Year 8 | Topic 3 | Computer Science | Python Programming

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.
`print(str(x))`	Prints the variable x, but converts it into a string first.
`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.

Print statements

In order to display text in the **shell** you need to use a **Print** statement.

```
print ("Hello World")
print ("I am a programmer")
```

This is the output:

```
Hello World
I am a programmer
```

Input statements

Using **var = input ()** we can ask a user to input some information.

We can then **print** this back to the console window.

```
userName = input("what is your name?")
print ("Welcome ", userName)
```

userName is a **variable.** This means we can change the information stored. We can also name it whatever we want.

Indents

When the next line of code is optional or belongs to something else. (See example in IF statements)

Data types

Different types of data are stored in variables as different **data types**. There are **three** main data types:
String, Integer & Float

String
A type of variable for storing **text** "strings" **e.g.** "Hello World"

```
string = str("This is a string")
```

Integer
A type of variable for storing **whole numbers**
e.g. 10, 182, -44

```
integer = int("This is an integer")
```

Float
A type of variable for storing **decimal numbers**. Also known as a **real** number
e.g. 2.5, 5.05, 3.14

```
decimal = float("This is a decimal")
```

- Commenting your code

IF statements

IF statements can be used to select different options in a program depending on a condition. Also known as **selection.**

```
question = input("Are you revising?")
if question == "yes":
    print ("Well done!")
elif question == "no":
    print("Oh dear!")
else:
    print("I don't understand")
```

Variables

A variable is something that can be **used to store information.** The information that is stored can be changed.

Comparative Operators	
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Arithmetic Operators		
Addition/ Subtraction	Adds or subtracts 2 values together	2 + 5 = 7 6 - 4 = 2
Multiply/ Division	Multiplies or divides 2 values together	3 * 7 = 21 9 / 3 = 3
Integer division	Finds the whole number of a division	10 % 3 = 3
Modulus	Finds the remaining value after a division	10 % 3 = 1
Exponent	Finds the power of a value	7 ** 2 = 49

Python	A high level programming language
Sequence	Parts of the code that run in order and the pathway of the program reads and runs every line in order.
Selection	Selects a pathway through the code based on whether a condition (if statement) is true
Iteration	Code is repeated (looped), either ***while*** something is true or ***for*** a number of times.
Algorithm	A step by step method of solving a problem.
Variable	A value that is stored with a unique identifier and can change whilst the program is executed.
Comparative operator	When comparing data, an operator is used to set the condition. E.g. == != > <
Arithmetic operator	Performing calculations on values using the operators. E.g. + - * /
Syntax	The set rules of the programming language so that the computer can understand it. Mostly this links to punctuation.
Data type	This indicates how the data will be stored and processed. E.g. integer, float, string, Boolean.

Syntax

Syntax is what we call the format that the code needs to be in, in order to be processed correctly.
If it is not in the correct format then the code will not work.

```
Traceback (most recent call last):
  File "C:/Python33/a.py", line 2, in <module>
    prin (greeting)
NameError: name 'prin' is not defined
>>>
```

Python tells us where the error is and what type it is.
Here it says the line the error is on
Here it says what type of error.

Algorithms and programming