

Session Title: Lists and Dictionaries

Session No: 4

#### Learning Objectives:

By the end of this session students will be able to:

- Create lists to store multiple data values
- Identify different list functions
- Combine lists and for loops
- Construct dictionaries to structure data

#### Session Outline

##### Intro & Framing

5 mins

This session focuses on lists and dictionaries. So far students have used string, integer, float and Boolean data types along with variables. With lists they can make collections of multiple data values. Creating lists, basic list operations, and using lists with for loops are covered in this session.

The second part of this session introduces the dictionary data type. Similar to lists, dictionaries allow programmers to create structure of multiple data values. Dictionaries also allow programmers to label the data to give it meaning through the use of dictionary keys.

##### Block #1:

10 minutes

The first block introduces lists. Lists allow students to have a collection of multiple data values in a single variable.

Students are taught how to access data in a list using indexes. It is important to emphasize to students that list indexes start counting from 0.

#### Relevant Exercise(s):

##### Exercise 4.1: 5-10 minutes

- In this exercise students will practice creating lists and accessing the list items using indexes
- The students need to add an if statement to check the value of the first item
- Recommend that the students try running the program with and without "shorts" as the first item in the list
- Extension: Change the other items in the list to clothing more appropriate to winter if the first item is shorts

##### Block #2: List Functions

This block introduces several functions that are useful when working with lists. The functions covered are the ``len()``, ``min()``, ``max()``, ``sorted()``, and ``reversed()`` functions.

Relevant Exercise(s):

Exercise 4.2: 5 minutes

- In this exercise students will practice using list functions
- The scenario is that they have a list of scores for a game and need to output stats for the scores
- Extension: Display the scores in descending order

Block #3: in and append()

5 minutes

This block covers two useful features of lists, specifically The ``in`` operator and ``append()`` method. The ``in`` operator can check whether a value is in a list. The ``append()`` method can add new values to a list.

Relevant Exercise(s):

Exercise 4.3: 10 minutes

- The exercise is designed to practice the ``in`` operator and ``append()`` function
- The scenario is that I want to add butter to my shopping list if bread is in my shopping list
- If students feel overwhelmed suggest building it up in stages i.e. check bread is in the list with an if before doing anything else
- Extension: only add butter to the list if it is not already in the list

Block #4: For Loops and Lists

5 minutes

This section is about combining lists and for loops. Students learned about using for loops with the ``range()`` function in session two.

When explaining the code examples on the slides, instructors should run through each repeat/iteration step-by-step. Explain what the value of the for loop's variable is, what happens in the code block, and how after running the code block the loop gets the next value and starts again.

Relevant Exercise(s):

Exercise 4.4: 10 minutes

- The scenario for this exercise is to calculate how much I've spent on lunch this week
- Students will need to create a for loop and add each of the items in the ``costs`` list to the `total_cost` variable

- Extension: work out the average that I spend on lunch for the week

#### Block #5: Dictionaries

5 minutes

This block introduces dictionaries. Instructors should emphasize that each item in a dictionary has a key and a value.

Relevant Exercise(s):

Exercise 4.5: 5 minutes

- Students are given a dictionary that has data about a location and need to get data from that dictionary
- Extension: Print the values of longitude and latitude from the inner dictionary

#### Block #6: Dictionaries in Lists

5 minutes

Working with dictionaries inside of lists is very common for programmers. This block introduces students to using dictionaries inside of lists.

Relevant Exercise(s):

Exercise 4.6: 5 minutes

- Students need to use a combination of a for loop and dictionary indexes to print the name, colour and price of each item in the example list
- Extension: Add more items to the list

#### Block #7: Random Choice

5 minutes

This block introduces students to the `choice()` function in the random module. This function is used to randomly select an item from a list.

If you are short of time in this session, feel free to skip this block

Relevant Exercise(s):

Exercise 4.7: 5+ minutes

- Students will use `random.choice()` to generate randomised names from a list of first names and last names
- Extension: Create randomised sentences using a list of nouns and a list of verbs
- If you are short of time in this session, feel free to skip this exercise

Recap & Closing

5 mins

Recap questions:

Question 1:

What shape brackets are used for creating a list and what shape brackets are used for creating a dictionary?

Answer: Square brackets for lists and curly brackets for dictionaries

Question 2:

What is the result of this program?

```
cheeses = [  
    'brie',  
    'cheddar',  
    'wensleydale',  
    'edam',  
]  
  
print(cheeses[4])
```

Answer:

The program will raise an error as list index 4 does not exist

Question 3:

This program raises an error when I run it. What do I need to change to get it to run?

```
trees = [  
    {'leaf_colour': 'green', 'height': 2120},  
    {'leaf_colour': 'green', 'height': 2300},  
  
    new_tree = {  
        'leaf_colour': 'green',  
        'height': 1020  
    }  
  
trees.append(new_tree)  
  
print(trees)
```

Answer:

The trees list is missing a closing square bracket

#### Homework Tasks

Learning Task:

Session 4 homework questions in the student guide

#### Guide for Instructors

General comments