1.1 Programming Basics

Program Design Process

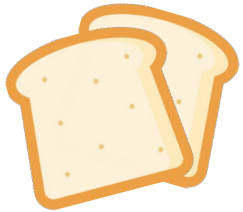
A common misconception is that development starts with writing code, which is not true, thats where implementation starts.

Development actually starts with organizing your thoughts, and making a logical plan, or a process better known as algorithm design.

What is a Algorithm?

An algorithm is essentially a set of instructions or steps that a computer (or a human) follows to accomplish a specific task. It's like a recipe for accomplishing something.

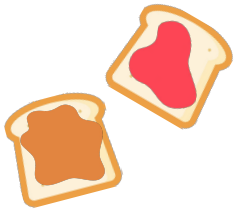
For example, let's say you wanted to make a peanut butter and jelly sandwich. You might have a set of steps you follow to accomplish this task:



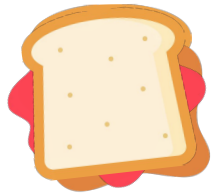
Get two slices of bread

Spread peanut butter on one slice

Spread jelly on the other slice



Put the two slices together with the peanut butter and jelly sides facing each other

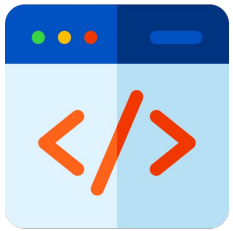


Cut the sandwich in half (if desired)

What is a Algorithm?

This set of steps is an algorithm for making a peanut butter and jelly sandwich. Similarly, a computer program might have an algorithm for sorting a list of numbers, finding the shortest path between two points on a map, or performing any other number of tasks. Algorithms are important because they allow us to solve problems and accomplish tasks in a structured and repeatable way. By breaking a task down into a set of steps, we can more easily understand it and ensure that we're accomplishing it correctly and efficiently.

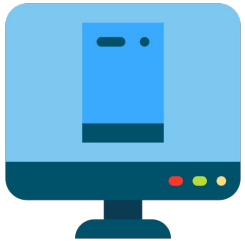
How do programs work?



Programs are written in a specific programming language that uses a set of rules to create instructions for computer to follow

Programs can use input from users, data from other sources, and external libraries or services to perform their tasks

The instructions are executed by the computer's processor and other components, such as memory an storage, to produce the desired result



Programs can also communicate with other programs or devices to exchange information or coordinate actions

What roles does JavaScript play in the ecosystem?

JavaScript is programming language that can be used to create interactive web pages and web applications

JavaScript runs on the client side (in the browser) and on the server side (on the web server), allowing for a wide range of applications and use cases

JavaScript has a large and active community of developers who create libraries and frameworks to extend its capabilities and simplify development

JavaScript is often used in conjunction with other web technologies, such as HTML, CSS, and APIs, to create rich and dynamic user experience

Program Design Steps

Before you write one line of code, you must complete the following 3 steps.

1. Set the goal of the algorithm
2. Define the steps
3. Write a algorithm

Step 1: Set the goal

Before you solve a problem, it is imperative that you first understand exactly what the problem is.

Scenario: Imagine being a plumber, and getting a call that there is a leak in a house. If you go into the house and replace all the pipes in the bathrooms, but the leak was in the kitchen did you solve the issue?

Even though you spent time, and completed a set of tasks, the job still remains undone. So before beginning any work, make sure you have a clear understanding of what the problem is, and what is an acceptable solution.

Step 1: Set the goal

Before using an algorithm, it's important to set a clear goal for what you want it to achieve.

This means identifying the problem you want the computer to solve and any calculations, sorting, or user input that may be involved.

Essentially, you need to know what you want the algorithm to do so that it can perform the task effectively.

Step 2: Define the steps

To create an effective algorithm, you need to divide the problem into smaller, individual steps.

Determine the order in which these steps need to be performed, and how they are related to each other.

Consider whether any of the steps need to be repeated, or performed only under certain conditions.

In upcoming slides, we will introduce flowchart notation and pseudocode techniques that can help you plan the flow of your algorithm.

Step 3: Write the algorithm

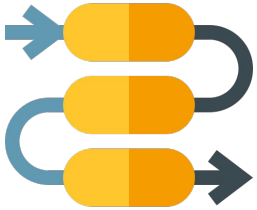
In order for the computer to understand and execute your algorithm, you must write it in a programming language, such as JavaScript.

The more complex the problem, the more crucial it is to carefully plan the individual steps before actually writing the code.

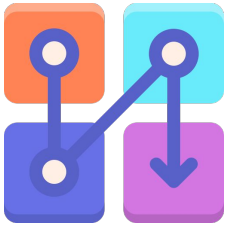
This is important because in programming and software development, "implementation" refers to the process of translating your algorithm or program flow into a specific programming language.

Flowcharts

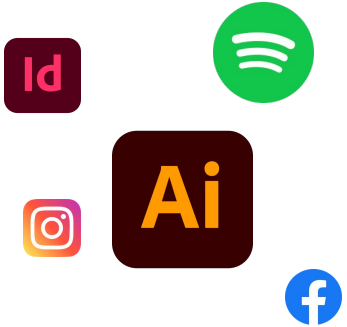
Flowcharts: What are flowcharts?



Flowcharts are diagrams that use shapes, symbols, and arrows to show the sequence of steps in a process or algorithm



They help to visualize the flow of information and decision-making within a system or program



They can be used to plan, design, and document software programs

Why are flowcharts important in software development?

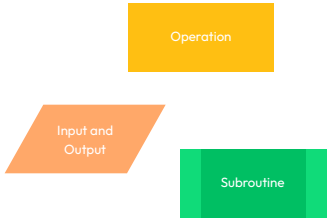
They help to break down complex processes into simpler, more manageable steps

They allow developers to visualize the flow of information and decision-making within a program

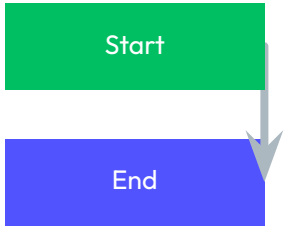
They help to identify potential errors or bottlenecks in the program flow

They serve as a valuable communication tool for developers, project managers, and stakeholders

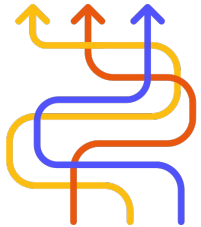
Symbols used in flowcharts



Flowchart symbols are standardized for consistency and ease of understanding

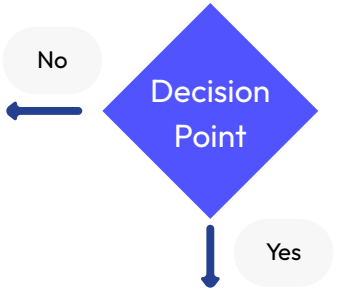


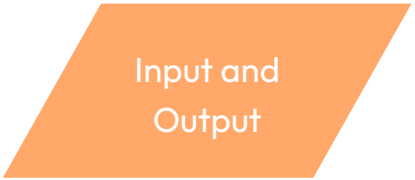
Basic symbols include start/end, process, decision, input/output, and connector



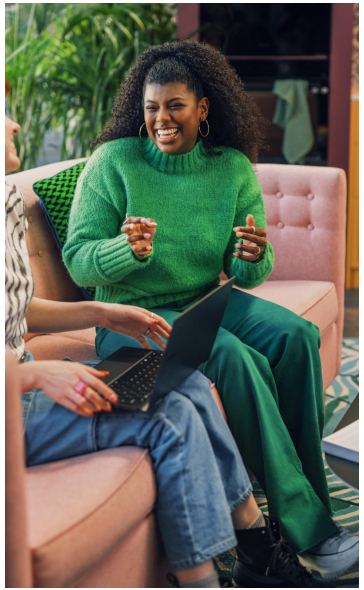
Each symbol has a specific meaning and is connected by arrows to show the flow of information or decision-making

Flowchart Symbols





Benefits of using flowcharts in software development



They help to identify potential errors or bottlenecks in the program flow before implementation

They allow developers to plan and design software programs more effectively

They serve as a communication tool for developers, project managers, and stakeholders

They can be used to document software programs for future reference and updates

Pseudocode

What is pseudocode?

Pseudocode is a simplified, high-level language used to describe the logic of an algorithm or program

It is not a specific programming language, but rather a way to express program logic in a more human-readable format

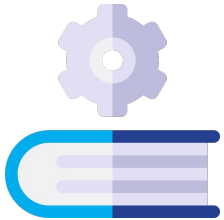
It is often used as a planning tool before writing actual code

Why is pseudocode important in software development?

It helps to clarify program logic and identify potential errors before writing actual code

It serves as a communication tool between developers, project managers, and stakeholders

It allows developers to plan and design software programs more effectively

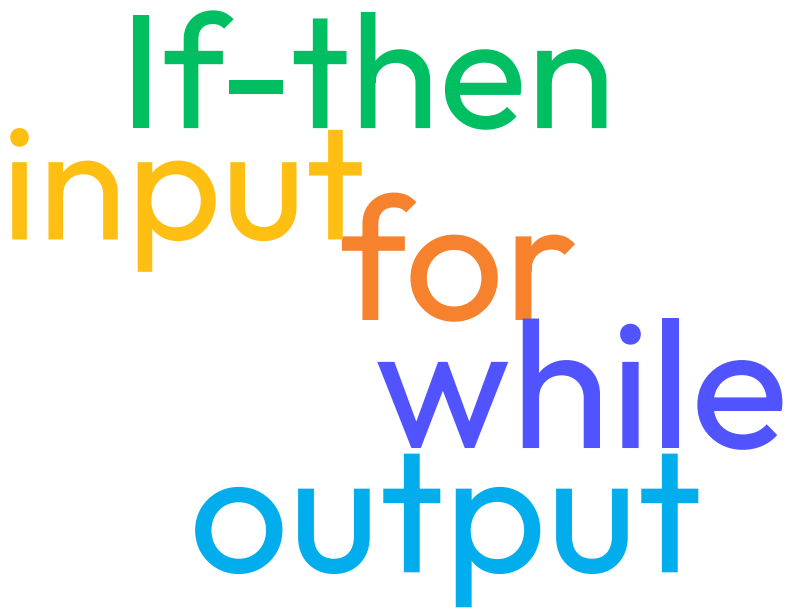


It is a useful tool for documenting software programs for future reference and updates

Pseudocode examples

```
PROGRAM START
  READ userInput
  IF userInput > 10 THEN
    PRINT "The number is greater than 10."
  ELSE
    PRINT "The number is less than or equal to 10."
  END IF
PROGRAM END
```

Pseudocode syntax



Pseudocode uses standard syntax for consistency and ease of understanding

Basic syntax includes keywords such as "if-then", "while", "for", and "input/output"

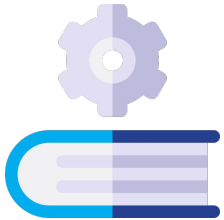
Each keyword has a specific meaning and is used to describe the logic of the program

Benefits of using pseudocode in software development

It helps to clarify program logic and identify potential errors before writing actual code

It serves as a communication tool between developers, project managers, and stakeholders

It allows developers to plan and design software programs more effectively



It is a useful tool for documenting software programs for future reference and updates

1.1 Programming Basics

Program Design Process

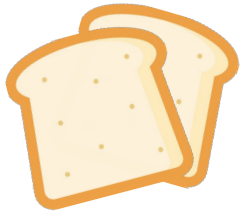
A common misconception is that development starts with writing code, which is not true, thats where implementation starts.

Development actually starts with organizing your thoughts, and making a logical plan, or a process better known as algorithm design.

What is a Algorithm?

An algorithm is essentially a set of instructions or steps that a computer (or a human) follows to accomplish a specific task. It's like a recipe for accomplishing something.

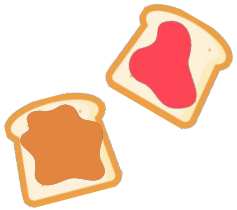
For example, let's say you wanted to make a peanut butter and jelly sandwich. You might have a set of steps you follow to accomplish this task:



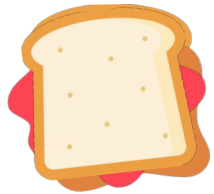
Get two slices of bread

Spread peanut butter on one slice

Spread jelly on the other slice



Put the two slices together with the peanut butter and jelly sides facing each other

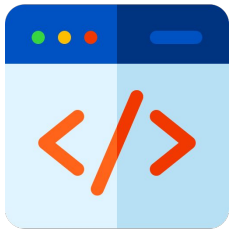


Cut the sandwich in half (if desired)

What is a Algorithm?

This set of steps is an algorithm for making a peanut butter and jelly sandwich. Similarly, a computer program might have an algorithm for sorting a list of numbers, finding the shortest path between two points on a map, or performing any other number of tasks. Algorithms are important because they allow us to solve problems and accomplish tasks in a structured and repeatable way. By breaking a task down into a set of steps, we can more easily understand it and ensure that we're accomplishing it correctly and efficiently.

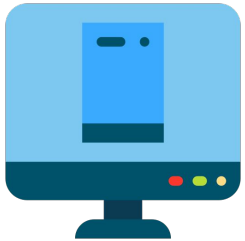
How do programs work?



Programs are written in a specific programming language that uses a set of rules to create instructions for computer to follow

Programs can use input from users, data from other sources, and external libraries or services to perform their tasks

The instructions are executed by the computer's processor and other components, such as memory an storage, to produce the desired result



Programs can also communicate with other programs or devices to exchange information or coordinate actions

What roles does JavaScript play in the ecosystem?

JavaScript is programming language that can be used to create interactive web pages and web applications

JavaScript runs on the client side (in the browser) and on the server side (on the web server), allowing for a wide range of applications and use cases

JavaScript has a large and active community of developers who create libraries and frameworks to extend its capabilities and simplify development

JavaScript is often used in conjunction with other web technologies, such as HTML, CSS, and APIs, to create rich and dynamic user experience

Program Design Steps

Before you write one line of code, you must complete the following 3 steps.

1. Set the goal of the algorithm
2. Define the steps
3. Write a algorithm

Step 1: Set the goal

Before you solve a problem, it is imperative that you first understand exactly what the problem is.

Scenario: Imagine being a plumber, and getting a call that there is a leak in a house. If you go into the house and replace all the pipes in the bathrooms, but the leak was in the kitchen did you solve the issue?

Even though you spent time, and completed a set of tasks, the job still remains undone. So before beginning any work, make sure you have a clear understanding of what the problem is, and what is an acceptable solution.

Step 1: Set the goal

Before using an algorithm, it's important to set a clear goal for what you want it to achieve.

This means identifying the problem you want the computer to solve and any calculations, sorting, or user input that may be involved.

Essentially, you need to know what you want the algorithm to do so that it can perform the task effectively.

Step 2: Define the steps

To create an effective algorithm, you need to divide the problem into smaller, individual steps.

Determine the order in which these steps need to be performed, and how they are related to each other.

Consider whether any of the steps need to be repeated, or performed only under certain conditions.

In upcoming slides, we will introduce flowchart notation and pseudocode techniques that can help you plan the flow of your algorithm.

Step 3: Write the algorithm

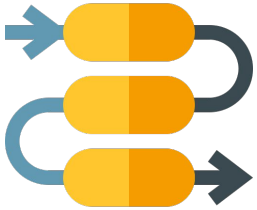
In order for the computer to understand and execute your algorithm, you must write it in a programming language, such as JavaScript.

The more complex the problem, the more crucial it is to carefully plan the individual steps before actually writing the code.

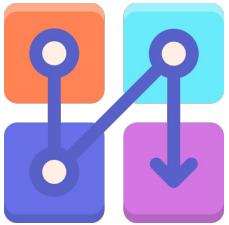
This is important because in programming and software development, "implementation" refers to the process of translating your algorithm or program flow into a specific programming language.

Flowcharts

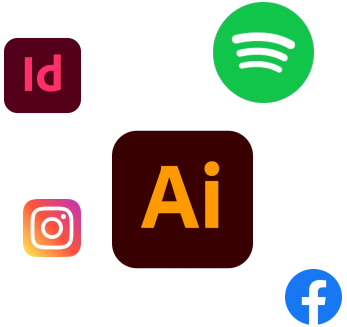
Flowcharts: What are flowcharts?



Flowcharts are diagrams that use shapes, symbols, and arrows to show the sequence of steps in a process or algorithm



They help to visualize the flow of information and decision-making within a system or program



They can be used to plan, design, and document software programs

Why are flowcharts important in software development?

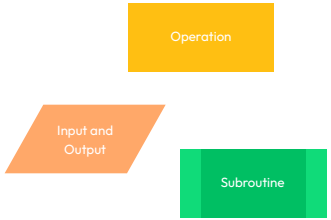
They help to break down complex processes into simpler, more manageable steps

They allow developers to visualize the flow of information and decision-making within a program

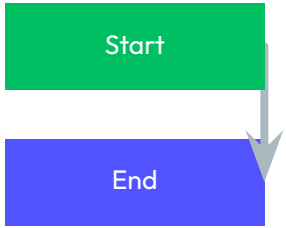
They help to identify potential errors or bottlenecks in the program flow

They serve as a valuable communication tool for developers, project managers, and stakeholders

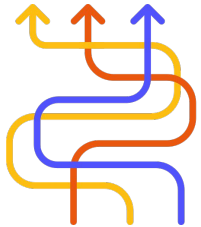
Symbols used in flowcharts



Flowchart symbols are standardized for consistency and ease of understanding

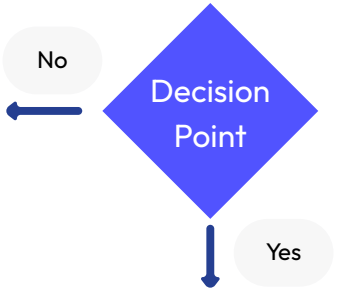


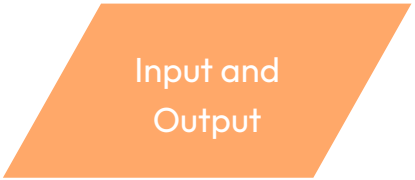
Basic symbols include start/end, process, decision, input/output, and connector



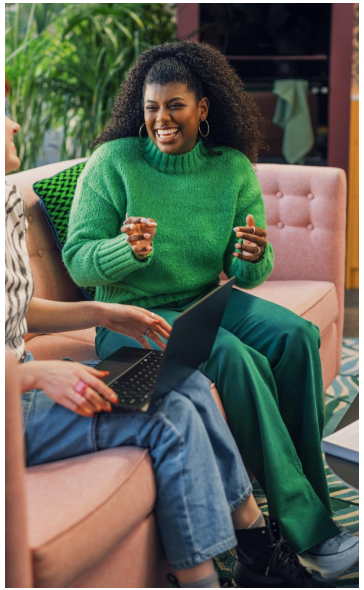
Each symbol has a specific meaning and is connected by arrows to show the flow of information or decision-making

Flowchart Symbols





Benefits of using flowcharts in software development



They help to identify potential errors or bottlenecks in the program flow before implementation

They allow developers to plan and design software programs more effectively

They serve as a communication tool for developers, project managers, and stakeholders

They can be used to document software programs for future reference and updates

Pseudocode

What is pseudocode?

Pseudocode is a simplified, high-level language used to describe the logic of an algorithm or program

It is not a specific programming language, but rather a way to express program logic in a more human-readable format

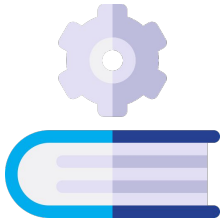
It is often used as a planning tool before writing actual code

Why is pseudocode important in software development?

It helps to clarify program logic and identify potential errors before writing actual code

It serves as a communication tool between developers, project managers, and stakeholders

It allows developers to plan and design software programs more effectively

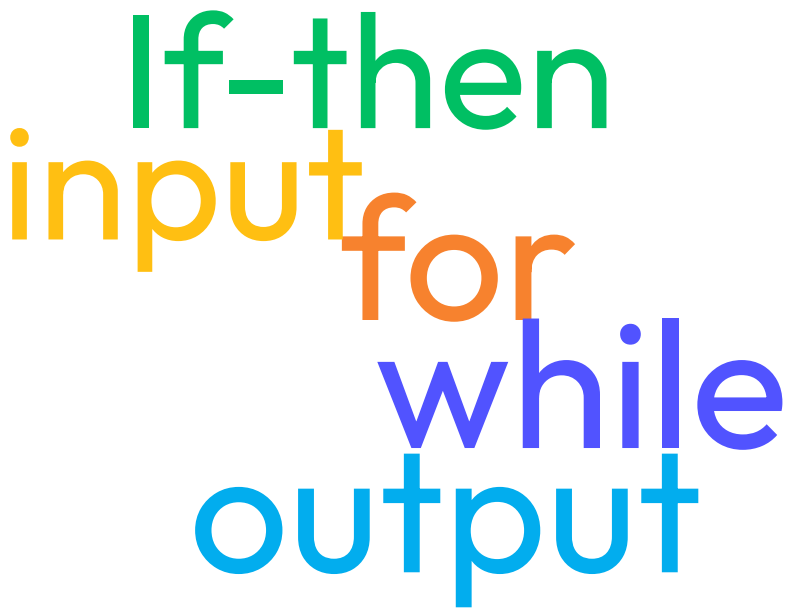


It is a useful tool for documenting software programs for future reference and updates

Pseudocode examples

```
PROGRAM START
  READ userInput
  IF userInput > 10 THEN
    PRINT "The number is greater than 10."
  ELSE
    PRINT "The number is less than or equal to 10."
  END IF
PROGRAM END
```

Pseudocode syntax



Pseudocode uses standard syntax for consistency and ease of understanding

Basic syntax includes keywords such as "if-then", "while", "for", and "input/output"

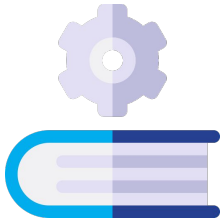
Each keyword has a specific meaning and is used to describe the logic of the program

Benefits of using pseudocode in software development

It helps to clarify program logic and identify potential errors before writing actual code

It serves as a communication tool between developers, project managers, and stakeholders

It allows developers to plan and design software programs more effectively



It is a useful tool for documenting software programs for future reference and updates