Teacher Training Notes

Session 3

Slide 1 – Review

* Let’s have a refresh of last week’s lesson.

Slide 2 - If / Else / Elif

* These are the conditionals that if true will execute some code.
* If you want more than one if statement, you use elif statements and an else statement to end it.
* In this example, the first statement is checked, does the variable name have an equal value to Alice, it doesn’t.
* Then the second statement is checked, does the variable name have an equal value to Alice, it does, so Hello Bob is printed.
* As the elif statement is satisfied, the else statement is not checked,

Slide 3 - And, Or, Not

* You can combine conditional with the and, or or not statements.
* In example 1, what would be printed if the age was 13? You are a teenager.
* In example 2, what would be printed if the age was 19? Nothing.
* In example 3, what would be printed if the age was 16? Nothing.

Slide 4 – List

* In Python, there are different ways to collect data. One of the ways is through a list.
* Lists are strings, captured in square brackets, separated by commas.
* You can use indexing to print one item from a list.
* To add to a list, you use append.
* To change an item in a list, you state the name of the list, the index of what you want to change and the new value.
* To delete an item in a list, you write del then brackets and the name of the list with the index of the item you want to delete.
* You can use an if statement to check if something exists in your list.
* You can also use a for loop to print your list.

Slide 5 - For Loops

* A for loop iterates through a collection of items.
* This for loop will go through each name in the list and print them out, one after the other.

Slide 6 - Ranges

* You can use a range to determine how many times you want a loop to run for.

Slide 7 - For Loops

* This example shows the step range being used to print all the years the Olympics have run.

Slide 8 - Questions?

* Does anyone have any questions from last week’s content?

Slide 9 – Modules

* A module contains reusable code.
* It allows you to make use of code already created, instead of having to reinvent it all.

Slide 10 – Modules

* To use a module, you must first import it.

Slide 11 – Modules

* You can do this either by using import module to just import the whole module.
* If you just need one function from the module, you can also do that by stating from the module import just the function you want.
* Then under that you can write the code that will then use the module or function.

Slide 12 - Random Module

* The random module allows you to generate random data (integers, floats and strings).

Slide 13 - Random Module

* In this example, first the random module is imported.
* When calling a function from within a module, you must type the module name, a dot, and then the function.
* Then the function random is used, this will pick a random float from 0.0 to 1.0.
* The function randint, will pick a random number from the range you provide, this one will pick between 1 and 10.

Slide 14 - Math Module

* The math module provides access to mathematical functions.

Slide 15 - Math Module

* In this example, from the math module we have imported the floor and ceil functions.
* The floor function will round any number down (to the floor) to one digit.
* The ceil function will round any number up (to the ceiling) to one digit.

Slide 16 - While Loops

* A while loop continues to loop while a condition is True.

Slide 17 - While Loops

* In this example, a variable guess has been assigned with a None value.
* While the guess does not equal 4, the loop will continue and ask the user to input a number, this will update the guess variable, until the guess equals 4 and then the loop will break.

Slide 18 - While Loops

* The syntax for a while loop is while and then your condition and then what code you want to be run inside the loop.
* This code will continue to be executed whilst the condition is true.

Slide 19 - While Loops

* In this example, we set the variable times\_in\_loop to 0 to start with.
* While this variable, times\_in\_loop is less than or equal to 10, print (“Hello”).
* Every time the loop is run, it adds 1 to the value of times\_in\_loop.
* This will keep incrementing until it gets to 10 and this will then make the condition, times\_in\_loop is less than or equal to 10 false and the loop will break.

Slide 20 - Infinite Loops

* An infinite loop is a loop that never ends, it never breaks out of the loop.
* The loop gets executed forever, unless the program is terminated.

Slide 21 - Infinite Loops

* As True is always True, this loop will loop forever.

Slide 22 - Break Statements

* To terminate the loop you are running, you can include a break statement.
* It will then look at the next piece of code to execute.
* This can be used on both for and while loops.

Slide 23 - Break Statements

* While ordinarily this loop would continue for ever, a break statement will stop the loop running.

Slide 24 - Coding Time - Section A

* Now, it’s your time to try some coding, please refer to Section A on your worksheets.
* You’ll have 20 minutes on this.

Slide 25 – Collections

* Last week, we learnt about a list which is a type of collection.

Slide 26 - Collections

* The main types of collections in Python are:
* Lists, tuples, sets and dictionaries.

Slide 27 - Collections — Tuple

* A tuple is a collection of data which are enclosed in round brackets.
* The items in a tuple can be called in the same way as a list.

Slide 28 - Collections — List Vs Tuple

* A tuple is the same as list except you can't change it after creation.
* You cannot add or remove from a tuple.

Slide 29 - Collections — Set

* A set is a collection of data which is enclosed in curly brackets.

Slide 30 - Collections — Dictionary

* In this course, we will be focussing mainly on lists and dictionaries.
* A dictionary can be thought of as an unordered set of key value pairs.
* You can look up values by using a key instead of an index.
* The order doesn’t matter in a dictionary.
* To create a dictionary, you state the name of your dictionary and within curly brackets you add your key in quote marks, a colon, and then the value in quote marks.

Slide 31 - Dictionary — Append

* To add a key/value pair to a dictionary, you state the name of the dictionary, square brackets with the new key in quotes with an equals sign and the new value in quote marks.

Slide 32 - Dictionary — Change

* To change a key/value pair to a dictionary, you state the name of the dictionary, square brackets with the current key in quotes with an equals sign and the new value in quote marks.

Slide 33 - Dictionary — Delete

* To delete a key/value pair to a dictionary, you state del, the name of the dictionary in round brackets and the key in square brackets in quote marks.

Slide 34 - Dictionary — In

* You can use an if statement to check if a key exists in your dictionary.
* What would be printed if the following example was run? Nothing – there is no material key in the dictionary.

Slide 35 - Dictionary — For

* You can use a dictionary in a for loop.
* The following example will print the key and the value concatenated together.

Slide 36 - Collections

* This is an overview of the different types of collections.

Slide 37 - Coding Time - Section B

* Now, it’s your time to try some coding, please refer to Section B on your worksheets.
* You’ll have 20 minutes on this.

Slide 38 - Nested Collections

* A nested collection is a collection within a collection, for example, a collection of lists within a single list or a number of dictionaries within a single dictionary.
* They are also referred to as list of lists or a list of dictionaries or a dictionary of lists etc.
* Nesting can be of great use, as the kind of information we can model is programs can be expanded greatly.

Slide 39 - Nested Collections

* So what do you think will happen if we run the following nested lists?
* There are two for loops, the outer loop will run first and pick the first row, the inner loop will then run and will print each number out in a row.
* Once the inner loop has finished, the outer loop will take over and shift to the next row.
* This will trigger the inner loop which will print each number out and this will continue until all the characters in each row are printed and then the loop will end.

Slide 41 - List of Dictionaries

* This is a list of dictionaries.
* This for loop will go through each dictionary within the list.
* Look at the for loop, what would be printed? Bob – as he is the only one with phone within the dictionary.

Slide 42 - Adding names

* This example, is a dictionary held within a list.
* While fname doesn’t equal a blank space, the program will continue to ask you for a first name and last name and will add that to a dictionary.

Slide 43 - Coding Time - Section C

* Now, it’s your time to try some coding, please refer to Section C on your worksheets, this is also on the board. The example in the coding examples on the worksheet should help with this.
* You’ll have the rest of the lesson to finish this as well as any other questions.