BALLAD, NOILEEN MAE E.

TM191-HRO191

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.

**Common symbols**

|  |  |  |
| --- | --- | --- |
| **ANSI/ISO Shape** | **Name** | **Description** |
| [Flowchart Line.svg](https://en.wikipedia.org/wiki/File:Flowchart_Line.svg) | Flowline (Arrowhead)[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) | Shows the process's order of operation. A line coming from one symbol and pointing at another. Arrowheads are added if the flow is not the standard top-to-bottom, left-to right. |
| [Flowchart Terminal.svg](https://en.wikipedia.org/wiki/File:Flowchart_Terminal.svg) | Terminal[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) | Indicates the beginning and ending of a program or sub-process. Represented as a [stadium](https://en.wikipedia.org/wiki/Stadium_(geometry)),[]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) oval or rounded (fillet) rectangle. They usually contain the word "Start" or "End", or another phrase signaling the start or end of a process, such as "submit inquiry" or "receive product". |
| [Flowchart Process.svg](https://en.wikipedia.org/wiki/File:Flowchart_Process.svg) | Process[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) | Represents a set of operations that changes value, form, or location of data. Represented as a [rectangle](https://en.wikipedia.org/wiki/Rectangle). |
| [Flowchart Decision.svg](https://en.wikipedia.org/wiki/File:Flowchart_Decision.svg) | Decision[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) | Shows a conditional operation that determines which one of the two paths the program will take.[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) The operation is commonly a yes/no question or true/false test. Represented as a diamond ([rhombus](https://en.wikipedia.org/wiki/Rhombus)).[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) |
| [Flowchart IO.svg](https://en.wikipedia.org/wiki/File:Flowchart_IO.svg) | Input/Output[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) | Indicates the process of inputting and outputting data,[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) as in entering data or displaying results. Represented as a [parallelogram](https://en.wikipedia.org/wiki/Parallelogram).[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) |
| [Flowchart Annotation.svg](https://en.wikipedia.org/wiki/File:Flowchart_Annotation.svg) | Annotation[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) (Comment)[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) | Indicating additional information about a step in the program. Represented as an open rectangle with a dashed or solid line connecting it to the corresponding symbol in the flowchart.[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) |
| [Flowchart Predefined Process.svg](https://en.wikipedia.org/wiki/File:Flowchart_Predefined_Process.svg) | Predefined Process[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) | Shows named process which is defined elsewhere. Represented as a rectangle with double-struck vertical edges.[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) |
| [Flowchart Connector.svg](https://en.wikipedia.org/wiki/File:Flowchart_Connector.svg) | On-page Connector[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) | Pairs of labeled connectors replace long or confusing lines on a flowchart page. Represented by a small circle with a letter inside.[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14)[[18]](https://en.wikipedia.org/wiki/Flowchart#cite_note-RFF-18) |
| [Off page connector.png](https://en.wikipedia.org/wiki/File:Off_page_connector.png) | Off-page Connector[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) | A labeled connector for use when the target is on another page. Represented as a [home plate](https://en.wikipedia.org/wiki/Baseball_field#Home_plate)-shaped [pentagon](https://en.wikipedia.org/wiki/Pentagon).[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14)[[18]](https://en.wikipedia.org/wiki/Flowchart#cite_note-RFF-18) |

**Other symbols**

|  |  |  |
| --- | --- | --- |
| **Shape** | **Name** | **Description** |
| [Flowchart database](https://en.wikipedia.org/wiki/File:Flowchart_database.svg) | Data File or Database | Data represented by a cylinder (disk drive). |
| [Flowchart Document](https://en.wikipedia.org/wiki/File:Flowchart_Document.svg) | Document | Single documents represented a [rectangle](https://en.wikipedia.org/wiki/Rectangle) with a wavy base. |
| [Flowchart Document multiple](https://en.wikipedia.org/wiki/File:Flowchart_Document_multiple.svg) | Multiple documents represented stacked [rectangle](https://en.wikipedia.org/wiki/Rectangle) with a wavy base. |
| [Flowchar Manual input](https://en.wikipedia.org/wiki/File:Flowchar_Manual_input.svg) | Manual operation | represented by a [trapezoid](https://en.wikipedia.org/wiki/Trapezoid) with the longest parallel side at the top, to represent an operation or adjustment to process that can only be made manually. |
| [Flowchart manual input](https://en.wikipedia.org/wiki/File:Flowchart_manual_input.svg) | Manual input | Represented by [quadrilateral](https://en.wikipedia.org/wiki/Quadrilateral), with the top irregularly sloping up from left to right, like the side view of a keyboard. |
| [Flowchart Preparation](https://en.wikipedia.org/wiki/File:Flowchart_Preparation.svg) | Preparation or Initialization | Represented by an elongated [hexagon](https://en.wikipedia.org/wiki/Hexagon), originally used for steps like setting a switch or initializing a routine. |

## Flowcharts for computer programming/algorithms

As a visual representation of data flow, flowcharts are useful in writing a program or algorithm and explaining it to others or collaborating with them on it. You can use a flowchart to spell out the logic behind a program before ever starting to code the automated process. It can help to organize big-picture thinking and provide a guide when it comes time to code. More specifically, flowcharts can:

* Demonstrate the way code is organized.
* Visualize the execution of code within a program.
* Show the structure of a website or application.
* Understand how users navigate a website or program.

Often, programmers may write pseudocode, a combination of natural language and computer language able to be read by people. This may allow greater detail than the flowchart and serve either as a replacement for the flowchart or as a next step to actual code.

Related diagrams used in computer software include:

* Unified Modeling Language (UML): This is a general-purpose language used in software engineering for modeling.
* Nassi-Shneiderman Diagrams: Used for structured computer programming. Named after Isaac Nassi and Ben Shneiderman, who developed it in 1972 at SUNY-Stony Brook. Also called Structograms.
* DRAKON charts: DRAKON is an algorithmic visual programming language used to produce flowcharts.

Bottom of Form

## How flowcharts are used in numerous other fields

Beyond computer programming, flowcharts have many uses in many diverse fields.

**In any field:**

* Document and analyze a process.
* Standardize a process for efficiency and quality.
* Communicate a process for training or understanding by other parts of the organization.
* Identify bottlenecks, redundancies and unnecessary steps in a process and improve it.

**Education:**

* Plan coursework and academic requirements.
* Create a lesson plan or oral presentation.
* Organize a group or individual project.
* Show a legal or civil process, like voter registration.
* Plan and structure creative writing, like lyrics or poetry.
* Demonstrate character development for literature and film.
* Represent the flow of algorithms or logic puzzles.
* Understand a scientific process, like the Krebs cycle.
* Chart an anatomical process, such as digestion.
* Map out symptoms and treatment for diseases/disorders.
* Communicate hypotheses and theories, like Maslow’s hierarchy of needs.

**Sales and marketing:**

* Plot out the flow of a survey.
* Chart a sales process.
* Plan research strategies.
* Show registration flows.
* Disseminate communication policies, like an emergency PR plan.

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

<https://en.wikipedia.org/wiki/Flowchart>

**Pseudocode**

* is an informal [high-level](https://en.wikipedia.org/wiki/High-level_programming) description of the operating principle of a [computer program](https://en.wikipedia.org/wiki/Computer_program) or other [algorithm](https://en.wikipedia.org/wiki/Algorithm).
* It uses the structural conventions of a normal [programming language](https://en.wikipedia.org/wiki/Programming_language), but is intended for human reading rather than machine reading. Pseudocode typically omits details that are essential for machine understanding of the algorithm, such as [variable declarations](https://en.wikipedia.org/wiki/Variable_declaration), system-specific code and some [subroutines](https://en.wikipedia.org/wiki/Subroutine). The programming language is [augmented](https://en.wikipedia.org/wiki/Augmented_cognition) with [natural language](https://en.wikipedia.org/wiki/Natural_language) description details, where convenient, or with compact [mathematical notation](https://en.wikipedia.org/wiki/Mathematical_notation). The purpose of using pseudocode is that it is easier for people to understand than conventional programming language code, and that it is an efficient and environment-independent description of the key principles of an algorithm. It is commonly used in textbooks and [scientific publications](https://en.wikipedia.org/wiki/Scientific_publications) that are documenting various algorithms, and also in planning of computer program development, for sketching out the structure of the program before the actual coding takes place.
* No standard for pseudocode syntax exists, as a program in pseudocode is not an executable program. Pseudocode resembles, but should not be confused with, [skeleton programs](https://en.wikipedia.org/wiki/Skeleton_(computer_programming)) which can be [compiled](https://en.wikipedia.org/wiki/Compiler) without errors. [Flowcharts](https://en.wikipedia.org/wiki/Flowchart), [drakon-charts](https://en.wikipedia.org/wiki/DRAKON" \o "DRAKON) and [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) charts can be thought of as a graphical alternative to pseudocode, but are more spacious on paper. Languages such as [HAGGIS](https://en.wikipedia.org/wiki/HAGGIS) bridge the gap between pseudocode and code written in programming languages.

**Pseudocode** is a simple way of writing programming code in English. Pseudocode is not actual programming language. It uses short phrases to write code for programs before you actually create it in a specific language. Once you know what the program is about and how it will function, then you can use pseudocode to create statements to achieve the required results for your program.

## Understanding Pseudocode

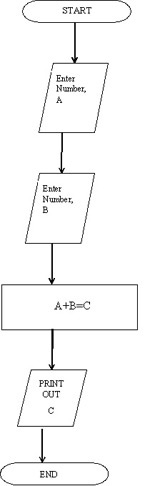
Pseudocode makes creating programs easier. Programs can be complex and long; preparation is the key. For years, flowcharts were used to map out programs before writing one line of code in a language. However, they were difficult to modify and with the advancement of programming languages, it was difficult 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.

<https://en.wikipedia.org/wiki/Pseudocode>

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