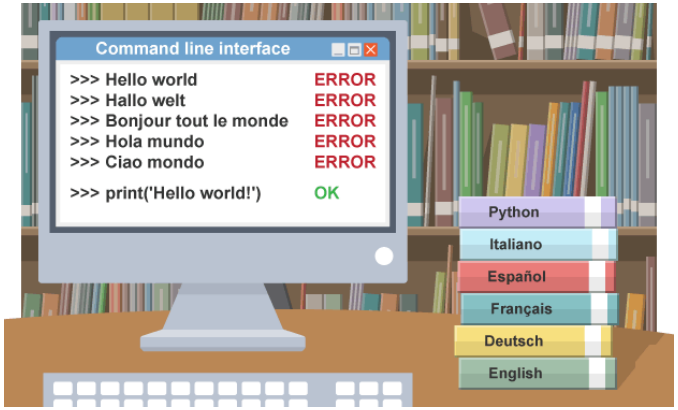
Computing Department Knowledge Organiser: Year 8 Programming in Python

What is programming?

- Programming is writing computer code to create a program, in order to solve a problem. Programs consist of a series of instructions to tell a computer exactly what to do and how to do it.



KEY TERMS

Algorithm	A program created to solve a problem
Variable	Part of the program code created to store data The data can be changed
Constant	Part of the program code created to store data The data is not changed while the program is running
Data Type	The type of data being store in a variable or constant (int, float, bool)
Casting	Changing one type of data into another
Operator	Math (+ - / *) or Logical (> < = !=)
Comment	Using a # to add explanations and annotations to your code
Integer	Number with no decimal places
Floating Point / Real	Number with a decimal place
Boolean	True or False / Yes or No

What is a programming language?

- A programming language is an artificial language that a computer understands. The language is made up of series of statements that fit together to form instructions. These instructions tell a computer what to do.
- Learn how to code:
  - www.w3schools.com/python/default.asp
  - www.bbc.co.uk/bitesize/topics/zhy39j6
  - www.codecademy.com

Computing Department Knowledge Organiser: Year 8 Programming in Python

Python -> English	
`print("hello!")`	Prints a value on screen (in this case, hello!)
`input("")`	Inputs a value into the computer.
`x = input("")`	Inputs a value and stores it into the variable x.
`x = int(input(""))`	Inputs a value into x, whilst also making it into an integer.
`answer = x + y`	Saves the result of x and y added together in a variable named answer.
`print(str(x))`	Prints the variable x, but converts it into a string first.
`print("Hello", "World")`	Prints the two strings concatenated with a space between. This code would output "Hello World".
`age = 12` `print("Age: " + str(age))`	The + joins together two variables when printing. Str has to be used to cast age to be a string. This code will output "Age: 12".
`if name == "Fred":`	Decides whether the variable 'name' ha a value which is equal to 'Fred'.
`else:`	The other option if the conditions for an if statement are not met (eg. name = 'Bob' when it should be Fred)
`elif name == "Tim":`	elif (short for else if) is for when the first if condition is not met, but you want to specify another option.
`# COMMENT`	# is used to make comments in code – any line which starts with a # will be ignored when the program runs. They are used to describe the code to a programmer.
`for i in range(0,10):` `  # WRITE CODE HERE`	Repeats any code indented after this line a set number of times, in this case, 10.
`while x < 10:` `  # WRITE CODE HERE`	Repeats any code indented after this line until a condition is met, in this case x becoming equal to or greater than 10.
`list = ["","",""]`	Creates a variable and makes it an array – a list which can store many values.

Tasks:

1. What is a programming language?
2. Describe what an algorithm is?
3. Describe what a variable is?

Tasks:

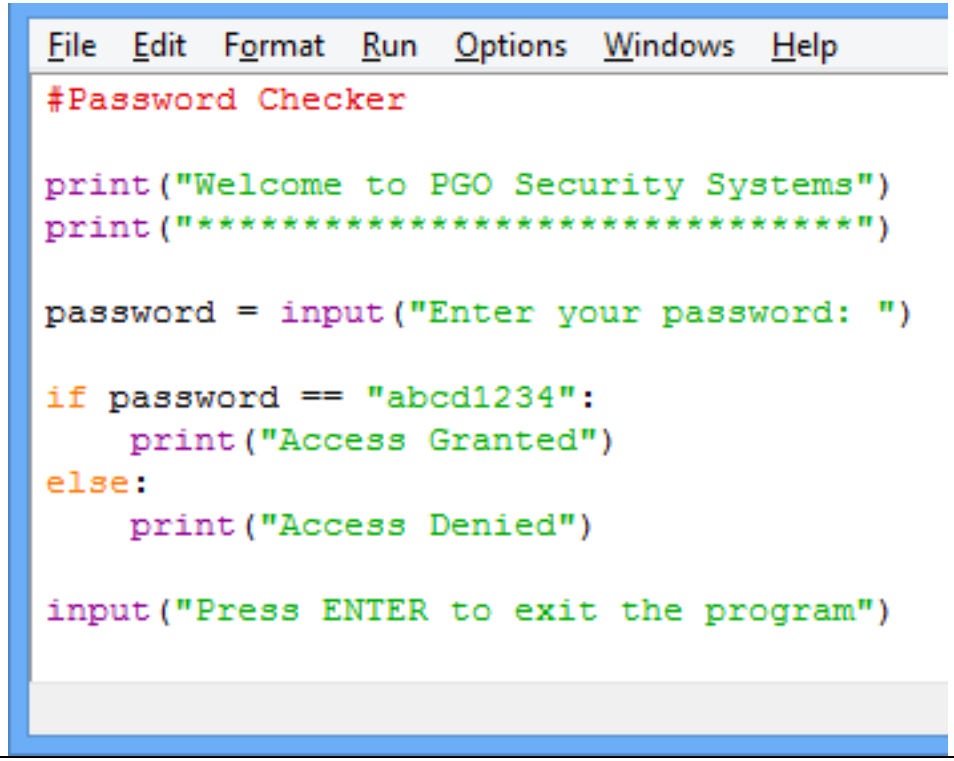
4. Describe what a data type is?
5. Describe what the terms, string, integer and float/Real mean?
6. Describe what the term Boolean means?

Data types		
Data Type	This indicates how the data will be stored. The most common data types are integer, string, and float/real.	Casting code
String	A combination of letters, numbers or characters. (eg, Hello, WR10 1XA)	`str(x)`
Integer	A whole number. (eg. 1, 189)	`int(x)`
Float/Real	A decimal number, not a whole number. (eg. 3.14, -26.9)	`float(x)`
Boolean	1 of 2 values. (eg. True, False, Yes, No)	`bool(x)`
Char	A single character	`char(x)`

Comparative operators	
==	Equal to
!=	Not equal to (or different to)
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Arithmetic operators			
Operation	**Symbol**	**Example**	**Output**
Addition	+	2 + 10	12
Subtraction	-	9 – 6	3
Multiplication	*	5 * 4	20
Division	/	5 / 2	2.5
Floor Division	//	7 // 2	3
Remainder	%	7 % 3	1

Computing Department Knowledge Organiser: Year 8 Programming in Python

Coding in Python	Tasks
1. **Writing error-free code** a. When writing programs, code should be as legible and error free as possible. b. Debugging helps keep code free of errors and documenting helps keep code clear enough to read. 2. **Syntax errors** - Syntax is the spelling and grammar of a programming language. In programming, a syntax error occurs when: a. there is a spelling mistake. b. there is a grammatical mistake 3. **Adding Comments** a. Comments are useful to help understand your code. b. They will not affect the way a program runs. Comments appear in **red** and have a preceding # symbol.	**Task 1:** Explain what the Top 3 Tips for Coding in Python are? **Task 2:** Read through the code below and write it out in your home learning book **Task 3**: Can you describe what each line of the code does? **Task 4:** Try writing out the code in Python. **Task 5:** Try writing your own code, you can do this in Python or your home learning book. 

File Edit Format Run Options Windows Help

```
#Password Checker

print("Welcome to PGO Security Systems")
print("********************************")

password = input("Enter your password: ")

if password == "abcd1234":
    print("Access Granted")
else:
    print("Access Denied")

input("Press ENTER to exit the program")
```