

## Overview

Computer programs are generally written in a **programming language**, which is a formal computer language used to provide instructions for a machine. Programming languages require that code be written in a very particular syntax. To express algorithms without using a programming language, many computer scientists will instead choose to use **pseudocode**, which is a programming tool that lets us present algorithms in a natural language.

### Key Terms

- programming language
- pseudocode
- assignment
- indent

```
1 | let n = 0
2 | for each person in room
3 |     set n = n + 1
```

```
1 | stand up
2 | assign yourself the number 1
3 | until only one person remains standing
4 |     pair off with someone else standing
5 |     add your numbers together
6 |     assign yourself the new number
7 |     choose one member of the pair to sit
8 |     if you are chosen
9 |         sit down and do nothing else
```

## An Example of Pseudocode

Consider how we might write an algorithm to count the number of people in a room. We might start by thinking of the number 0, and then for each person in the room, think of the number one greater than the one we're currently thinking of.

The first block of pseudocode to the left expresses this idea. It's not written in a programming language, but it's described formally so that the steps are very precise and clear. We start by giving a name, like **n**, a value: **0**. This process is called **assignment**, and we use it so that we can refer to our value by name later in our code. Now, for each person in the room, we can re-assign **n** to be **n + 1**, so that the value increases by one for each person. Now, at the end of the algorithm, **n** is the number of people in the room.

We can try another method of counting the number of people in the room, to demonstrate some more complicated (but more efficient) pseudocode. Start by having everyone stand up, and assign themselves the number 1. Now, everyone pairs off with someone else standing, adds their numbers together, and then one person sits down. If this process repeats until there's only one person left standing in the room, then the number that they have been assigned should be the total number of people in the room.

This algorithm is expressed by the second block of pseudocode above. Notice how, while not expressed in a programming language, the algorithm is still precise. The code also **indents** some of the lines (shifts them to the right by a certain amount of space) to show what blocks of code go with which statements. For instance, since lines 4-9 are all indented, it's a sign that all of those lines should repeat 'until only one person remains standing,' per the instructions on line 3. Likewise, since 'sit down and do nothing else' on line 9 is indented, it's a sign that it should happen 'if you are chosen' per the instructions on line 8.

## Elements of Pseudocode

There's no one correct way to write pseudocode. Sometimes your pseudocode will be more or less detailed, depending on what your purpose is. Unlike a programming language, there's also no defined syntax for how pseudocode needs to be written.

There are, however, some elements that are likely to reoccur in pseudocode. Pseudocode will often involve concepts like assigning values—such as in the above examples. Pseudocode may also contain conditions (where certain code blocks will only be executed under certain conditions) as well as loops (where certain code blocks will be repeated a number of times). These concepts, which can be represented in pseudocode, are also important concepts for programming in a programming language. Even after you've learned a programming language, pseudocode can still be helpful for expressing the steps of an algorithm without having to worry about syntax, so that you can better understand your program's logic.