**FLOWCHART & PSEUDOCODE**

**What is a flowchart?**

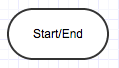
* A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence. They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes. If we consider all the various forms of flowcharts, they are one of the most common diagrams on the planet, used by both technical and non-technical people in numerous fields. Flowcharts are sometimes called by more specialized names such as Process Flowchart, Process Map, Functional Flowchart, Business Process Mapping, Business Process Modeling and Notation (BPMN),  or Process Flow Diagram (PFD). They are related to other popular diagrams, such as Data Flow Diagrams (DFDs) and Unified Modeling Language (UML) Activity Diagrams.

**Basic flowchart symbols**

First things first, in order to create a good flowchart, you must first familiarize yourself with the most commonly used flowchart symbols.

1. The Oval

An End or a Beginning

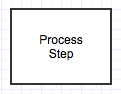
[](https://cdn2.hubspot.net/hubfs/3779096/Imported_Blog_Media/Screen-Shot-2015-02-10-at-9_09_03-AM-2.png)

**Oval**

The oval is used to represent the start and end of a process. Use the Gliffy flowchart tool to drag and drop one of these bad boys and you’ve got yourself the beginnings of a flowchart. Use the same symbol again to show that your flowchart is complete

2. The Rectangle

A Step in the Flowcharting Process

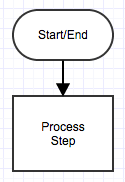
[](https://cdn2.hubspot.net/hubfs/3779096/Imported_Blog_Media/Screen-Shot-2015-02-10-at-9_12_42-AM-1.png)

**Rectangle**

The rectangle is your go-to symbol. It represents any step in the process flow you’re diagramming and is the workhorse of the flowchart diagram. Give it a lump of sugar and it will love you forever.

3. The Arrow

Directional Flow

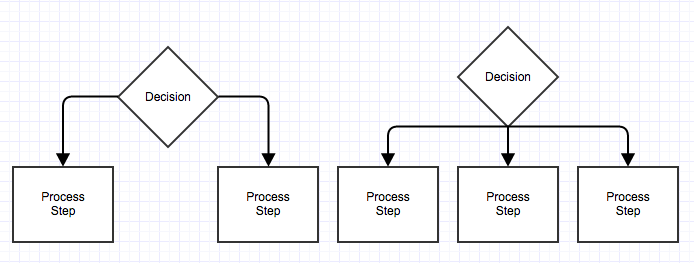
[](https://cdn2.hubspot.net/hubfs/3779096/Imported_Blog_Media/Screen-Shot-2015-02-10-at-9_13_04-AM-1.png)

**Arrow**

The arrow is used to guide the viewer along their flowcharting path. And while there are many different types of arrow tips to choose from, we recommend sticking with one for your entire flowchart. It’s less confusing and generally more aesthetically pleasing.

4. The Diamond

Call for a Decision

[](https://cdn2.hubspot.net/hubfs/3779096/Imported_Blog_Media/Screen-Shot-2015-02-10-at-9_14_03-AM-1.png)

**Diamond**

The diamond symbolizes that a decision needs to be made.  If there are only two choices, you can draw arrows directly from the diamond to the next step (example on the left). If there are more than two choices, you can draw them neatly by copying the example on the right.

**Why should I use Flowchart?**

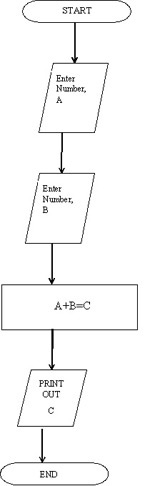
* With a flowchart you can graphically represent every step of a process and how you arrive at it without all the insider jargon that makes people’s eyes glaze over with confusion. Flowcharts can be used for more than just process documentation, too. You can use them for training, to help with troubleshooting, to give logical structure to brainstorming sessions, and more.

**What is a Pseudocode?**

* Pseudocode (pronounced SOO-doh-kohd) is a detailed yet readable description of what a computer program or algorithm must do, expressed in a formally-styled natural language rather than in a programming language. Pseudocode is sometimes used as a detailed step in the process of developing a program. It allows designers or lead programmers to express the design in great detail and provides programmers a detailed [template](https://whatis.techtarget.com/definition/template) for the next step of writing code in a specific programming language.
* Because pseudocode is detailed yet readable, it can be inspected by the team of designers and programmers as a way to ensure that actual programming is likely to match design specifications. Catching errors at the pseudocode stage is less costly than catching them later in the development process. Once the pseudocode is accepted, it is rewritten using the vocabulary and [syntax](https://whatis.techtarget.com/definition/syntax) of a programming language. Pseudocode is sometimes used in conjunction with computer-aided software engineering-based methodologies.
* It is possible to write programs that will convert a given pseudocode language into a given programming language.

**Understanding Pseudocode**

* Pseudocode makes creating programs easier. Programs can be complex and long; preparation is the key. For years, flowcharts were used to map out programs before writing one line of code in a language. However, they were difficult to modify and with the advancement of programming languages, it was difficult to display all parts of a program with a flowchart. It is challenging to find a mistake without understanding the complete flow of a program. That is where pseudocode becomes more appealing.
* To use pseudocode, all you do is write what you want your program to say in English. Pseudocode allows you to translate your statements into any language because there are no special commands and it is not standardized. Writing out programs before you code can enable you to better organize and see where you may have left out needed parts in your programs. All you have to do is write it out in your own words in short statements. Let's look at some examples.

**Examples of Pseudocode**

Let's review an example of pseudocode to create a program to add 2 numbers together and then display the result.

Start Program  
Enter two numbers, A, B  
Add the numbers together  
Print Sum  
End Program

Compare that pseudocode to an example of a flowchart to add two numbers

|  |
| --- |
|  |

Now, let's look at a few more simple examples of pseudocode. Here is a pseudocode to compute the area of a rectangle:

Get the length, l, and width, w  
Compute the area = l\*w  
Display the area

Now, let's look at an example of pseudocode to compute the perimeter of a rectangle:

Enter length, l  
Enter width, w  
Compute Perimeter = 2\*l + 2\*w  
Display Perimeter of a rectangle

Remember, writing basic pseudocode is not like writing an actual coding language. It cannot be compiled or run like a regular program. Pseudocode can be written how you want. But some companies use specific pseudocode syntax to keep everyone in the company on the same page. Syntax is a set of rules on how to use and organize statements in a programming language. By adhering to specific syntax, everyone in the company can read and understand the flow of a program. This becomes cost effective and there is less time spent finding and correcting errors.

REFFERENCES:

<https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial>

<https://www.gliffy.com/blog/how-to-flowchart-basic-symbols-part-1-of-3>

<https://whatis.techtarget.com/definition/pseudocode>

<https://study.com/academy/lesson/pseudocode-definition-examples-quiz.html>