KEYWORD ARGUMENTS

Keyword Arguments

Recall that positional parameters can, optionally be passed as named (keyword) arguments

func(1, 2, 3)
$$\rightarrow$$
 a = 1, b = 2, c = 3

func(a=1, c=3, b=2)
$$\rightarrow$$
 a = 1, b = 2, c = 3

Using named arguments in this case is entirely up to the caller.

Mandatory Keyword Arguments

We can make keyword arguments mandatory.

To do so, we create parameters after the positional parameters have been exhausted.

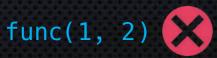
```
def func(a, b, *args, d):
    #code
```

In this case, *args effectively exhausts all positional arguments and d must be passed as a keyword (named) argument

func(1, 2, 'x', 'y',
$$d = 100$$
)
 $\rightarrow a = 1$, $b = 2$, $args = ('x', 'y')$, $d = 100$

func(1, 2, d = 100)

$$\rightarrow$$
 a = 1, b = 2, args = (), d = 100



d was not a keyword argument

We can even omit any mandatory positional arguments:

func(1, 2, 3, d=100)
$$\rightarrow$$
 args = (1, 2, 3), d = 100
func(d=100) \rightarrow args = (), d = 100

In fact we can force no positional arguments at all:

func(1, 2, 3,
$$d=100$$
) \rightarrow Exception

$$func(d=100) \rightarrow d = 100$$

Putting it together

a: mandatory positional argument (may be specified using a named argument)

b: optional positional argument (may be specified positionally, as a named argument, or not at all), defaults to 1

args: catch-all for any (optional)
*: no additional positional arguments

d: mandatory keyword argument

e: optional keyword argument, defaults to True

Code