

## Create Lists with Different Types - 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 titled "Create Lists with Different Types". The exercise is part of the "Introduction to Python" course. The instructions state: "Although it's not restricted to one type, a list can also contain a mix of Python types including strings, floats, and booleans. You're now going to add the room names to your list, so you can easily see both the room name and size together. Some of the code has been provided for you to get you started. Pay attention here! 'bathroom' is a string, while bath is a variable that represents the float 9.50 you specified earlier."

The exercise includes a code editor with the following code:

```
1 hall = 11.25
2 kit = 18.0
3 liv = 20.0
4 bed = 10.75
5 bath = 9.50
6
7 # Adapt list areas
8 areas = [____, hall, ____, kit, "living room", liv, ____, bed, "bathroom", bath]
9
10 # Print areas
11 ____
```

The interface also includes a "Python Shell" section for running the code and a "Submit Answer" button.

### Recreated Question and Terminal

#### Create Lists with Different Types

Although it's not restricted to one type, a list can also contain a mix of Python types including strings, floats, and booleans.

You're now going to add the room names to your list, so you can easily see both the room name and size together.

Some of the code has been provided for you to get you started. Pay attention here! "bathroom" is a string, while bath is a variable that represents the float 9.50 you specified earlier.

#### Instructions:

- Finish the code that creates the areas list. Build the list so that the list first contains the name of each room as a string and then its area. In other words, add the strings "hallway", "kitchen" and "bedroom" at the appropriate locations.

- Print areas again; is the printout more informative this time?

## Terminal Output

Output not provided, as it is expected to match the input formatting.

## Answer

```
# Given areas
hall = 11.25
kit = 18.0
liv = 20.0
bed = 10.75
bath = 9.50

# Adapt list areas
areas = ["hallway", hall, "kitchen", kit, "Living room", liv, "bedroom", bed,
        "bathroom", bath]

# Print areas
print(areas)
```

## Explanation of the Answer

The code constructs the areas list to include both room names as strings and their corresponding areas. Adding room names makes the output clearer and more descriptive. The list is then printed, showing names alongside values, improving the readability and understanding of the room sizes.