4d 61 6b 65 20 65 76 65 72 79 74 68 69 6e 67 20 61 73 20 73 69 6d 70 6c 65 20 61 73 20 70 6f 73 73 69 62 6c 65 2c 20 62 75 74 20 6e 6f 74 20 73 69 6d 70 6c 65 72 2e

A Level Computer Science

Preparation – Bridging Lesson 1

Objectives

- Introduction to the sessions
- Programming and Python Idle
- Python Programming part 1
  - The IDE (shell and editor)
  - Print statements (print)
  - Variables and data types in programming
  - Input statements (input)
  - Calculations
  - Exercises
- Useful Links - Python
  - Data Types
  - Variables

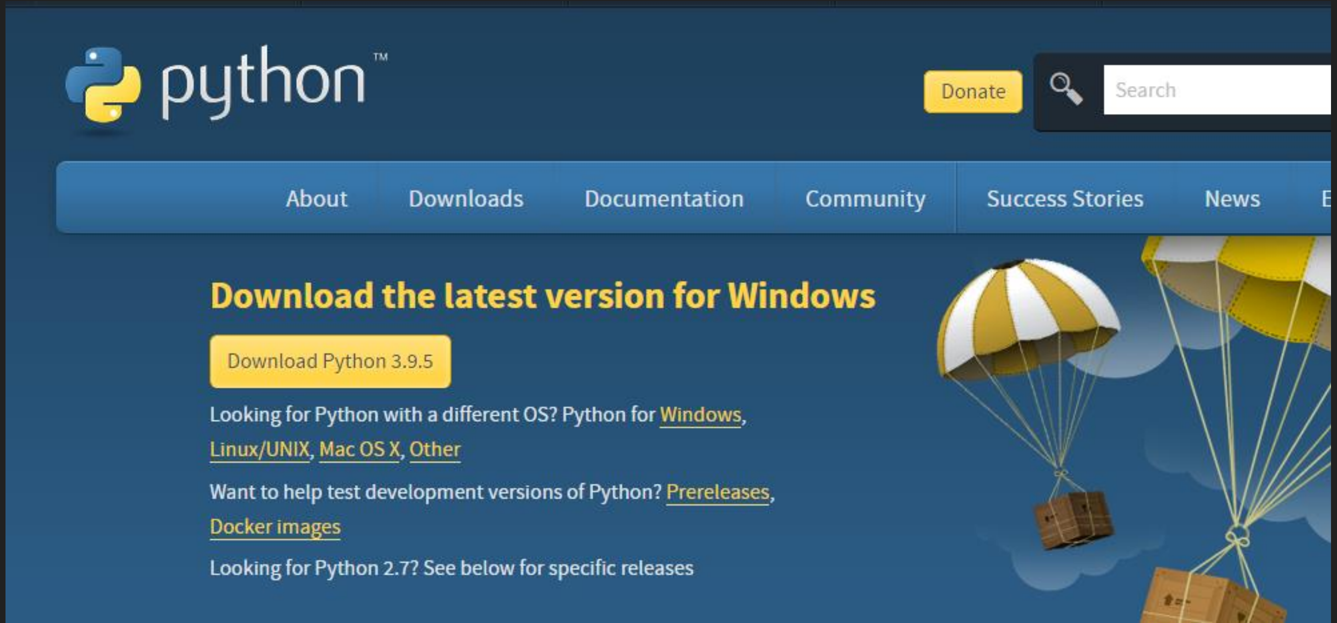
Introduction

- The purpose of these sessions is to ensure that you are fully prepared for the A level Computer Science course.
- If you have not studied GCSE Computer Science or have not programmed (using Python) the contents of the three bridging sessions will be essential to you.
- If you have studied GCSE Computer Science and are competent using Python, you should still review the content of these lessons as a refresher.
- There will be a programming task at the end of the final session. This is a compulsory task for all students and should be completed before the first lesson in Year 12.
- I hope that you enjoy these bridging sessions and I look forward to seeing you in September.

- **Mr C Murray**
- Head of Computer Science

Programming and Python Idle

- **Programming** – we will use a program called **Python Idle**
- This is a free software package that will enable us to write and execute Python programs
- The software is available from https://www.python.org/downloads/
- Download the latest version of **Python 3** (in this screen shot, I would click the button for Python 3.9.5



Download the latest version for Windows

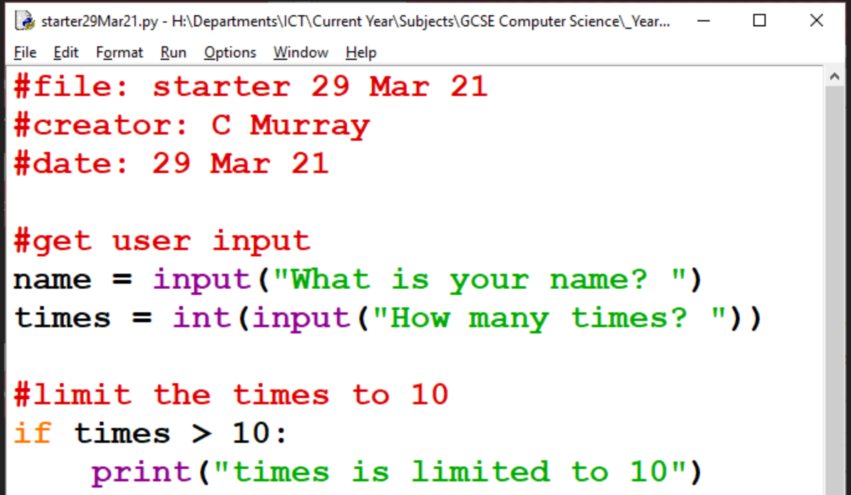
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Prereleases, Docker images

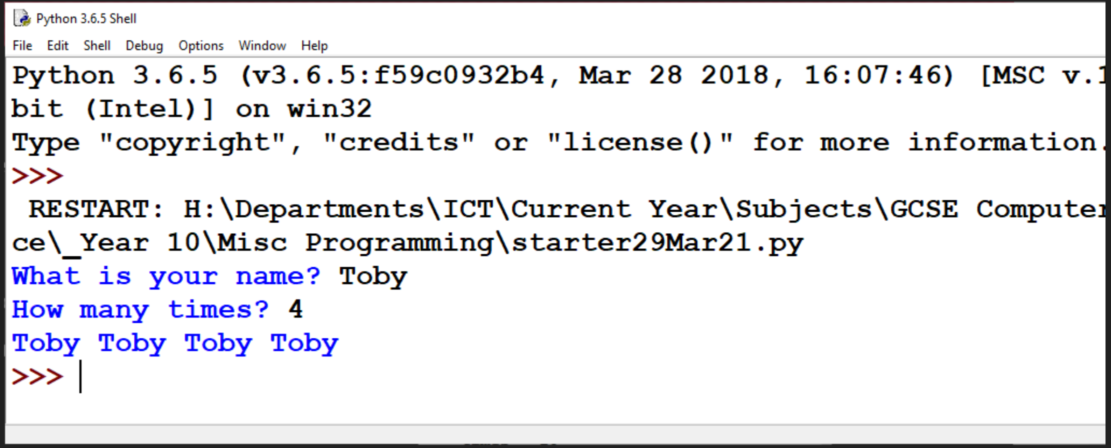
Looking for Python 2.7? See below for specific releases

Programming – The IDE

- Python Idle is an **IDE**
- IDE – **Integrated Development Environment**. This is a program that allows us to write and run programs in Python.
- Python Idle has a:
  - **Editor** – this is where you **type python statements** in order to create a program.
  - **Shell** – this is where your program **output** normally appears


starter29Mar21.py - H:\Departments\ICT\Current Year\Subjects\GCSE Computer Science_Year...
File Edit Format Run Options Window Help

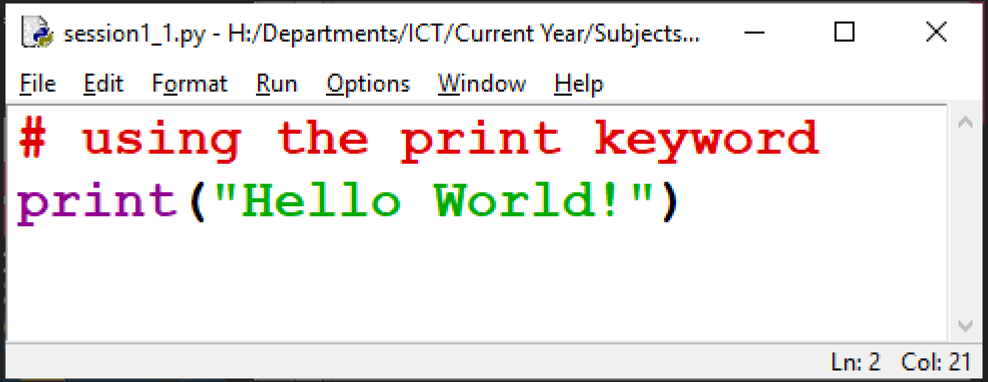
```
#file: starter 29 Mar 21
#creator: C Murray
#date: 29 Mar 21

#get user input
name = input("What is your name? ")
times = int(input("How many times? "))

#limit the times to 10
if times > 10:
    print("times is limited to 10")
```



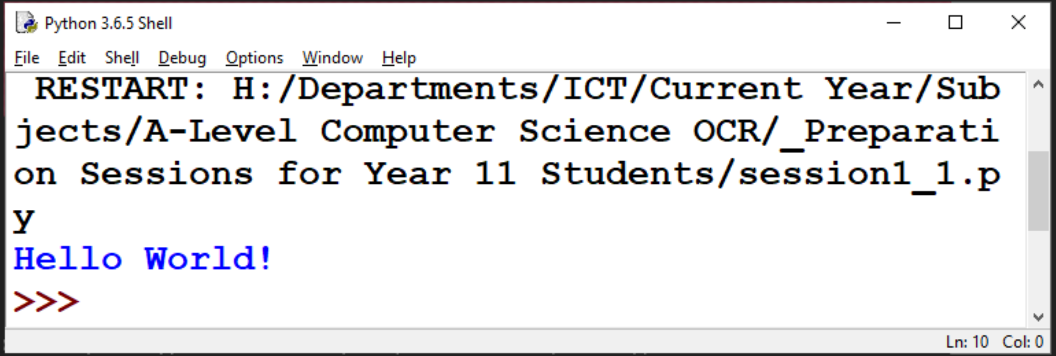
```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: H:\Departments\ICT\Current Year\Subjects\GCSE Compute
ce\_Year 10\Misc Programming\starter29Mar21.py
What is your name? Toby
How many times? 4
Toby Toby Toby Toby
>>> |
```

Programming – PRINT

- The print statement will print data/information out to the shell.
- Example
  - print(“Hello World!”)
- It is important to type the keyword correctly – lowercase and spelt correctly
- The message that you are printing out is normally in speech marks, and typed inside brackets
- These strict rules are called syntax (grammar rules for programming languages)


session1_1.py - H:/Departments/ICT/Current Year/Subjects...
File Edit Format Run Options Window Help
Ln: 2 Col: 21

```
# using the print keyword
print("Hello World!")
```



```
 RESTART: H:/Departments/ICT/Current Year/Sub
jects/A-Level Computer Science OCR/_Preparati
on Sessions for Year 11 Students/session1_1.p
y
Hello World!
>>>
```

Programming – Variables and Data Types

- **Data Types** – programming languages group data into categories called data types. Below are some common data types.

- **String** – a sequence of letters, numbers and punctuation marks, enclosed in speech marks, such as "Joe 90", "this is a string!", "341234".
- **Integer** – a whole number, such as -3, 0, 234. These are sometimes referred to as int in Python.
- **Float** – a decimal number, such as 3.02, -17.444, 2.0.
- **Boolean** – True or False. This data type can only contain one of these special values.

Programming – Variables and Data Types

- **Variables** – a variable is a location in the computer's memory where data values are stored during the execution of the program.
- Variable name – we refer to a variable by using a variable name. Variables store values using a specific data type.
- Examples
  - name = "Jaivir"
  - age = 15
  - Both of these examples are called *assignment statements*, because they are assigning a value to a particular variable.
  - print(name)
  - The above statement would print out the *value* associated with the variable name.

Programming – INPUT

- input – in Python, we use the input keyword to get data from the user.
- The input keyword can provide a *prompt* to ask the user for something, and is used in an *assignment* statement, which will assign a variable with the value that the user enters.
- Example:
  - name = input(“Please enter your name: “)
  - print(name)
- The input command creates a string data type…
- If you asked someone:
  - Age = input(“Enter your age: “)

  and the user responded by typing in **18**, the variable age would contain a string (in this case **“18”**)

Programming – Calculations

- Python can be used to carry out mathematical calculations.
- The main operators are shown in the table

Operation	Operator
Addition	+
Subtraction	-
Multiplication	*
Division	/ or //
Exponentiation	**

- Example:
    - y = 2 * 10 + 3 * 4
    - print(y) … this would produce 32 (bodmas)

    - a = 3
    - b = 2
    - c = a ** b
    - print(c) … this would produce **9**

Programming – Exercises

- Exercise 1 – make a program that will ask for a user's name and the town or city they were born in. You should print out a statement with this information (e.g. "Hi Tom, so you are from Leeds!")

Note – this will require concatenation of strings. You should carry out a web search to find out how to do that.

- Exercise 2 – create a program that asks for 3 integers (whole numbers) and prints out the total of those numbers.

Note – the input command generates data type string, but you need to have integers. Research casting.

- Exercise 3 – Create a program does the following calculations (printing out the results)
  - 10 // 4
  - 10 / 4
  - 1 – 0.8