Lesson 2: Variables

Introduction

Learners will find out about variables during this lesson. They will learn about the purpose of variables, but also the technical aspects of creating variables to a uniform standard. Variable **declaration** is not used in Python, so a wider look at this through other programming languages will help learners gain an insight into its meaning.

Learning objectives

- Use meaningful identifiers
- Determine the need for variables
- Distinguish between declaration, initialisation, and assignment of variables
- Demonstrate appropriate use of naming conventions
- Output data (e.g. `print (my_var)`)

Key vocabulary

Declaration, assignment, initialisation, variable, naming convention

Preparation

Subject knowledge:

An understanding of variable declaration, initialisation, and assignment is needed for this lesson. You will also need to know about the naming conventions used for Python, which can be found in the user documentation (https://peps.python.org/pep-0008/#overriding-principle).

Common misconceptions:

M1 A variable can store multiple items; it may store the ‘history’ of values assigned.

M7 Assignment statements such as a=b work both directions: they swap the values of two variables.

M8 A variable name needs to be a single letter; longer identifiers are interpreted as (parts of) commands.

M12 Variables are initially empty containers and do not need to be initialised.

You will need:

- Slides
- Python naming conventions documentation
- Silly sentences: A2 worksheet and solutions

Optional:

- The starter activity (slide 2) contains small text that you may wish to print and enlarge for those that need it
- Lesson vocabulary

Assessment opportunities

Review the answers that learners have provided for the ‘Silly sequence’ activity. A model answer sheet has been provided to help. You could do this as a self- or peer-assessment if required.

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 2–4) 3–5 mins	**Compare and contrast the user areas** Present learners with two screenshots of two user areas. One screenshot shows many Python files with random names that have just been created by tapping the keyboard without much thought. The other screenshot contains Python files that have clear names given to them. In pairs, learners should discuss the two user areas and decide which one is better. They need to provide reasons for their answers. Ask learners why it is important to name files correctly. Hopefully they will come up with suggestions like: ● It helps you find the file again next time ● It helps other people know what the file is for if you share it ● It stops you losing track of files ● It saves time trying to find the files in the future ● A clear name lets you know what is inside the file without having to open it Slide 4 introduces learners to the concept of naming conventions and tells them that these will be explored in this lesson.

Activity 1 (Slides 6–20) 8 mins	**Variables** This activity has been designed to introduce learners to the key terms relating to variables. It shows them with a definition and an example. The clear descriptions of how a variable works have been written to avoid common misconceptions about variables. There is a short piece of code for learners to try out, but you could complete this as a short live coding activity if that makes more sense in your setting. A link to the naming conventions section of the Python documentation has been provided for you on slide 19. You might wish to show this to your learners or ask them to explore it to give them a feel for how documentation works. This will be covered properly later in the unit. The learners are not expected to remember all of the key terms just from this slide show. The PRIMM activity that follows should help to embed those terms further through practical activities.

Activity 2 (Slide 21) 40 mins	**Silly sentences** This activity has been designed to allow learners to gain a deeper understanding of variables and how they can be used in a program. Learners have a “silly sentences” program that fills in the blanks with nouns, adjectives, and adverbs. They will predict what will happen when it is run, investigate the code, then modify it to personalize the program. After completing steps, give the learners a step where they need to create their own silly stories. The expectation is that learners will start to make steps during the lesson, but this can also be completed for homework. Model answers have been provided if you wish to assess this work. It could also be used for peer- or self-assessment, if that feels appropriate.

Plenary (Slide 22) 5–10 mins	**Peer review** For the last part of the lesson, give your learners a chance to sit with a partner and peer-review their silly stories. They should focus on aspects like: • Has the programmer used a variety of variables to fill in the blanks? • Are there any syntax errors? • Are they any logic errors? • How could the story be improved?
Homework	Learners can improve their silly stories for homework by adding more paragraphs to the program. They can also make improvements based on the feedback given in the peer review.