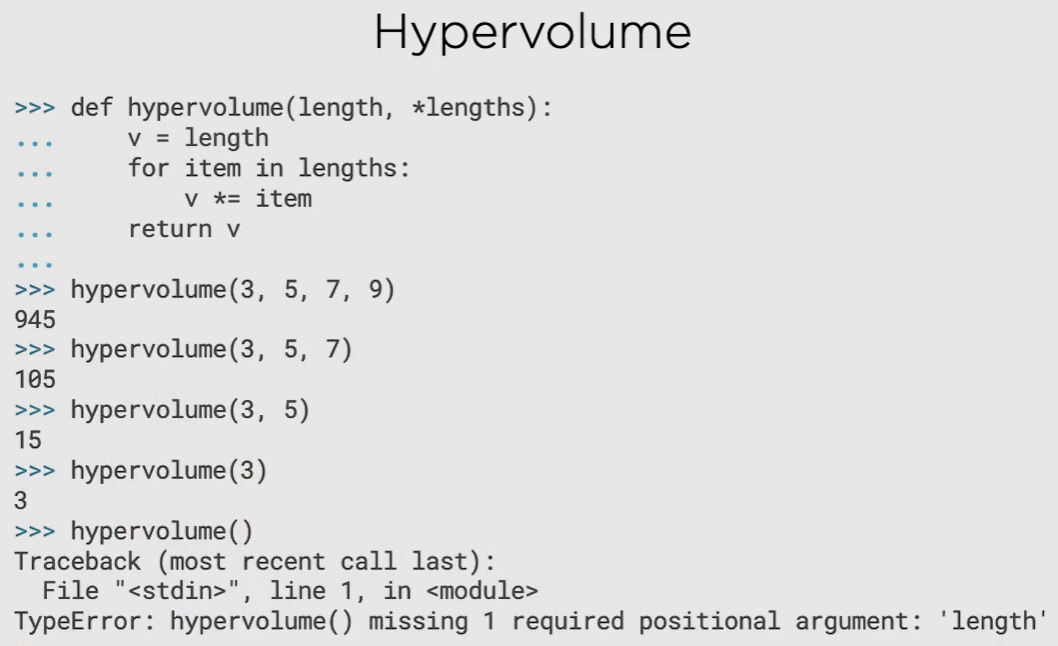
****

**Extended Argument Syntax**

****

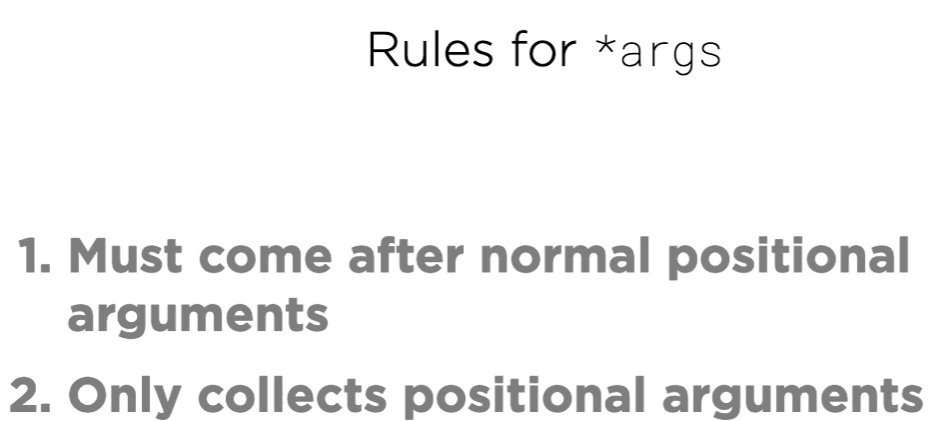
**Arbitrary number of positional arguments**



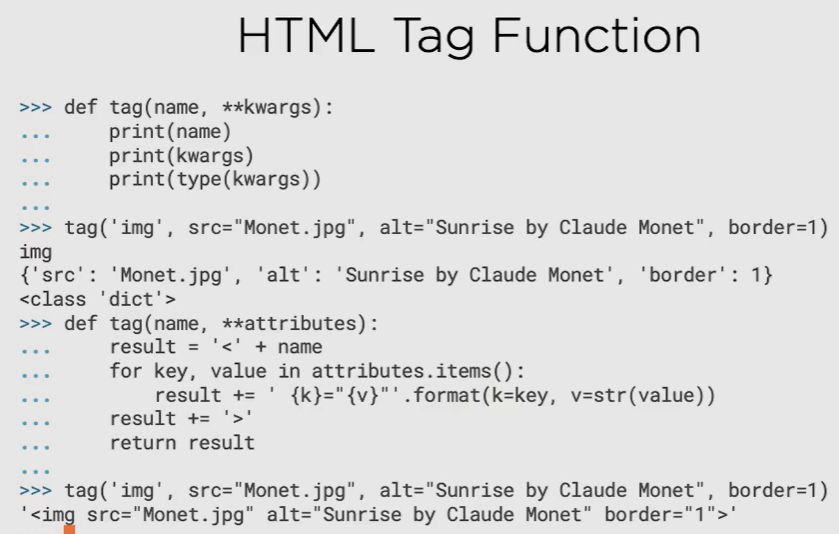


* Here length is normal positional argument



* When you need to accept a variable number of arguments with a positive lower bound you should consider this practice of **using regular positional arguments for the required arguments and \*args to deal with any extra arguments**
* 

**Arbitrary Keyword Arguments**

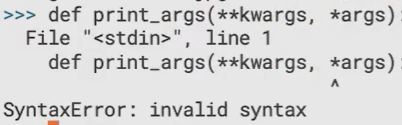
****

This example shows that it is quite possible to combine both positional arguments and keyword arguments.

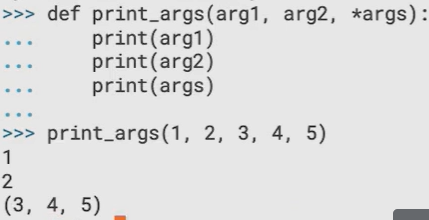
* \*\*kwargs returns dictionary

**RULES**

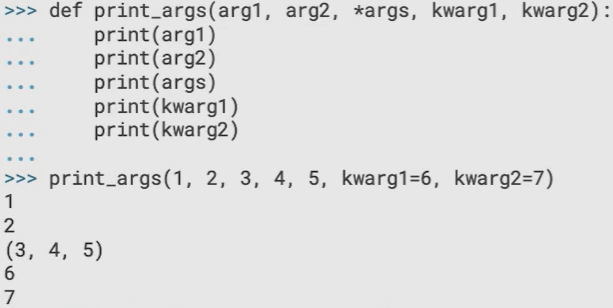
1. If \*args is present, it must always precede \*\*kwargs, so below snippet code isn’t allowed



1. Any arguments preceding \*args are taken to be regular positional arguments

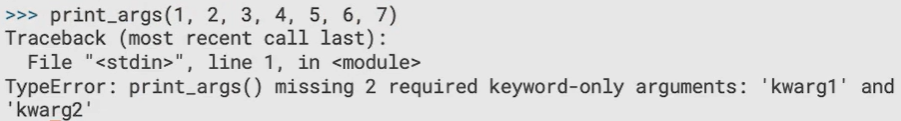


1. Any regular arguments after \*args must be passed as mandatory keywords arguments

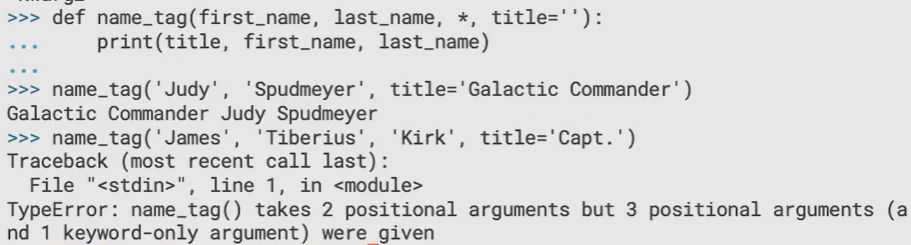


Here above kwarg1=6, kwarg2 =7 are mandatory keyword arguments accepted by hardcoding

1. If failed to pass them as keyword arguments will results in a type erro



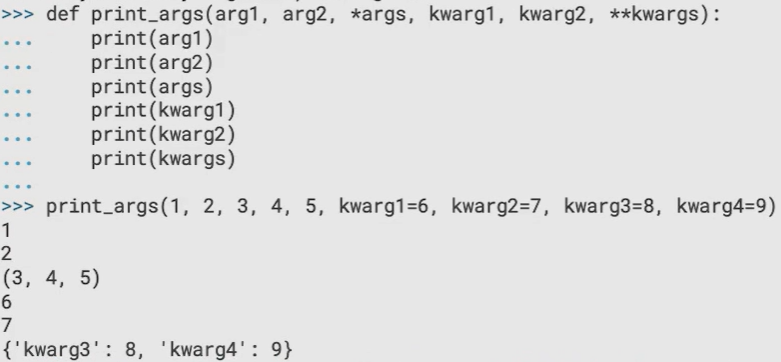
1. There are situations where you will want to use mandatory keyword arguments without accepting/hardcoding an arbitrary number of positional arguments. To support this, Python allows you to omit the name of the \*args argument. The arguments following the asterisk can only be passed to the function as keyword arguments, but Python will not allow you to pass extra positional arguments to the function.

****

**Here above title = ‘ ’, the value is given through function call , this is known as keyword only argument**

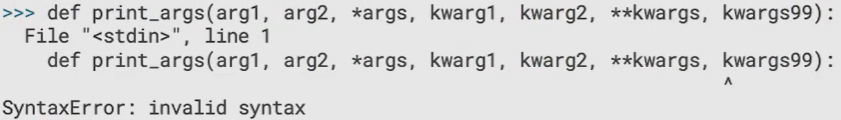
Whereas keyword‑only arguments can only be passed as keyword arguments,

1. Fourthly, and finally, we have the \*\*kwargs arbitrary keyword arguments which, if present, must be last in the argument list

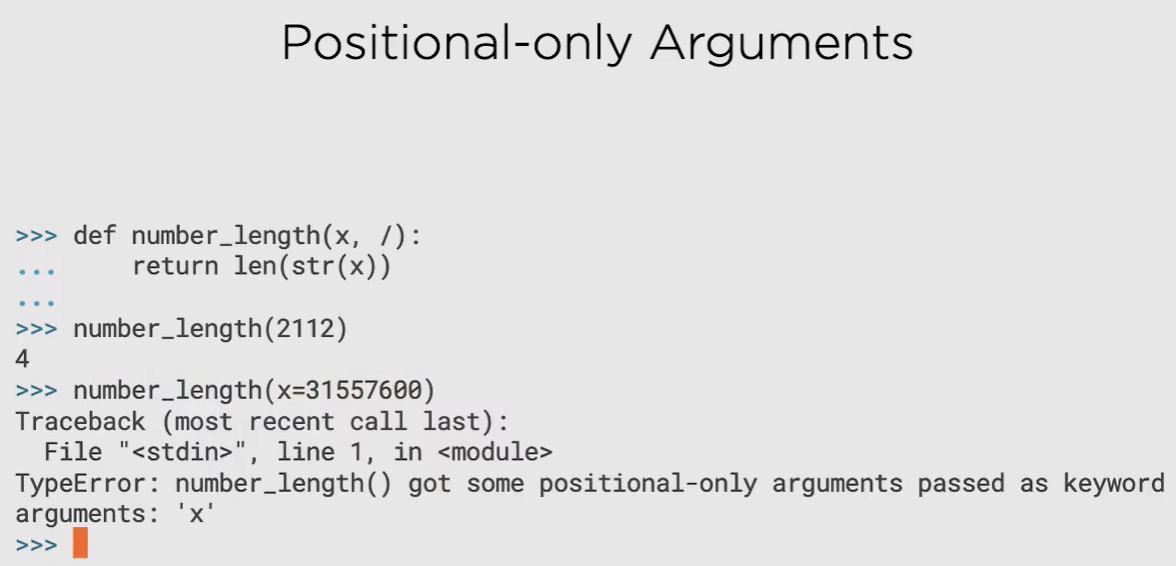


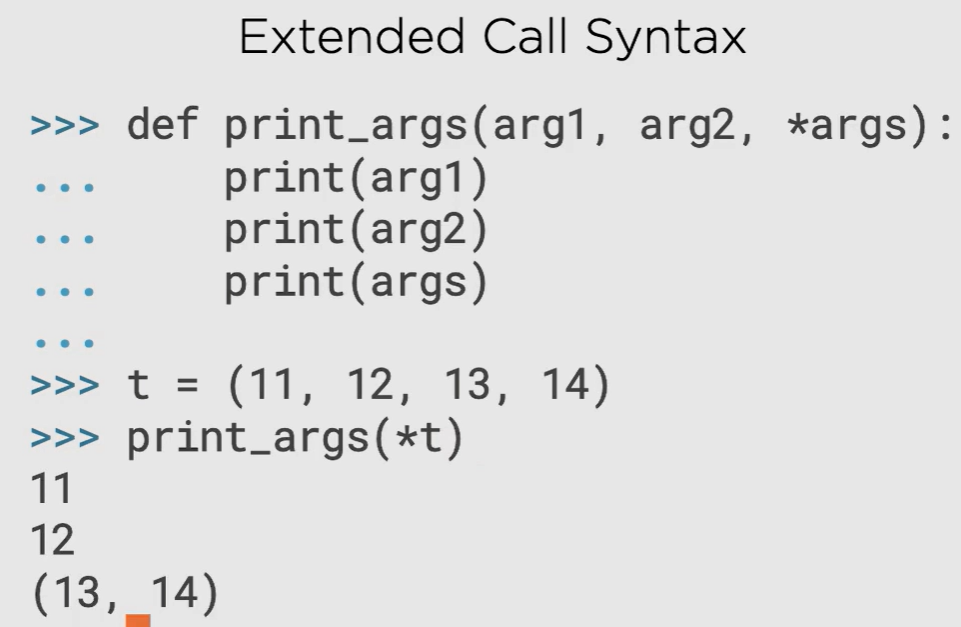
* \*\*kwargs returns dictionary

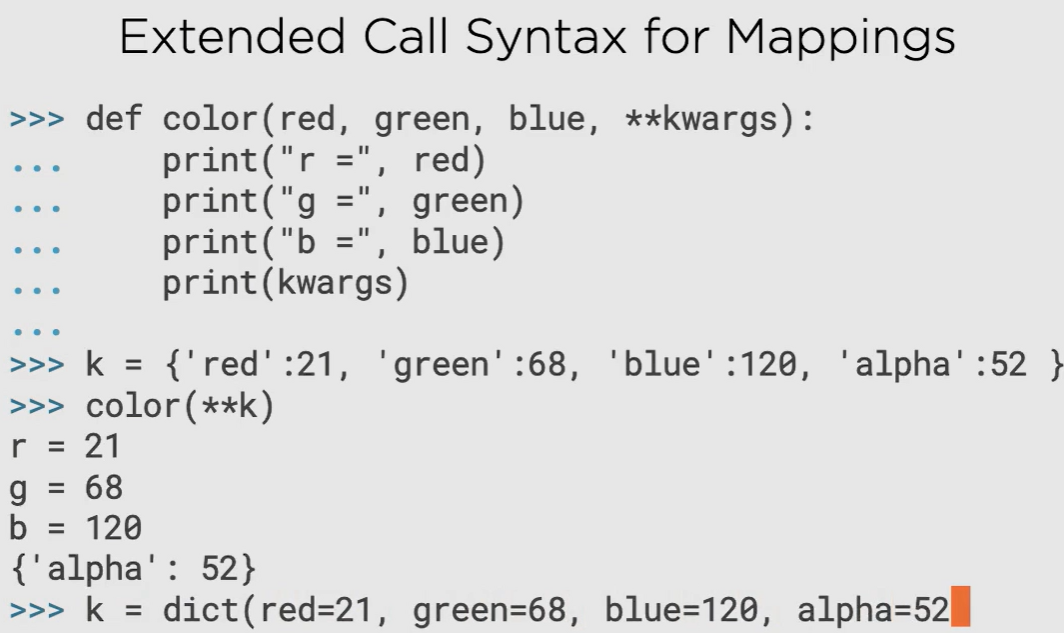
1. Any attempt to define an additional formal argument after \*\*kwargs results in a syntax error

****

1. positional‑only arguments can never be passed with the keyword. To specify positional‑only arguments, you include a forward slash in your function's argument list. All the arguments before the forward slash are positional only, so you could pass them to the function positionally. Any attempt to pass them by keyword will result in a type error.

****

****

****