1.1 Key Terms

Word	Definition
variable	A location in memory, this can **change** throughout the runtime of a program
string	**Linear sequence** of alphanumeric characters, between parenthesis and speech marks
concatenation	Using the plus, or the comma, to **join** more than one string together
data type	Tells the program what **kind of data** is being used
indentation	Moves code **inwards** to show it belongs to the same subsection of code
interpreter	Mainly **used for testing**, will run one line of code at a time.
compiler	**Collects** every line of code together and checks for errors before executing
debugging	Searching for, and removal of, **errors** in code
parenthesis	A synonym for **brackets**
comment	Ignored by the program, but useful for the developer. a **brief explanation** of what the code is expected to do
Scratch	**Block-based** graphical programming language
Python	**Text-based**, high level programming language
=	**Assigns** a value to a variable

1.2 searching and sorting algorithms

binary search	Finds a value in a **sorted list** by repeatedly finding the **middle value** and comparing values
linear search	Searches a **sorted list one by one** until a match is found or the entire list has been searched
insertion sort	Builds the final sorted array (or list) one item at a time, by **placing the value in** at the correct place
bubble sort	Steps through the list → compares adjacent elements → swaps them → repeat
merge sort	Breaks the list down into **individual elements** before rearranging in order

1.3 Debugging

Print ("this is my string") – Shouldn't have a capital on a print command

print ("my string is wrong) – Missing speech marks

print ("It's still a bit wrong" – missing parenthesis

// - double forward slash used to provide information of debugging

1.4 Data Types

string	Alphanumeric characters
Boolean	True/False
float/real	Decimal numbers
integer	Whole numbers
list	Array of data, list = [value1, value2...]

1.5 Maths in Python

7 + 7	Provides the **sum** of two numbers
10 - 3	**Subtracts** two numbers
6 * 6	**Multiplication** of two numbers
3 ** 3	**Exponentiation** (the power of)
10 % 3	Provides the **remainder**
9 / 2	**Divides**, returns a **float** value

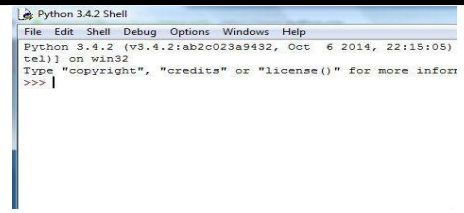
1.6 Comparison Operators

==	equal to
!=	does not equal
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

1.7 PRIMM programming model (Sue Sentance et al)

PRIMM	Structured scaffold for writing code
Predict	Evidence based guess before the code is executed
Run	Execute the code and assess prediction
Investigate	Discover the program through close inspection
Modify	Adapt elements of the program
Make	Using the program as model, develop own code

1.8a interpreter



Executes one line of code at a time
Suitable for **testing**
Highlights **errors** in each line, making it easier to fix
Faster than a **compiler** translates and executes
Low **memory** requirement

1.8b compiler

Translates data into **high-level** language
Compiles all lines of code before **executing**
Will not run if an **error** exists in the code – can be hard to locate the error
Translates and **executes** entire program
Requires more **memory**