

Lesson 3: Input

Introduction

During this lesson learners will start to add interactivity to their programs by introducing the `input()` function. Learners will be given a demonstration of how `input()` is used in Python and then asked to add this feature in their silly stories that were created last lesson. Learners will then be introduced to the five main data types that they need to understand. This is done first through theory teaching and then a practical activity where learners predict, run and investigate a program that includes the five data types. Whilst learning about input, learners will be introduced to functions and data validation techniques. These will be covered in more detail later on in the course.

Learning objectives

- Obtain input from the keyboard in a program
- Differentiate between the data types: integer, real, Boolean, character, string
- Cast variables by calling a function that will return a new value of the desired data type
- Define runtime errors in programs
- Define validation checks

Key vocabulary

Input, Boolean, integer, string, character, real/float

Preparation

Subject knowledge:

You will need to know how inputs and variables are used in the context of flowcharts and Python programming. You will also need to know the definitions for the main data types used and how to use these within Python.

You will need:

- Slides
- Mini data collection program: A3 worksheet and solutions
- [A3 start code](#)

You may also need:

- Lesson vocabulary
- Live coding demonstration video - this is for teacher use only

Assessment opportunities

The starter activity gives you the opportunity to see what learners remember from the last lesson and if they have developed any **misconceptions** around variables.

The 'Mini data collection program' activity contains a series of investigation tasks for learners to complete. Model answers have been provided for this to help with peer- or self-assessment.

The plenary activity briefly checks if learners remember the five data types covered in the lesson.

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*

<p>Starter activity</p> <p>(Slides 2–4)</p> <p>5–8 mins</p>	<p>Multiple choice questions</p> <p>Three multiple choice questions have been designed to assess what learners understood from the last lesson and to address any misconceptions.</p> <ul style="list-style-type: none"> • The first question checks learner understanding of the basic syntax for displaying a variable with a print statement • The second question checks that learners have understood that variables need to be initialized before they are referenced • The third question checks that learners know that a variable can only hold one value at a time and assignment can change the value held in the variable <p>Misconceptions — the questions will help you check whether learners understand that:</p> <ul style="list-style-type: none"> • During program execution, a variable must have been assigned a value before that value is referenced • A variable can only hold one value at a time
--	--

<p>Activity 1</p> <p>(Slides 6–15)</p> <p>15 mins</p>	<p>Introducing inputs</p> <p>Show slide 6 and ask the learners to make a prediction about some code that uses the input function. They should think, write, pair, share their predictions.</p> <p>The next few slides will reveal the answer and steps for the learners to see how the user will interact with the program when it is executed. You could use these slides or add the code into your IDE and talk learners through it.</p> <p>After the demonstration, ask the learners, “How could we use inputs to improve our silly story programs?”. Hopefully learners will respond by asking the user questions like “pick a color” and then incorporating this into the program by using user answers to fill in their variables rather than just setting them at the beginning.</p> <p>Use live coding to demonstrate to learners how to add inputs to their silly stories. Be sure to make little mistakes along the way and ask the learners questions as you go. A teacher demonstration video has been created to support you, which you can view before the lesson. This video is not meant for learners.</p> <p>A link to a premade silly story has been added to the slide deck for you to use if you have not created your own version for this lesson. Learners will have developed their own silly stories, but you could also use this file with learners that may have been absent last lesson.</p> <p>After the demonstration, give your learners some time to add inputs to their own programs and test them out. If learners finish early, then they should test each other's programs.</p>
--	---

<p>Activity 2</p> <p>(Slides 16–33)</p> <p>10 mins</p>	<p>Types of data</p> <p>Use the slides to talk through how data types work in Python. Learners should already be aware that Python doesn't require the explicit declaration of data types in order to use a variable. However, data types are still important in Python because incorrect data types can cause errors.</p> <p>The slides introduce learners to the five main data types that they need to be familiar with and their uses. It then demonstrates how incorrect data types can cause problems with our programs.</p> <p>There is some information on how <code>input()</code> and <code>int()</code> work as functions and how functions can be used to convert the data types of a value. The slides take learners through the input, process, and output for each function to give them some insight into how these work.</p> <p>The slides also briefly cover data validation checks and how these can be used to avoid runtime errors in programs. These will be covered in more detail at a later point in the course.</p> <p>A demonstration of the code has been used on the slides, but you might want to type these directly into the IDE to illustrate them through a short live coding activity.</p>

<p>Activity 3</p> <p>(Slide 34)</p> <p>20 mins</p>	<p>Mini data collection program</p> <p>Give learners the activity sheet, which asks them to predict, run, and investigate a mini data collection program that displays some text based on their answers.</p> <p>The investigation tasks take them through all of the data types that they have learnt about in the previous activity. It also involves some reading of the Python documentation to try and understand how some of the built-in functions work.</p> <p>An explorer task has been provided to challenge learners if required.</p> <p>Model answers have also been provided if you wish to use these for peer- or self-assessment.</p>
<p>Plenary</p> <p>(Slides 35–40)</p> <p>5 mins</p>	<p>Match the data types</p> <p>Ask the class to decide which data type matches the example value given. You could print this as an individual activity or ask for votes on the correct answer.</p>
<p>Homework</p>	<p>N/A</p>