**Introduction to 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.

**What is programming?**

Programs are created to implement algorithms. **Algorithms** can be represented as **pseudocode** or a **flowchart**, and programming is the translation of these into a computer program.

To tell a computer to do something, a program must be written to tell it exactly what to do and how to do it. If an algorithm has been designed, the computer program will follow this algorithm, step-by-step, which will tell the computer exactly what it should do.

**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.

There are many different programming languages, some more complicated

and complex than others. Among the most popular languages are:

* **Python**
* **Java**
* **C++**
* **BASIC**
* **Scratch**

Different languages work in different ways. For example, in Python

all instructions are written in lowercase, but in BASIC they tend

to be written in uppercase.

Programming languages are designed to be easy for a human to understand and write in. However, a computer cannot run programs written in these languages directly. Most programming languages have to be translated into **machine code** before the computer can **execute** the instructions.

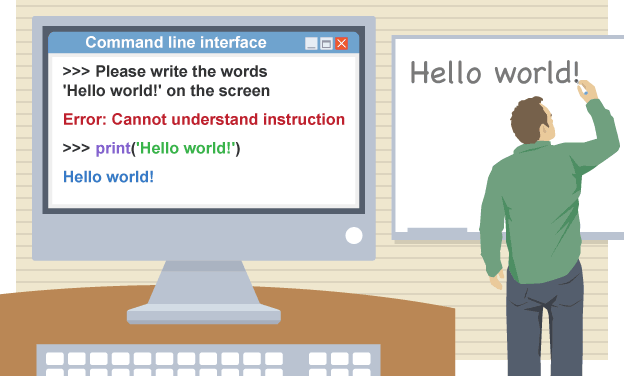
**What is a program?**

Programs are made up of statements that the programming language knows and understands. Just as words are put together to form a sentence, a program puts one or more statements together to form an instruction. Each statement tells the computer to perform a specific task, and the instructions tell a computer what to do.

**Statements**

Different programming languages use different statements. A few of these are listed in this table:

| **Statement** | **Purpose** |
| --- | --- |
| print | Output a message on the screen |
| input | Get data from the user |
| if…else | A decision |
| while | A loop controlled by a decision |
| for | A loop controlled by a counter |
| def | Create a procedure or function |

**Examples**

The following sentence asks someone to write a message on a whiteboard:

“Please write the words ‘Hello world!’ on the board.”

This sentence is an instruction, which contains a single statement.

The statement is ‘write the words’. In Python (3.x), the equivalent statement

is print. *print ("Hello world!")*

The following Python (3.x) program contains two instructions,

each built up from one statement:

*if age >= 17*

*print("You are old enough to drive a car!")*

**Creating a program from an algorithm**

Consider this simple problem. A cinema is offering discount tickets to anyone who is under 15. Decomposing this problem, gives this algorithm:

1. find out how old the person is

2. if the person is younger than 15 then say “You are eligible for a discount ticket.”

3. otherwise, say “You are not eligible for a discount ticket.”

In pseudocode, the algorithm would look like this:

*OUTPUT "How old are you?"*

*INPUT User inputs their age*

*STORE the user's input in the age variable*

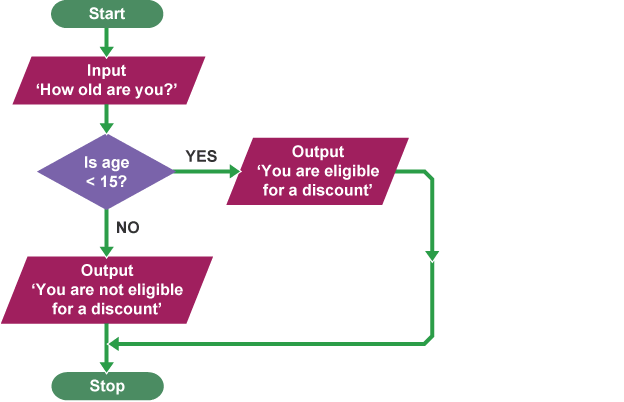
*IF age < 15 THEN*

*OUTPUT "You are eligible for a discount."*

*ELSE*

*OUTPUT "You are not eligible for a discount."*

To convert the flowchart or pseudocode into a program, look at each individual step, and write an equivalent instruction. Sometimes the steps will not match exactly, but they will be fairly close. In a flowchart, this algorithm would look like this:



**Creating the program in Python**

A **Python** (3.x) program to meet this algorithm would be:

*age = int(input("How old are you?"))*

*if age < 15:*

*print("You are eligible for a discount.")*

*else:*

*print("You are not eligible for a discount.")*

Note the similarities between the flowchart and pseudocode, and the finished program.

**Task 1. Learn new words!**

**1. algorithm -** A sequence of logical instructions for carrying out a task. In computing, algorithms are needed to design computer programs.

**2. BASIC -** A group of general-purpose, high-level programming languages that are relatively easy to use. BASIC stands for Beginner's All-purpose Symbolic Instruction Code.

**3. execute -** To run a computer program.

**4. flowchart -** A diagram that shows a process, made up of boxes representing steps, decision, inputs and outputs.

**5. instruction -** A single action that can be performed by a computer processor.

**6. Java -** A popular high-level computer programming language.

**7. machine code -** Also called object-code, this is low-level code that represents how computer hardware and CPUs understand instructions. It is represented by either binary or hexadecimal numbers.

**8. program -** Sequences of instructions for a computer.

**9. programming -** The process of writing computer software.

**10. programming language -** A language used by a programmer to write a piece of software.

**11. pseudocode -** Also written as pseudo-code. A method of writing up a set of instructions for a computer program using plain English. This is a good way of planning a program before coding.

**12. Python -** A high-level programming language.

**13. Scratch -** A high-level programming language that is presented in graphical blocks.

**14. statement -** The smallest element of a programming language which expresses an action to be carried out.

**Task 2. Move on to Test** [**https://www.bbc.com/bitesize/guides/zts8d2p/test**](https://www.bbc.com/bitesize/guides/zts8d2p/test)