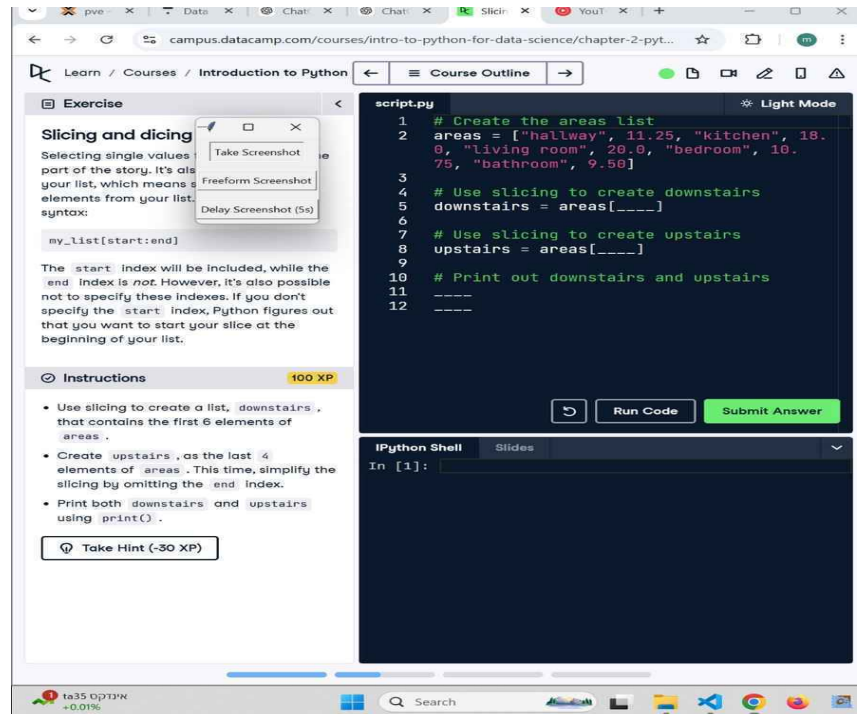


## Slicing and Dicing - Python Exercise

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



## Recreated Question and Terminal

### Slicing and Dicing

Selecting single values from a list is just one part of the story. It's also possible to slice your list, which means selecting multiple elements from your list. Use the slicing syntax:

```
my_list[start:end]
```

The start index will be included, while the end index is not. However, it's also possible not to specify these indexes. If you don't specify the start index, Python figures out that you want to start your slice at the beginning of your list.

### Instructions:

- Use slicing to create a list, downstairs, that contains the first 6 elements of

areas.

- Create upstairs, as the last 4 elements of areas. This time, simplify the slicing by omitting the end index.
- Print both downstairs and upstairs using print().

## Answer

```
# Create the areas List
areas = ["hallway", 11.25, "kitchen", 18.0, "living room", 20.0, "bedroom",
10.75, "bathroom", 9.50]

# Use slicing to create downstairs
downstairs = areas[:6]

# Use slicing to create upstairs
upstairs = areas[6:]

# Print out downstairs and upstairs
print(downstairs)
print(upstairs)
```

## Explanation of the Answer

The code uses slicing to divide the areas list. `downstairs = areas[:6]` selects the first 6 elements, while `upstairs = areas[6:]` selects the remaining elements from index 6 onward. This approach efficiently splits the list into two parts for better data organization.

\*

Let's clarify why the slicing is done as `areas[6:]` and not `areas[4:]`.

### Understanding the Indexes in the areas List

Here's the areas list:

python

Copy code

```
areas = ["hallway", 11.25, "kitchen", 18.0, "living room", 20.0, "bedroom",  
10.75, "bathroom", 9.50]
```

### **Explanation of the Indexes**

Each element in the areas list has an index:

- "hallway": index 0
- 11.25: index 1
- "kitchen": index 2
- 18.0: index 3
- "living room": index 4
- 20.0: index 5
- "bedroom": index 6
- 10.75: index 7
- "bathroom": index 8
- 9.50: index 9

### **Slicing for downstairs and upstairs**

#### **1. Creating downstairs:**

- o We want the first 6 elements of areas, which means from index 0 to index 5.
- o This is done using areas[:6].

#### **2. Creating upstairs:**

- o We want the last 4 elements of areas. These elements are at indexes 6, 7, 8, and 9.
- o To get these elements, we start from index 6 and go to the end of the list.
- o That's why we use areas[6:].

### **Why Not Use areas[4:]?**

If we used areas[4:], we would be slicing from the 5th element ("living room") onward, which is incorrect. The instruction specifies that upstairs should include the last 4 elements, and those are:

- "bedroom" (index 6)
- 10.75 (index 7)
- "bathroom" (index 8)
- 9.50 (index 9)

Thus, `areas[6:]` correctly captures the last 4 elements of the list.