| **Session Objectives** | * Identify IF statements and how they are used. * Identify correct syntax and the structure of if-elif-else to carry out multiple tests * Identify Boolean expressions * Identify comparison operators * Demonstrate IF statement with Lists and Loops |
| --- | --- |
| **Key Points** |  |
| **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. |
| **Assessment** | * Via practical challenges - see task sheet below |
| **Materials** | * [TIFC1-PF-5 - If Statements - Tasks](https://docs.google.com/document/d/1D4sjgpBNfLq7RzDogJcR-9VF6l6aPr4lEM-uWeDYBmI/edit?usp=sharing) * [TIFC1-PF-5 - If Statements - Slides](https://docs.google.com/presentation/d/1mPIwQO6wGzM-_fuDYKS8CzLV_S5yajx1Zu4KYs1fteo/edit?usp=sharing) |

| **Time** | **Activity** |
| --- | --- |
| 5 mins | **Slide 2 - Objectives**   * **Show Slide 2 - Read** session objectives |
| 10 mins | **Slide 3 - 4 - Introduction to IF statements**   * **Show Slide 3 -** Reintroduce your ‘scenario’ * **Show Slide 4 - 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 by declaring a variable which has the value of a list of foods, then a second variable which uses a new function called input(). This function will both print a message to the user, typically a question, and wait for the user to ‘input’ a response. The response is then passed through as the value of the variable.*  *Input is a simple but very useful function, which we will discuss further in a future module*  *The code then goes on to use an if statement which is a very common and powerful code structure that is used in all coding and scripting languages. At it’s most basic, an If statement allows you to carry out a test or check in your code, then provide different outcomes based on the result of the test*  *We’ll come back to this code later*” |
| 10 mins | **Slides 5 - 8 - Exploring If statements**   * **Show Slide 5 - Ask or select an individual** whether they have heard of boolean and what it means. If positive responses are forthcoming, thank the contributors then **explain** if not already covered:   “*Boolean values are True or False values, sometimes you might want to assign these values to a variable. Boolean Expressions are the result of a true or false test, such as* ***‘Is A equal to B’*** *or* ***‘Is X bigger than Y’****. The answer to such a test is simply yes (True) or no (False)*.”   * **Show Slide 5 - Explain** that if statements include a condition or test, and then an operation to carry out when the condition is true. If the condition is false the if statement may do something else, or nothing. * **Show Slide 7 - Explain** that these are some examples of common comparison operators, they compare one value against another, and the outcome is either true or false. Some of the comparisons may not apply to certain data types, one string isn’t greater than another for example. But you could certainly verify whether one string is equal to another, such as verifying a password entered by a user is equal to the correct password you have stored. |
| 10 mins | **Slide 9 - In or Not In**   * **Show Slide 9 - 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:   “*Hopefully everyone can see that this code snippet starts with a list of foods, then declares a string variable. After that we have an if statement using a very simple conditional test ‘not in’ which does exactly what you would expect. It returns true or false based on whether the value in the ‘unhappy\_foods’ variable is found within the ‘foods’ variable*”   * **Advise** learners to try experimenting with the in/not in comparisons when trying out the code - Prompt: try using it with strings and string fragments. * **Explain** that there are a couple of extra points to notice in this code snip, the first is simply the colon at the end of the if statement, this is a syntax requirement, and is often missed.   The next point is the indentation; Your if statement includes a conditional test, and the next indented line is the code that runs when the test is true. If the test is false this line is skipped. |
| 15 mins | **Slide 10 - If-Elif-Else**   * **Show Slide 7 - Advise** that this is the same code we saw at the beginning. **Ask** for a volunteer to **explain** in more detail what is happening from line 5. If some positive responses are forthcoming, thank the contributors then **explain** if not already covered:   “*We know that the ‘answer’ variable is captured from the user by the input() function, but one challenge that you need to consider as a developer is accounting for different user behaviours. What if they use a capital letter when typing their response? Python is case sensitive, so it might not match the items in the list.*    *To avoid this error the lower() method is used to return just the lowercase version of whatever value is in ‘answer’, and this is compared with the string ‘tuna’ which is also in lower case.*  *If this comparison is true, then Python will run the next indented line, which is line 6. If the comparison is not true then line 6 is skipped and line 7 runs.*”   * **Explain** that:   “*Elif is short for ‘Else If’, or to think of it a different way, remember, Python is known as a high-level programming language which means it’s designed to be similar to how people think and talk. Imagine a friend needed to cook a meal for someone and they don’t know what to cook, they ask you for advise, so you say the following:*  *If they like Italian make lasagne*  *Else, if they like Spanish make Paella*  *Else, if they like Chinese make a stir-fry*  *Else, if they like Mexican make Tacos*  *Else make them fish and chips*”   * **Explain** to learners that this is the logic of Elif, it combines Else and If together so that you can add multiple levels to your if statement, with another conditional test for each one. * Before moving on, **ask** learners if they can **identify** any **problems** or **improvements** to this code? If any valid responses are forthcoming thank the contributors, if not identified ask the following: * Is this code repetitive?   + **Answer**: It does look repetitive, lines 5, 7, 9, & 11 look like they do similar tests. Sometimes this is unavoidable, but often you can use a different strategy in your code. * How might we avoid repetition?   + **Answer**: Use lists and loops * Could there be a problem in the future with the print statements?   + **Answer**: Over time the list might need to be updated, which means the items could change, or their positions could change. If this happens hard coded index numbers might result in the wrong items being compared. |
| 15 mins | **Slide 11 - Using if statements with loops and lists**   * **Show slide 11 -**  **Explain** that this code isn’t exactly the same as the previous example, but very similar. **Ask** for a volunteer to **explain** what the key differences in logic are. If any valid responses are forthcoming thank the contributors, if not explain the following:   “*In this case the list of foods is a bit different, then a For loop is used before the if statement. Remember, a For loop will iterate through the input, in this case the list of foods. The For loop requires a temporary variable, called just “food” here, and this will take the value of each individual item as the loop iterates.*  *At line 4 our If statement is indented into the For loop, which means it will run on every pass, this avoids having to write the same if statement over and over.*  *For this example the If statement is testing whether the temporary variable ‘food’ is ‘equal to’ the string “chicken”; How could you change this code to look for whatever food the user wants? Don’t answer now, see if you can code it yourself when you start the next task.*  *If the “food” variable does match ‘chicken’ then like 5 runs and prints out a string, but if they don’t match line 5 is skipped and the next line indented within the For loop runs, line 6 the Else statement. If statements typically end with an Else statement, and there can be as many Elif’s in between as you want. The Else is actually optional if you don’t need a final action or exit message, the If statement will simply end when there are no more lines of code to run.*  *The last few lines then, if the temporary variable does not match “chicken” then the Else statement runs, which runs line 7, and the value of the temporary variable is embedded into an f-string. This also simplifies the previous example, because it doesn’t rely upon recalling an item from the list by a specific index number which could change.*” |
| 5 mins | **Slide 12 - Using if statements with multiple lists**   * **Show slide 11 -**  **Explain** that this code demonstrates one way you could test for more than one match, i.e. not just ‘chicken’. The For loop at line 4 iterates through the list of ‘available\_foods’, then within each iteration our if statement on line 5 will verify whether the temporary variable exists within the list of ‘disliked\_foods’. We don’t need to loop through the second list, the ‘in’ keyword will take care of that for us. |
| 50 mins | **Slide 13 - Review**   * **Show slide 13 -** Explain that returning to the original code example, we’ve not explored a number of different ways to use lists, loops, If statements, and For loops. We can provide variables to our code, or capture them from the user, and we’ve mixed all of these features together. * It is recommended that you take this code example we’ve seen a couple of times now, and try to rewrite it to take advantage of the improvements we’ve discussed, before you move onto the additional practical tasks for this module. * Share the following task sheet with learners: [TIFC1-PF-5 - If Statements - Tasks](https://docs.google.com/document/d/1D4sjgpBNfLq7RzDogJcR-9VF6l6aPr4lEM-uWeDYBmI/edit?usp=sharing) |