

Quiz Answers

1. The output here is **1 2 5**, because 1 and 2 are passed to a and b by position, and c is omitted in the call and defaults to 5.
2. The output this time is **1 2 3**: 1 is passed to a by position, and b and c are passed 2 and 3 by name (the left-to-right order doesn't matter when keyword arguments are used like this).
3. This code prints **1 (2, 3)**, because 1 is passed to a and the *pargs collects the remaining positional arguments into a new tuple object. We can step through the extra positional arguments tuple with any iteration tool (e.g., for arg in pargs: ...).
4. This time the code prints **1 {'b': 2, 'c': 3}**, because 1 is passed to a by name and the **kargs collects the remaining keyword arguments into a dictionary. We could step through the extra keyword arguments dictionary by key with any iteration tool (e.g., for key in kargs: ...). Note that the order of the dictionary's keys may vary per Python and other variables.
5. The output here is **1 5 6 4**: the 1 matches a by position, 5 and 6 match b and c by *name positionals (6 overrides c's default), and d defaults to 4 because it was not passed a value.
6. This displays **(1, ['x'], {'a': 'y'})**—the first assignment in the function doesn't impact the caller, but the second two do because they change passed-in mutable objects in place.