

### 5.1.2 Branching

There are two main ways of controlling the sequence of actions in a program. The first of these is called **branching**, or **selection**. Branching allows you to get the computer to take one of a number of paths based on the value of a *condition*.

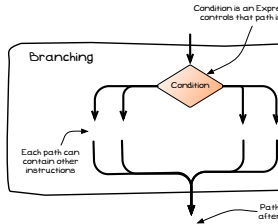


Figure 5.4: Branching commands the computer to take one of a number of paths.

#### Note

- Branching is a kind of **action**. You can command the computer to perform a branching action.
- A branch has a **condition** that is evaluated, and based on the condition, the computer takes one path.
- The branch is the act of choosing the path, when its condition is evaluated, the computer evaluates the condition and then moves to the instruction on the chosen path.
- Languages usually offer two kinds of branching statements:
  - If Statement** to select between two paths based on a Boolean condition.
  - Case Statement** to select a path based on an ordinal value.
- The Branch will have one entry point, and one exit point. This is why we combine statements together like building blocks. This idea comes from **Structured Programming**, where each component in the program has a single entry and exit point.

<sup>1</sup>Integers and Characters are ordinal values. Ordinal values have a defined sequence which value comes next in the sequence. Integers are Ordinal as you can say that the number 2 is greater than the number 1, and the number 1 is greater than the number 0.

#### CHAPTER 5. CONTROL FLOW

##### If Statement

The if statement is the most frequently used branching statement. It allows you to selectively run code based on the value of a Boolean expression (the condition). The if statement has an optional else branch that is executed when the condition is false.

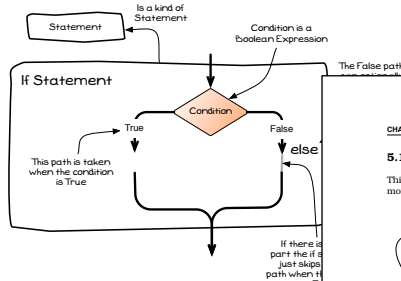


Figure 5.5: If statement lets you selectively run a branch of code.

#### Note

- An if statement is an **action**. It allows you to command the computer to perform a branching action.
- The if statement has two branches, one that is taken when the condition is True, and one that is taken when the condition is False.
- The False branch may *optionally* have instructions that are executed when the condition is False.
- If there are no instructions you want performed when the condition is False, you do not need to include an else branch, and the if statement will branch when the condition is False.
- The if statement has one entry point, two paths, and then one exit point.

#### CHAPTER 5. CONTROL FLOW

### 5.1.7 Summary

This section has introduced a number of new actions that you can use in your code to make more dynamic programs.

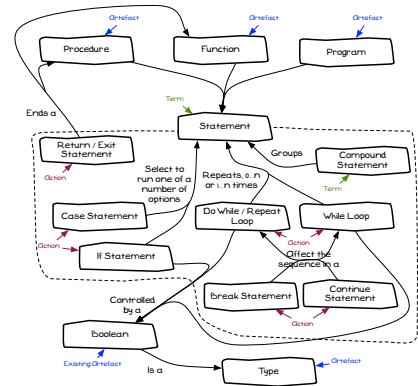


Figure 5.17: Key Concepts introduced in this Chapter

#### Note

- Artefacts** are things you can create and use.
- Terms** are things you need to understand.
- Actions** are things you can command the computer to perform.