Lesson 8: Logical expressions

Introduction

This lesson continues to develop learners' understanding of logical expressions by introducing the operators AND, and OR. It begins with a Parson's Puzzle to check their comprehension of selection. It then moves on to an unplugged activity that introduces AND, and OR. Learners will then walk through code and investigate it, before writing their own logical expressions.

Learning objectives

- Describe how Boolean/logical operators can be used in expressions
- Walk through code that uses conditions with Boolean/logical operators (AND, OR)
- Write and use expressions that use Boolean/logical operators (AND, OR)

Key vocabulary

Boolean expression, logical expression, Boolean operator, logical operator, expression, operator

Preparation

Subject knowledge:

For this lesson you will need to be able to write an expression in Python that uses Boolean/logical operators.

Pedagogy:

The starter activity uses a Parson's Puzzle. It is an exercise that requires learners to rearrange lines of code in the correct sequence in order to form a working code segment.

The benefits of Parson's Puzzles are that they:

- Maximize engagement of learners
- Constrain the logic
- Avoid common errors that can be barriers to learning to code
- Model good code
- Provide immediate feedback
- Allow for easier analysis of what learners don't know

You will need:

- Slides
- Parson's Puzzle: A0 worksheet
- Parson's Puzzle code (https://replit.com/@awade05/parsonspuzzle)
- A set of jumbo playing cards
- Sandwich order calculator: A2 worksheet and solutions
- Spot the five errors: AP worksheet

You may also need:

- Pizza complete code (https://replit.com/@awade05/pizzacomplete#main.py)
- Vocabulary sheet

Assessment opportunities

The starter activity has a Parson's Puzzle that can be used to assess what learners remember about selection. The solution to the Parson's Puzzle is available on the slide deck so that learners can self mark. The 'Playing cards' unplugged activity will allow you to check your learners' understanding of the use of AND, and OR in expressions before they move on to the PRIMM activity. Solutions have been provided for the 'Using AND, and OR in Python' activity, which can be used for peer, self, or teacher marking. The plenary activity will allow learners to spot errors in a program that they should recognise. Answers are provided for self- or peer-assessment.

Outline plan

Please note that the slide deck labels the activities in the top right-hand corner to help you navigate the lesson.

**Timings are rough guides*

Starter activity (Slides 2–6) 5–6 mins	**Parson's Puzzle on selection** Distribute the 'Parson's Puzzle' worksheet for the starter activity. Learners will be completing a Parson's Puzzle. They should solve the Parson's Puzzle and check if they are correct using the link for the Paron's Puzzle code. You can then go through the slides to further explore the solution and walk through the code.

Activity 1 (Slides 8–15) 5–7 mins	**Unplugged activity: The playing cards** Playing cards are a great addition to the computer science classroom, we recommend that you get some jumbo playing cards to use in the classroom. However, if these are not available, you could find some online and print them at school. Playing cards can also be used to teach searching and sorting algorithms in the algorithms unit, they are a really versatile resource. This unplugged activity requires that all learners have one playing card each. Note that traditionally you might teach `and`, `or`, and `not` together. However, you will be using not equal to (`!=`) instead of the word `not` in your Python code. You could use `if not suit == "hearts":` but it makes sense for simple comparisons to use `if suit != hearts:` The word `not` will be introduced when using string handling and lists later in the course as it is more suited to expressions such as `not in guest_list:` Use the slides to introduce new expressions to your learners that include the words `and`, and `or`. Learners must look at their playing cards and use the values on them to decide whether the condition is True or False for their card. They then move to the correct part of the classroom for True or False to make that decision. Slide 12 contains an `if` statement that is always False. This might throw them a little but it will certainly challenge their thinking. The slides also introduce example code with the conditions written. After each condition has been used with the learners, show them the relevant Python code on the slides.

Activity 2 (Slide 16) 35 mins	**Using AND, and OR in Python: A PRIMM approach** The ‘Sandwich order calculator’ worksheet uses the PRIMM approach to allow your learners to investigate and modify a program that calculates the cost of a sandwich. It introduces them to `and`, and `or` within the context of a program. Once learners have investigated and modified the program, the worksheet gives them a **make** task. The **make** task asks learners to create a pizza-cost program. You will notice that there are strong similarities between this program and the sandwich calculator program. The worksheet guides learners through the make task, while scaffolding is gently removed to help increase their confidence.
Plenary (Slides 17–18) 5 mins	**Spot the five errors** Distribute the worksheet that contains a program that has five errors. Ask learners to circle each error that they find and draw an arrow to it, explaining what the error is. After giving the learners time to complete the activity, provide the answers on slide 18. Learners can self or peer mark their work.

Homework	**Optional**: Complete the **make** program from the ‘Sandwich order calculator’ handout.