

Multiple Arguments - Python Exercise

Below is the image provided along with the recreated question, terminal output, and answer:

The screenshot shows a web browser window displaying a DataCamp exercise page. The page title is "Multiple arguments". The instructions state: "In the previous exercise, you identified optional arguments by viewing the documentation with `help()`. You'll now apply this to change the behavior of the `sorted()` function. Have a look at the documentation of `sorted()` by typing `help(sorted)` in the IPython Shell. You'll see that `sorted()` takes three arguments: `iterable`, `key`, and `reverse`. In this exercise, you'll only have to specify `iterable` and `reverse`, not `key`. Two lists have been created for you. Can you paste them together and sort them in descending order?"

The exercise includes a code editor with the following Python code:

```
1 # Create lists first and second
2 first = [11.25, 18.0, 20.0]
3 second = [10.75, 9.50]
4
5 # Paste together first and second: full
6 full = ____ + ____
7
8 # Sort full in descending order: full_sorted
9 full_sorted = ____
10
11 # Print out full_sorted
12 ____
```

Below the code editor is an IPython Shell window with the prompt "In [1]:".

Recreated Question and Terminal

Multiple Arguments

In the previous exercise, you identified optional arguments by viewing the documentation with `help()`. You'll now apply this to change the behavior of the `sorted()` function.

Have a look at the documentation of `sorted()` by typing `help(sorted)` in the IPython Shell.

You'll see that `sorted()` takes three arguments: `iterable`, `key`, and `reverse`. In this exercise, you'll only have to specify `iterable` and `reverse`, not `key`.

Two lists have been created for you.

Can you paste them together and sort them in descending order?

Instructions:

- Use + to merge the contents of first and second into a new list: full.
- Call sorted() on full and specify the reverse argument to be True. Save the sorted list as full_sorted.
- Finish off by printing out full_sorted.

Answer

```
# Create lists first and second
first = [11.25, 18.0, 20.0]
second = [10.75, 9.50]

# Paste together first and second: full
full = first + second

# Sort full in descending order: full_sorted
full_sorted = sorted(full, reverse=True)

# Print out full_sorted
print(full_sorted)
```

Explanation of the Answer

The code merges the lists first and second using the + operator. The sorted() function is called on the combined list full, with the reverse argument set to True to sort in descending order. The result is stored in full_sorted and printed, displaying the numbers from largest to smallest.