Lesson 4: Randomisation

Introduction

In computer science, random numbers are something that you are likely to use regularly. They are also used in areas such as cryptography, while pseudorandom numbers are used in video games, modeling, and simulations. In this lesson, learners will be introduced to the concept of random numbers using Python documentation. Learners will determine what the random module is capable of, and how random numbers can be generated in Python.

Learning objectives

- Be able to locate information using the language documentation
- Import modules into your code
- Demonstrate how to generate random numbers

Key vocabulary

True random number, pseudorandom number, module, import, library

Preparation

Subject knowledge:

You will need an understanding of why random numbers are important in computer science and how they can be generated using the in-built Python module random. You will also need to be familiar with the location and format of Python documentation in helping to identify what is possible with elements of the language. https://docs.python.org/3.7/

Note: Please be aware that the contents of this lesson relate to the use of Python version 3.7. You are advised to check which version of Python you have installed in your setting and make appropriate amendments to the lesson documentation.

You will need:

- Slides
- Activity 1 worksheet
- Activity 1 solutions
- Another function: A5 activity sheet and solutions
- A4 starter code

Optional:

- Vocabulary sheet

Assessment opportunities

Assessment opportunities exist within each activity from the answers given on the slides, also through discussion around the activities with the class to determine understanding.

Outline plan

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

Starter activity (Slides 2–5) 5 mins	**Roll the dice!** Ask learners why we use dice in real-life games. Make sure discussions lead to the fact that dice allow for a random number to be obtained to determine the number of moves to take in a game (for instance). Show slide 3, which presents the learners with some information regarding rolling a dice, specifically the factors that affect the outcome. Whilst these are possible factors in the physical world, they can't be factored in easily in a computer program. Show slide 4, which asks, "Where might you need to generate randomness in a game?". Give the learners some time to think about an answer. Slide 5 has some suggestions.

Activity 1 (Slides 7-15) 5 - 8 mins	**Introducing random** In this activity, present learners with a piece of code to look at and think, pair, share to predict what the code will do. There is a multiple choice question on slide 7 to determine if they can predict the output of the program correctly. Use the slides to help you explain how this piece of code works. Tell learners that Python has external modules available that are collections of code written by someone else to solve a specific problem. Tell learners that they can make use of these in their programs. To use them, they need to use the import statement, along with the module name at the top of their code, as illustrated in this example. Next, ask learners to run and investigate the code using the activity 1 worksheet. They can also have a go at the explorer task if they are able to. Following this, tell learners that they are not expected to know every single detail of a programming language and that each language has its own documentation to refer to. It's important to emphasize that even seasoned programmers use this documentation as a reference guide when they need to look something up. Encourage learners to get into the habit of using it regularly to improve their knowledge. Share this information using slides 9 and 10.

Activity 2 (Slides 16 - 19) 5 mins	**Truly random vs pseudorandom** Show learners slides 11 and 12, and explain the differences and potential uses of the different types of random numbers. Show learners slide 13 and ask them to spot the differences in the two pieces of code. Essentially, they do the same thing but in a different way. Then show slide 14 and cover the different methods of importing a module vs importing functions from it. The key point here is to highlight that you only need to import what you need. To determine the different functionality of a module, again explain that this is always available in the language documentation.
Activity 3 (Slide 20) 5 mins	**Can you determine what is missing?** In the previous activity, learners were taught about only importing what they need. In this activity learners will import and use the randint function in their code. In this live coding activity, learners have to determine a missing line of code to successfully import the correct function from the module. See if learners can determine what is missing from the options available to them to come to the correct answer. Ask them to type it in to see if they were correct. If not, can they fix it?

	Following this activity, learners will document their code accordingly to show what their code does.
Activity 4 (Slides 21-22) 10 mins	**Check out the comments** Learners will now be shown how to comment code so that it is useful for themselves in the future and anyone else reading their code. From this lesson onwards, learners will be encouraged to comment their code where appropriate in all of their programs. Reinforce the need for good comments in programs in case the original writer of the code is not available to explain it.

Activity 5 (Slides 23-27) 15 mins	**Another function** Now introduce the learners to the function randrange on slide 18. The code can be accessed at randrange or typed directly into Mu. You should then display and explain slide 19, which gives an indication of the three values required in the randrange function. Learners will now execute the code a number of times, making a note of the output on the ‘Another function’ activity sheet. Learners will complete the rest of the activity sheet, which requires them to modify the code and make some predictions as to potential output, followed by some testing of their predictions through executing the modified code. Present learners with the modified version of the code on slide 21 and pose the question, “How will the changes in values affect the output?” . In pairs learners should discuss the question based on the code shown on the screen. The answers and examples of the potential output of the program are on slide 22.
Plenary (Slides 28-29) 5 mins	**Pair the parameters** Present the learners with the five statements including the randrange function and various parameters, along with potential outputs. Learners have to match the right expression up to the right output. The answers are on slide 24.