

Session Title: Logic and If Statements

Session No: 3

Learning Objectives:

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

- Identify Boolean data values
- Explain and combine comparison and logical operators to compare data values
- Apply if, else and elif Statements to control the flow of programs
- Build short games using randomisation

Session Outline

Intro & Framing

5 mins

The teaching blocks and exercises in this session focus on comparison and logical operators, and if statements.

Students are first introduced to comparisons and logical operators and shown how they return Boolean values. After they have a good grasp of these concepts, they are shown how to combine them with if statements to automate decision making in their programs.

At the end of the session students can select one (or more) of three exercises that combine what they have learned in the session. These three exercises are in order of difficulty: Exercise 3.6 is intended for students that need more support, Exercise 3.7 for a student that has made good progress, and 3.8 for students that are happy to take on a more challenging task.

Block #1: Comparisons and Logical Operators

2 x 5 minutes

This first part of this block introduces comparisons, such as equals (==) and less than (<). There is not enough time in the session to dedicate time to each comparison, so they are summarised in a table.

Make sure you point out to students that `==` is used to compare values and `=` is to set variables, as students frequently get confused between the two.

The second part of this block introduces the boolean operators `and`, `or`, and `not`.

Relevant Exercise(s):

Exercise 3.1: 5-10 minutes

- The scenario for this exercise is that the program needs to decide whether the student should go to a burger restaurant based on the price
- Students will practice operators in this exercise and will need to choose the most appropriate operator for the program

- The solution is to use the less than comparison to check whether a number they input is less than 10
- Students can refer to the previous example to help them with their code
- Make sure that students have a solution for this exercise as it will be used throughout the following exercises

Exercise 3.2: 5 minutes

- This exercise builds on the previous exercise
- The scenario is that the program needs to check whether the burger restaurant has a vegetarian option
- The solution is to add a second input, check the input with the equals operator, and use the `and` operator to check whether the input meets both conditions
- Extension: Add a check to see if the restaurant's rating is 3 or more

Block #2: If Statements

5 minutes

This next block introduces students to if statements. They are shown if statements that use a single comparison as well as those that combine multiple comparisons using Boolean operators.

Relevant Exercise(s):

Exercise 3.3: 5 minutes

- This exercise requires students to adapt their earlier burger program to use an if statement to determine the output of the program

Block #3: Else Statements

5 minutes

This block builds on the previous one and introduces `else` statements.

It is important to point out to students that `else` statements are always paired with an `if` statement. `else` statements cannot be used without an `if` statement.

Do not mention that `else` statements can also be used with `for` loops or `try/except` as it is too much information for the student at this stage.

Relevant Exercise(s):

Exercise 3.4: 10 minutes

- Students will practice using `if` and `else` statements in this exercise
- This exercise scenario requires students to create a bill calculator
- They will need to complete the code given in the exercise
- The solution is to use an `if` statement to apply a discount depending on user input and use an `else` statement to say if no discount was applied

Block #4: Elif Statements

5 minutes

This block introduces `elif` statements that are tied closely with `if` and `else`. Mention to students that they can use multiple `elif` statements together with a single `if` statement.

Relevant Exercise(s):

Exercise 3.5: 10 minutes

- The scenario for this exercise is that the students are making a pizza and need to determine if the temperature is set correctly
- The solution to this exercise is to use if, elif and else to output different messages depending on the input
- Extension: Ask at the start of the program whether you're cooking a cake or a pizza. The suggested temperatures for a cake should be different from the pizza.

Block #5: Random

5 minutes

This final block introduces the `random` module in Python's standard library.

The exercises in this block combine `if`, `else` and `elif` with the random module to create basic games of chance.

Students have a choice of three different exercises and should choose whichever exercise they feel comfortable attempting. The exercises are ordered by difficulty.

Exercise 3.6 is for students that might need additional support, exercise 3.7 is for students that have made good progress, and exercise 3.8 is for students that want a more challenging exercise.

Relevant Exercise(s):

Students should choose to complete one exercise, but can attempt others if they finish.

Exercise 3.6: 10-15 minutes

- This program is a basic heads or tails guessing game
- If the random coin flip matches the choice input by the user then they win
- If the random coin flip does not match the choice input by the user then they lose
- The students should use the code included in the exercise as a starting point
- Extension: The program should show a message to the user if they enter a choice that isn't heads or tails

Exercise 3.7: 10-15 minutes

- This program simulates rock-paper-scissors
- Students will need to use if, else and elif statements to complete the rest of the program

- Extension: the program should show a message to the user if they enter a choice that isn't rock, paper or scissors
- If the students finish this exercise, they should try to complete exercise 3.6 without copying the starter code

Exercise 3.8: 10-15 minutes

- This exercise is a simplified version of a roulette table
- The requirements of the program are listed in the exercise
- Some starter code for the random result is given in the exercise

Recap & Closing

Time: 5 mins

Recap questions:

Question 1:

Equals to (==) is a comparison operator. Name two more comparison operators

Answers: Greater than, less than, greater than or equal, less than or equal, not equal to

Question 2:

What is the output of this code?

Answer:

True, False, True, True

Question 3:

I expect this code to output "This is too many apples", but instead it outputs "That is a sensible number of apples". Why does this happen?

Answer:

The program checks if the number of apples is more than ten before it can check if the number is more than 50. The order of these two comparisons needs to be swapped.

Homework Tasks

Learning Task:

Session 3 homework questions in the student guide

Guide for Instructors

General comments

