| **Session Objectives** | * Explain what a variable is and why they’re used * Be able to identify a number of variable types * Select the appropriate variable type for a given scenario * Store and recall data using a variable |
| --- | --- |
| **Key Points** | * Working with Variables * What is a string * Using strings in Python - concatenating and using string methods * Using numbers in Python - integers, floats * Understanding and avoiding TypeErrors |
| **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** | * [Variables Strings and Data Types - Slides](https://docs.google.com/presentation/d/1XsnVfWNYuDzcABOHxv0T5v4nrfvWT64v5HKpLh4dAs4/edit?usp=sharing) * [Variables Strings and Data Types - Tasks](https://docs.google.com/document/d/1EDHs0M_av2piUgvujJTNb045EOcKhWd_lzDltcCAjD4/edit?usp=sharing) * INSTRUCTOR ONLY - [Variables Strings and Data Types - Example Solutions](https://docs.google.com/document/d/1CRBGyC8ORdCE-7SOu_lWdSP0okaRKbYiprwmr3GOWKo/edit?usp=sharing) |

| **Time** | **Activity** |
| --- | --- |
| 5 mins | **Slide 2 - 3 - Hello World**   * **Show Slide 2 - Ask** learners to recall the line of Python code they typed previously to verify that VSC was working - print(“hello world”)   + **Optional - demonstrate** above code if there has been a break since previous session, not necessary if it was only a few mins ago. * **Show Slide 3 - Explain** that:   “*This is basically programming.* *You’ve chosen a programming language, then using appropriate syntax for that language, written a line of code which can be interpreted and followed by the computer*”. |
| 10 mins | **Slide 4 - Learning Objectives**   * **Read Slide 4 - Ask** learners to call out if they can explain how to meet and of the objectives already. If some positive responses are forthcoming, thank the contributors and reinforce to the group the benefits of supporting each other by explaining the following:   “*We’re trying to simulate a professional environment, and it’s common for there to be some people who are more experienced in some areas. It’s good to know who these people are so that you have another support option.*  *Supporting each other benefits both parties; The person seeking support improves their understanding - instructors don’t take it personally if someone else is able to explain something in a better way that clicks for you. But the person providing support also benefits from practising their communication skills, particularly technical communication, which is definitely going to be a big part of your future career.*  *If someone does support you effectively, be sure to thank them, let them know they’re good at it, because after our school career there aren’t many occasions when someone lets you know what you’re good at*.” |
| 10 mins | **Slide 5 - 9 - Introducing Variables**   * **Show slide 5 - Introduce** your ‘variable’ Noche (or your alternative) * **Show slide 6 - Ask** any learners who have Python or programming experience if they can explain the code snip.   + If so, listen, clarify, correct as necessary, and thank contributors. Don’t dwell on ‘title method’ for now.   + If no learner has volunteered an accurate explanation don’t explain now, return in a couple of slides * **Show slide 7 - Read** sentence, and offer further context by providing real-world or theoretical examples that learners may be familiar with or be able to imagine, such as:   + An alarm clock might use a variable to represent the time you want to wake up, when the real time matches the variable the alarm goes off.   + A modern car’s radio doesn’t have a dial to tune it, it remembers what station you last listened to. When you turn the car off this might be stored in a variable.   + When you turn your home heating up or down it may store your target temp in a variable, then turns the boiler off or on until the ambient temperature reaches your target. * **Ask** learners if they can think of any more real-world examples for using variables, whether it really works that way or not is irrelevant. * **Show Slide 8** - same as slide 6, if not already done so, **explain** the logic and syntax of the code example. Key points:   + Begins by declaring a variable, and assigning a value of “Noche”   + Utilises print statement to display information to the user   + Line 3 prints a short string, adds another string to it by recalling the value of the variable, then appends a third string to form a complete sentence - this is also known as ‘concatenation’   + In addition to recalling the value of the variable, we’ve also changed the formatting of the string using a method - more on these shortly.   + Line 4 prints one longer string with no external values needing to be concatenated.   + We don’t need to declare the end of our code, Python just interprets each line in turn and stops when it gets to the end. * **Show Slide 9 - Read** the points regarding appropriate variable names. |
| 5 mins | **Slides 10 - 15 Variable Errors**   * **Show Slide 10 - Read** the types of errors that are commonly encountered when developing Python code * **Show slides 11 - 15 - Point out** the error in each example, and the corresponding error generated. Explain to learners:   “*When you first start learning about code, or lots of things, it can take a while for your eyes and brain to start to make sense of what you’re seeing, for a little while it just looks like gibberish. However, with time and familiarity you should start to be able to pick out some relevant info. Such as, in these cases, the type of errors encountered, and the location of the errors.*  *So, in your own code, don’t dismiss the errors, review them, identify the type of error, and look at the location it points to.*” |
| 10 mins | **Slide 16 - 22 - Introducing Strings**   * **Show Slide 16 - Read** key points regarding strings * **Show Slide 17** - same as previous example - **Call Out** a learner to identify the string.   + If not done so already, explain the function of the title method:     - Identify the initial variable stored in lower case     - The correct syntax to correctly apply the method to the variable     - Show the output String in Title case       * Code provided in Presenter’s Notes of slide to copy/paste if needed. * **Show Slide 18 - Read** the additional string methods and their effects * **Show Slide 19 - Point out** the title method again, switch to code in VSC and change the title method to the upper method, run, and demonstrate the new output. |
| 5 mins | **Slide 20 - 21 - Concatenating**   * **Show Slide 20 - Explain** that “Building a long string output from lots of smaller string-pieces and variables is quite a common task, referred to as concatenating”.   + **Read** bullet points * **Show Slide 21, switch to live code** in VSC. Add example escape characters into your string, run, and demonstrate the new output. |
| 5 mins | **Slide 22 - F Strings**   * **Show Slide 22 - Explain** that F Strings are a more modern way of writing strings. You can embed your variable calls and methods into your strings without having to break them into pieces.   + Identify the ‘f’ preceding the quotes enclosing the string   + Identify the curly brackets enclosing the variable call and method   + Point out that the plus signs to build up the pieces of the string isn’t required |
| 5 mins | **Slide 23 - Whitespace**   * **Show Slide 23 - Read** bullet points, rather than demonstrate everything, advise that learners will shortly be trying out this small snippet of code, so they’re encouraged to try and experiment with whitespace, as well as methods and everything else we’ve discussed. In this case they could add additional whitespace to their stored variable, confirm the impact on the output, then use the strip methods to correct it. |
| 5 mins | **Slide 24 - 25 - Comments**   * **Show Slide 24 - Explain** the following:   “*The majority of the roles you’re seeking will require you to work with text files for speed, accuracy, and convenience. For repetitive tasks, instead of slowly navigating a GUI over and over again, you can write a text file once and reuse it over and over again instantly. It’s same with code, you write your code once, then copy it for reuse and sharing.*  *Because you’ll be using many different complex text files, some of which you’ll create yourself, many made by other people, they all provide a way of adding comments. You can use them to provide guidance and info to the next developer who looks at your code - perhaps months or years later.*  *Some organisations will provide a standard ‘comment template’ which you’re expected to add to the top of all code files which includes info like the date, time, developer name, department, project, etc.*”   * **Show Slide 25 - Point out** the example comment |
| 20 mins | **Slides 26 - 27 - Try it Out**   * **Show Slide 26 - 27 - Advise** learners that they have 20 mins to try this sample code. It is recommended that they:   + Copy the code as is into their own VSC, save and run, confirm that the output is as expected   + Rewrite the variable value and sentences, ensure that your code still runs and the output is as expected   + Try using different methods   + Try adding whitespace to the variable and stripping it with methods   + Try re-writing the code to use f-string syntax * **Open breakout rooms** - instructor to select number/mix |
| 5 mins | **Slides 28 - 29 - Numbers in Python**   * **Recall** learners - **Advise** that we have a couple of extra points to cover before they work on some challenges. * **Show Slide 28 - Read** points regarding how Python handles numbers * **Show Slide 29 - Read** points regarding differences between integers and floats |
| 5 mins | **Slides 30 - 32 - TypeErrors**   * **Show Slide 30 - Read** points regarding TypeErrors, demonstrate through VSC using code examples in Presenter Notes. * **Show slides 31 - 32 illustrating** another TypeError example |
| 60 mins | **Slide 33 - Hands-On Challenges**   * **Share link** to [task sheet](https://docs.google.com/document/d/1EDHs0M_av2piUgvujJTNb045EOcKhWd_lzDltcCAjD4/edit?usp=sharing), instruct learners that they now have 60 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 |