|  |  |
| --- | --- |
| **Create a list**  100xp  As opposed to int, bool etc, a list is a **compound data type**: you can group values together:  a = "is"  b = "nice"  my\_list = ["my", "list", a, b]  After measuring the height of your family, you decide to collect some information on the house you're living in. The areas of the different parts of your house are stored in separate variables for now, as shown in the script.  **Instructions**   * Create a list, areas, that contains the area of the hallway (hall), kitchen (kit), living room (liv), bedroom (bed) and bathroom (bath), in this order. Use the predefined variables. * Print areas with the [print()](https://docs.python.org/3/library/functions.html#print) function. | # area variables (in square meters)  hall = 11.25  kit = 18.0  liv = 20.0  bed = 10.75  bath = 9.50  # Create list areas  # Print areas |

|  |  |
| --- | --- |
| **Create list with different types**  100xp  A list can contain any Python type. Although it's not really common, a list can also contain a mix of Python types including strings, floats, booleans, etc.  The printout of the previous exercise wasn't really satisfying. It's just a list of numbers representing the areas, but you can't tell which area corresponds to which part of your house.  The code on the right is the start of a solution. For some of the areas, the name of the corresponding room is already placed in front. Pay attention here! "bathroom" is a string, while bath is a variable that represents the float 9.50 you specified earlier.  **Instructions**   * Finish the line of code that creates the areas list such that the list first contains the name of each room as a string, and then its area. More specifically, add the strings "hallway", "kitchen" and "bedroom" at the appropriate locations. * Print areas again; is the printout more informative this time? | # area variables (in square meters)  hall = 11.25  kit = 18.0  liv = 20.0  bed = 10.75  bath = 9.50  # Adapt list areas  areas = [hall, kit, "living room", liv, bed, "bathroom", bath]  # Print areas |

|  |  |
| --- | --- |
| Select the valid list 50xp  A list can contain any Python type. But a list itself is also a Python type. That means that a list can also contain a list! Python is getting funkier by the minute, but fear not, just remember the list syntax:  my\_list = [el1, el2, el3]  Can you tell which ones of the following lines of Python code are valid ways to build a list?  A. [1, 3, 4, 2] B. [[1, 2, 3], [4, 5, 7]] C. [1 + 2, "a" \* 5, 3] Possible Answers     A, B and C      B      B and C      C |  |

|  |  |
| --- | --- |
| List of lists 100xp  As a data scientist, you'll often be dealing with a lot of data, and it will make sense to group some of this data.  Instead of creating a flat list containing strings and floats, representing the names and areas of the rooms in your house, you can create a list of lists. The script on the right can already give you an idea.  Don't get confused here: "hallway" is a string, while hall is a variable that represents the float 11.25 you specified earlier. Instructions  * Finish the list of lists so that it also contains the bedroom and bathroom data. Make sure you enter these in order! * Print out house; does this way of structuring your data make more sense? * Print out the type of house. Are you still dealing with a list? | # area variables (in square meters)  hall = 11.25  kit = 18.0  liv = 20.0  bed = 10.75  bath = 9.50  # house information as list of lists  house = [["hallway", hall],  ["kitchen", kit],  ["living room", liv]]  # Print out house  # Print out the type of house |