Teacher Training Notes

Session 4

Slide 1 – Review

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

Slide 2 – Modules

* A module contains reusable code.
* It allows you to make use of code already created, instead of having to reinvent it all.
* To use a module, you must first import it.
* 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 3 - Random Module

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

* The math module provides access to mathematical functions.
* 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 5 - While Loops

* A while loop continues to loop while a condition is True.
* 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 6 - 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.
* As True is always True, this loop will loop forever.

Slide 7 - 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.
* While ordinarily this loop would continue for ever, a break statement will stop the loop running.

Slide 8 – Collections

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

Slide 9 –

* Last week we looked at tuples, which were similar to lists, except they were in round brackets instead of square brackets.
* A tuple is the same as list except you can't change it after creation.

Slide 10 - Collections — Set

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

Slide 11 - Collections — Dictionary

* In this course, we focussed 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.
* Like a list, you can add, change or delete key value pairs in a dictionary.
* You can use if statements with them and loop through them/

Slide 12 - 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.
* 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 13 - 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 14 - Questions?

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

Slide 15 – Functions

* A function is a block of code which only runs when its called.
* You can pass it data, known as parameters into the function.
* A function can return data as a result.
* We have already come across lots of functions, such as print, length, upper, lower.

Slide 16 - Functions — Create

* In Python, a function is defined using the def key word followed by round brackets, that can be left empty or have a parameter passed through it and then a colon.

Slide 17 - Functions — Create

* This is the syntax for a function.
* You should name a function like you would a variable, lowercase, no punctuation, underscore instead of spaces and have the name explain what the function does.

Slide 18 - Functions — Call

* To use or call a function, you type the name of the function followed by the round brackets.
* So In this example, if you look at the code within the function, it will print “Hello World”, so if you call the function, at the bottom of the example, this will run the code within the function and print “Hello World”.

Slide 19 - Functions — Parameters

* As stated earlier, parameters are data we can pass the function.
* In this example, in the brackets we have defined a variable ‘name’, so anything we pass in the brackets when calling the function will be passed to the name variable within the code.
* So as the example shows, when calling the function and adding the different names, it will print “Hello Alice”, “Hello Bob” and “Hello Charlie” concurrently.

Slide 20 - Functions — Parameters

* This example shows that you can pass multiple parameters to your function, this one will take a name and age and put that into the code.
* So when the first one is called, what will it print?
* Hello Alice. I’m 22 years old. In 10 years time, I will be 32.

Slide 21 - Functions — Parameters

* This function will take three parameters, and multiply them together.
* It then casts the answer to a string so it can be concatenated with a string.
* So what will be printed in these two examples?
* The area is 144
* The area is 840

Slide 22 - Functions — Parameters

* Here is the syntax for including parameters to your function.

Slide 23 - Coding Time - Section A

* Now, it’s your time to try some coding, please refer to Section A on your worksheets.
* You’ll have 30 minutes on this.

Slide 24 - Functions — Returning

* You can have your function return a value, unless stated this will not be printed.
* You can use the return data in your code to input into other variables in your code.

Slide 25 - Functions — Returning

* You use the word return in your code to get a valued return.

Slide 26 - Functions — Single Job

* Best practice for functions is that each function should have a single job.
* Here, we have split up the previous function into two different ones, one that accepts your name and age, the other purpose is to return a value that is then input back into the first function.
* Single jobs make it easier if you need to update the code, as it will be easier to pinpoint the area that needs updated.

Slide 27 - Functions — Recursion

* Recursion is a mathematical and programming concept.
* It means that functions calls itself.
* This has the benefit of you being able to loop through data to reach a result.
* In this example, recursion is used to call the calc\_factorial within itself.

Slide 28 - Functions — Recursion

* This is a break down of how it works.
* So the first the function is called, it takes the number 4 as a parameter, as this does not equal 1, it goes to the next code to execute, which is the number 4 multiplied by calc factorial 4 minus 3, and this keeps looping until x equals 1 and stops the loop.
* This is then input into the print statement, which would print ‘The factorial of 4 is 24’.

Slide 29 - Coding Time - Section B

* Now, it’s your time to try some coding, please refer to Section B on your worksheets.
* You’ll have to the end of the session to work on these.