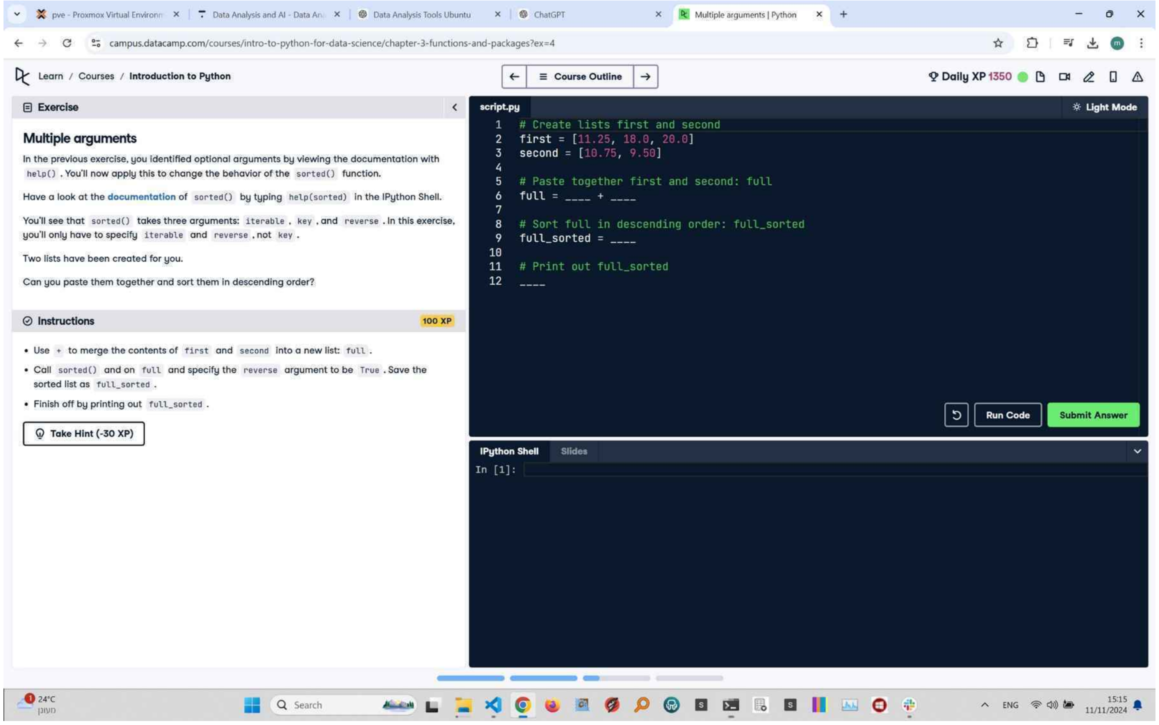
Multiple Arguments - Python Exercise

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



pve - Proxmox Virtual Environ
Data Analysis and AI - Data An
Data Analysis Tools Ubuntu
ChatGPT
Multiple arguments | Python
campus.datacamp.com/courses/intro-to-python-for-data-science/chapter-3-functions-and-packages?ex=4
Learn / Courses / Introduction to Python
Course Outline
Daily XP 1350
Exercise
script.py
Light Mode
Run Code
Submit Answer
IPython Shell
Slides
Search
24°C
ENG
15:15
11/11/2024

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 100 XP

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

Take Hint (-30 XP)

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

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

# Sort full in descending order: full_sorted
full_sorted = ____

# Print out full_sorted
____
```

```
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.