

Keywords

KEYWORD	DEFINITION	EXAMPLE
1. Algorithm	A series of steps or instructions written in sequential order to solve a problem.	Pick up pen, Write out name, Put pen down.
2. Sequence	A series of instructions put into a particular order	1, 2, 3 A, B, C
3. Selection	When the computer makes a decision, can be written out like an IF statement.	If it is hot outside I'll go out If it's cold I'll stay in
4. Iteration	A loop in a program where something repeats over and over.	Sit down, sit down, sit down.
5. Variable	A variable is like a box in the computer's memory, the box holds a value that can be accessed and changed.	Names Ages Shoe Sizes
6. Constant	A constant is like a box in the computer's memory, the box holds a value that can be accessed BUT CANNOT be changed.	N/A
7. Input	When you put data into a program to be processed.	Name=input("What is your name?")
8. Output	When data has been processed and information has been given as a result.	What is your name? Bob
9. Arithmetic	Used in programming to solve problems/calculations	Multiply * Divide / Plus + Subtract -
10. Flow Chart	A diagram that shows a process, made up of boxes representing steps, decision, inputs and outputs.	Rectangle - instruction Diamond - decision Equilateral - Input/output Oval - start/stop
11. Pseudo Code	A method of writing up a set of instructions for a computer program using plain English.	OUTPUT 'What is your name?' INPUT user inputs their name STORE the user's input in the name variable OUTPUT 'Hello' + name

WHILE LOOP



FOR LOOP

There are two types of loops(iteration) A **while** loop will keep repeating **whilst** a condition has not yet been met. Below the code keeps the score of a game that the user is playing. The maximum they can get to is 5. The print statement tells the computer to print out "Well done" when the score reaches 5, meaning the user got full marks.

```
score = 5
while score > 4:
    print("Well done")
```

A **for** loop will loop **for** however long you tell it to. Below this program counts 1-11. Then stops. It tells the computer to count from 0-12 and then stop. It tells it to keep +1 to the the count until it gets to 12.

```
for i in range (0, 12, +1):
    print(i)
```

PROGRAMMING CONSTRUCTS

These are the 3 main programming constructs that you will learn in programming. They are called **sequence, selection and iteration, coded examples below..**

Sequence

```
print("Welcome to SFA computer program!")
name = input("What is your name?")
print("Hi", name)

age = input ("How old are you?")
print ("You are", age)
```

Selection

```
if temperature < 19:
    print("Turn the heating ON")
else:
    if temperature > 21:
        print("Turn the heating OFF")
```

Iteration

```
score = 5

while score > 4:
    print("Well done")
```

VARIABLES AND CONSTANTS

A **variable** is like a box in the computer's memory, the box holds a value that can be accessed and changed.

A **constant** is like a box in the computer's memory, the box holds a value that can be accessed **BUT CANNOT** be changed.

12. Programming	The process of writing computer programs.	N/A
13. Iteration	A loop in a program.	For i in range
14. Debug	To find and fix any errors or mistakes in your code.	N/A
15. Data Types	This is the way that the data has been formatted within the program.	Integer, Float, Boolean, String
16. String	A sequence of alpha numeric characters used when entering data.	123 School Road
17. Integer	A whole number	1, 2, 10, 50, 100
18. Real	This is a number that contains a decimal place. It is sometimes called a float.	1.2, 3.6, 9.7
19. Testing	Checking that your program works the way that it was intended to work, and there are no mistakes within it.	N/A
20. IDLE	The window we program in.	
21. Function	A sub program that takes parameters and returns a value.	Function ticketprice (adult, child)
22. Procedure	A sub program that carries out a list of instructions	Def function1: Print ("hello") function1()
23. Sub program	A set of code within a program that can be called at any time from the main program	N/A
24. Parameter	A value that a sub program requires an order to run – it is only defined within a sub program	N/A

Print and Variable Functions

This must be done in sequence so that it runs in order. You must use the WriteLine command to get it to print out your message. The **VARIABLE** is typed in using a suitable title followed by the = sign. You must ensure that your instructions are in the **correct order** to ensure that they work.

HINT - to identify a variable in a Python program, look out for the = sign and the word input. This means the user will need to input some sort of data which will be stored in the variable until it is recalled.
`name = input ("What is your name? ")`

Data type	Description	Example
Integer	A whole number.	23
Real/floating point	A decimal number.	12.65
Character	A single alpha-numeric character.	D
String	A sequence of alpha-numeric characters.	Hello
Boolean	Either TRUE or FALSE.	FALSE

Selection - IF Statements

You would use an **IF statement** in your quiz to check whether your player has entered the correct answer. These statements are made up of 3 parts in Small Basic (1) IF (2) ELSE (3) ENDIF. You can build your score count in the IF statement in order to give a score in the quiz. Using IF statements can change the outcome of the coded program as it makes it follow a different path.

```
name = input ("What is your name? ")
age = input ("Please enter your name: ")

if age >=:
    print ("You will need to pay for an adult ticket")
else:
    print ("You can pay for a child's ticket")
```

Operators

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

Data Types

This is the way that the **data has been formatted** within the program. This is important because if it has the wrong data type then the program might not work.

Websites

- <https://www.bbc.com/bitesize/>
- www.teach-ict.com
- <http://social.technet.microsoft.com/wiki/contents/articles/17553.wiki-small-basicportal.aspx#GettingStarted>

Skills Checklist – by the end of this unit you will...

1. Be able to define the keywords.
2. Be able to demonstrate how to use programming constructs.
3. Be able to develop computer programs with independence.