Computer Science - Python

Year 9 Term 2

Boolean Operators

AND	The AND operator will output True if both sides are true, otherwise it will be
OR	The OR operator will output True if either or both sides are true, otherwise it will
NOT	The NOT operator reverses the statement, if it's True it become False, if it's False it becomes True.

Comparative Operators

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

Arithmetic Operators

+	Addition
—	Subtraction
*	Multiplication
/	Division
/ /	Integer division
%	Remainder
* *	Exponent

Trinity TV

For more help, visit Trinity TV and watch the following videos:

Trinity TV > Year 9 > Computing > Term 2

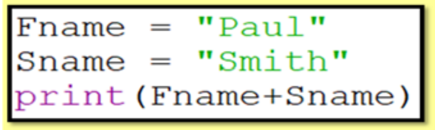
Key Terms

Python	A programming language close to English
Syntax	The rules of a language – how it is written and presented.
Sequence	Parts of the code that run in order
String	A sequence of letters, numbers and symbols in quotation marks.
Function	A piece of reusable code.
Variable	A variable is storage location for values. The values can change.
Concatenation	Adding strings and variables together.
Selection	Use of logic commands to alter the flow of a program.
Indentation	Moves code inwards to show it belongs to the same subsection of code.
Integer	Whole numbers, no decimal point.
Float	Decimal Numbers.
Boolean	Can only output the result of True or False.
Module	A file containing a set of functions you want to include
Iteration (Loops)	Repetition of a section of code for a set number of times or until a condition is met.
Array	An array is a 'list' of data items which are all the same data type.
Random Module	Allows the computer to generate a random number or option.
Comparison Operator	When comparing data, a comparison operator is used to test the condition.

Python to English

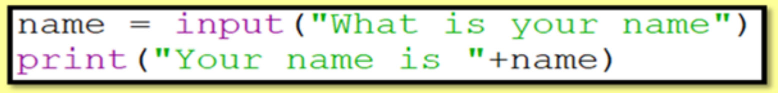
print('hello')	Prints a value on the screen
input('')	Inputs a value into the computer
x=input('')	Inputs a value and stores it into the variable x
if name == 'Fred':	Checks to see if the variable 'name' has a value that is equal to 'Fred'
else:	The other option if the conditions for an if statement are not met (e.g. name = 'Bob' when it should be Fred)

Variables / IF / ELSE / WHILE LOOPS



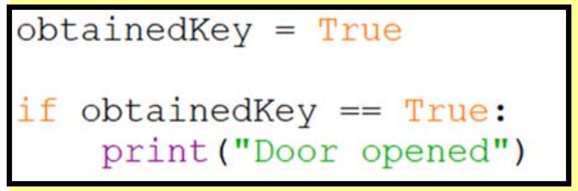
```
Fname  =  "Paul"
Sname  =  "Smith"
print(Fname+Sname)
```

A **variable** can hold a value that can be changed. We can assign a value to a variable by using an equals(=) sign.



```
name = input("What is your name")
print("Your name is "+name)
```

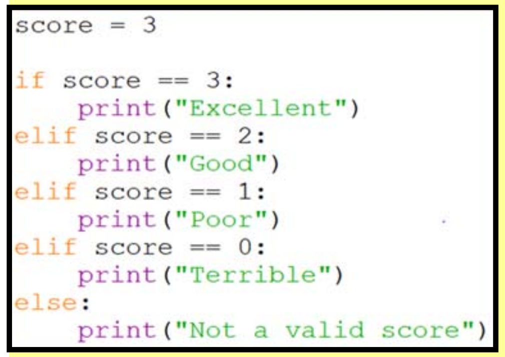
We can add 2 strings together using +, this is known as concatenating. We can get a keyboard input from the user using the input function. This example will ask the user for their name and store it in the "name" variable. We can then print that value. Combine the inputs with other Strings to print a clear message.



```
obtainedKey = True

if obtainedKey == True:
    print("Door opened")
```

F Statements allow a section of code to only run when a certain condition is met. The print will only happen if the player has the key (the variable being True).

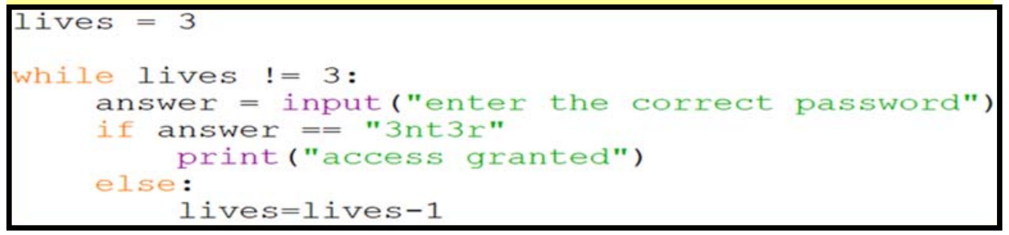


```
score = 3

if score == 3:
    print("Excellent")
elif score == 2:
    print("Good")
elif score == 1:
    print("Poor")
elif score == 0:
    print("Terrible")
else:
    print("Not a valid score")
```

ELIF and **ELSE** allows us to check variables against more conditions. We can have as many ELIF as we need but only one if and else in an else if statement block.

A **while loop** will keep repeating code until a certain condition is met. For example repeat until lives do not equal 0.



```
lives = 3

while lives != 3:
    answer = input("enter the correct password")
    if answer == "3nt3r"
        print("access granted")
    else:
        lives=lives-1
```