| **Session Objectives** | * Understand and use For Loops * Use the Range function * Some mathematical functions that work with lists * Slice and Copy lists * Understand Tuples vs. Lists |
| --- | --- |
| **Key Points** | * What is a For Loop, what is the correct syntax, how and when is it used? * How can the range function be used to make code more efficient? * What functions and methods may be used with lists? * How do you use slices to copy all or part of a list and why would you? * What is a tuple and how does it compare to a list? |
| **Assessment** | * Via practical challenges - see task sheet below |
| **Instructor Prep** | * **Note**: These session plans include sections where long explanations are provided simply because it is important to ensure that these tricky concepts are communicated as clearly as possible. However, it is not expected that the instructor will recite these verbatim, feel free to use your own refined and well-honed approach as long as the learning point is covered. The provided narrative is always available if needed. * **Learning and delivery** can be more effective if resources and tasks are personalised. If time permits, feel free to update slides with your own examples such as replacing cat examples with your own pets, or anything else. |
| **Materials** | * [TIFC1-PF-4 - Working with Lists - Slides](https://docs.google.com/presentation/d/12thS9EatJOurPSfCwm1kIAlmM7GJSZkRy3ViC77rhqQ/edit?usp=sharing) * [TIFC1-PF-4 - Working with Lists - Tasks](https://docs.google.com/document/d/1B9bxzBzHnWYctHqVOUYHp9l4hYPd-_MRY-2eOMMr3J0/edit?usp=sharing) * [TIFC1-PF-4 - Working with Lists - Solutions - INCOMPLETE](https://docs.google.com/document/d/1vlMvUyLhEbRrwdmjtk2KgDVC2q9TrZcUIaCshVKfM8E/edit?usp=sharing) |

| **Time** | **Activity** |
| --- | --- |
| 5 mins | **Slides 2 - 3 - Objectives**   * **Show Slide 2 - Read** session objectives * **Show Slide 3 - Ask** for a volunteer to **explain** what the code in the example does. If some positive responses are forthcoming, thank the contributors then **explain** if not already covered:   “*This code starts with a list just like we’ve already used, then a string is printed on line 3, then we initiate a For loop which is a very useful and common feature available in many coding and scripting environments*.” |
| 15 mins | **Slides 4 - 5 - For Loops**   * **Show Slide 4 - Read** through points and **explain** that:   “*A For Loop* ***iterates*** *through a given input, in this case a list (but it can iterate through other things). One reason this is so useful is because we often want to do things repeatedly, we may need to carry out the same actions or steps against a number of objects. Although, sometimes it is not clear that this is what we’re doing.*”  **Explain** This is where you have to start trying to think like a developer. For a simple example, consider an app that represents a store:   * Your available stock is all stored in a list * Add a text box for customers to search for items * Use a For Loop to compare the requested item against all of the available items in the list one by one. * In the loop you write some code for what should happen when the match is found or not   **Advise** that:  “*In reality this isn’t the best example of how you would create an app for a store, you’d want a dedicated database on the back-end at least, but the loop concept is valid.*”   * **Explain** that the basic syntax isn’t tricky, but there are two important concepts:   + Because the For Loop is typically written independently from the items it’s going to loop through, it requires a **temporary variable** that takes the value of each successive item. This variable’s value will update as it processes item 1, 2, 3, and so on.   + The next important syntax point is **indentation**, this is where you leave a fixed empty gap at the start of each line, typically a single press of the ‘tab’ key for each level of indentation (although some people use 2x standard spaces). Indentation is how the Python interpreter understands what lines of code should run in the loop, and which parts are outside of it. In this case lines 6 and 7 will run on every pass through the loop, however, line 9 is not indented, so this will only run once the For Loop has finished.   + Note - You don’t need to tell the loop to go back to the top, it knows to keep restarting until every item has been processed (although you can stop it early such as if a condition is met). * **Show Slide 5 - Ask** or **Call Out** whether any learners can identify these syntax elements in the code example:   + **Identify** ‘food’ on line 5 as the temp’ variable which will take each value from the list     - **Explain** to learners that the name of this temporary variable should be carefully checked. It is common to use a singular noun when a plural noun is the name of the list, such as *food* and *foods*, or *book* and *books* which can make it tricky to spot the difference when reviewing the code.   + **Identify** consistent indentation on line 6 & 7 showing these lines should run upon each iteration of the loop   + **Point out** that the lines of code running within our loop are just the same ones we’ve already explored |
| 5 mins | **Slides 7 - 12 - Pseudo-Code**   * **Show Slide 7 - Ask** for a volunteer to write some pseudo-code in the chat representing a For Loop which will iterate through a list of the dogs on the slide - make up some names. The code may simple print the names out, or do something more advanced.   Explain that:  “*Pseudo-code is just an approximation of the logic and structure of the final solution. It is not concerned with accurate syntax, partly because you may not have decided what language you’re going to use, but you can still include features that are common across all languages like variables, loops, containers such as lists, operators, etc.*”  When a pseudo-code example is provided, **ask** the contributor to **explain the logic** of their code, thank them and **show Slide 8** which is a simple implementation. |
| 15 mins | **Slides 9 - 12 - Numerical Lists and the range function**   * **Show Slide 9 - Ask** for a volunteer, or select an individual, to explain the code on slide.   *If necessary, remind the cohort that explaining code in plain English is an important skill to practice*.   * **Ask** if anyone can think of a drawback to the approach on the slide?   + Prompt: Is this the best way of making a loop run a fixed number of times?   + Prompt: Is this the most efficient way of writing code?   + Prompt: What if you needed to run your loop 100x or 10,000x? * If not forthcoming, **advise** that:   *“Needing to create a loop which will run a set number of times, rather than once for each item or variable, is quite common. Creating a list of numbers will work, but isn’t very efficient, consider what a list of 1 - 10000 (or more) would look like in your code!”*   * **Show Slide 10 - Explain** that:   “*The range function is a more convenient way of meeting this requirement;*  *In this example we can see that the syntax and structure of the For Loop is basically the same, except we don’t have an external list or input to iterate through, instead the range function generates the numbers to loop through instead. All we provide is the first and last numbers in our desired range - with one caveat, the range stops at the last number, it doesn’t include it. This is known as ‘Off by one’ and is because the computer starts counting at 0, humans start at 1 (remember index number 0? Same thing).*”   * **Show Slide 11 - Ask** for a volunteer or select an individual to explain what they think is happening in this code snip. If accurate responses are forthcoming thank the contributor, else explain:   “*The range() function can be used independently to a For Loop, in this case it’s combined with the list() function which will create a list for you rather than having to type it out, so it will automatically create a list of numbers based on the range you specify, which can then be used and manipulated just like a manually created list.*”   * **Show Slide 12 - Ask** for a volunteer or select an individual to explain what they think is happening in this code snip. If accurate responses are forthcoming thank the contributor, else explain:   “*We can add one more instruction to the range function, which will is an increment. By default the increment is one, but you you can specify any increment you want*” |
| 5 mins | **Slides 13 - Functions That Work With Lists**   * **Show Slide 13 - Explain** that these are three mathematical functions that can be useful when working with numerical lists allowing you to quickly find the largest, smallest or return a sum or all numerical items.   Click on the link in the slide (<https://www.w3schools.com/python/python_ref_list.asp>) to remind of some of the methods already seen, and some others we’ve not used. |
| 15 mins | **Slides 14 - 17 - Slicing and Copying**   * **Show Slide 14 -** Reintroduce your ‘scenario’ * **Show Slide 15 - Ask or select** someone to try and explain what is happening in the code snip. If accurate responses are forthcoming thank the contributor, else explain:   “The syntax in this print statement on line 3 will return a slice of the specified list. The numbers either side of the colon refer to the index numbers of the items in the list.”  Ask “What do you think will be returned if this code is run?” When an answer is provided copy/paste code into VSC and verify. Explain:  “*The slice starts at index number 2 i.e. Mackerel, then stops at number 3 without including item 3. So the only output is Mackerel, but you can output whatever slice you like.*”   * **Show Slide 16 - Ask or select** someone to try and explain what is happening in the code snip. If accurate responses are forthcoming thank the contributor, else explain:   “*In this example our For Loop iterates through a slice of the original list, effectively a temporary copy.*  *In this case it’s a slice of the whole list, but you might just want to loop through a smaller section.*”   * **Show Slide 17 - Ask or select** someone to try and explain what is happening in the code snip. If accurate responses are forthcoming thank the contributor, else explain:   “*Rather than a temporary slice of a list, which is used once for looping then discarded, here a variable is declared and it’s value is a list of foods. On line 2 a second variable is declared and it’s value is a slice of the first list.*  *Note the syntax of this slice, if you don’t give index numbers it will go from the very beginning to the very end, regardless of the list’ length. The rest of the code then demonstrates how you can work with each list individually.*” |
| 5 mins | **Slides 18 - 19 - Working With Tuples**   * **Show Slide 17 - Ask or select** someone to try and explain what is happening in the code snip. If accurate responses are forthcoming thank the contributor, else explain:   “*Tuples are very similar to lists, they behave very similarly, and many of the same methods can be used with them, we simply declare them with parentheses rather than square brackets.*  *They do have one important difference vs. lists though - they are immutable, which means they cannot be changed once declared.*  *So, you can access and read the items, you can loop through a Tuple, but you cannot add, remove, or change an item within a Tuple.*   * **Show Slide 17 - Advise** that: “*You can however re-declare it if you need to - effectively overwriting the tuple*” |
| 55 mins | **Slide 21 - Hands-On Challenges**   * **Share link** to task sheet, instruct learners that they now have 55 minutes to work through the task sheet. Once the time is up they should take a screenshot/snip of their last completed challenge and submit it on Canvas. * **Open breakout rooms** - instructor to select number/mix |