PROGRAMMING LOGIC IN 4 LANGUAGES

JS, C++, Python & PHP Made Simple



Prepared by 369 Tesla | Kalviyogi

Powered by Edunivra Pvt Ltd

About This Book

Programming Logic in 4 Languages

Same Logic. Different Syntax. Easy Learning.

1 Book Overview

Programming Logic in 4 Languages is a beginner-friendly programming foundation book created to help students understand coding concepts through **JavaScript**, **C++**, **Python**, and **PHP** together.

Many students think programming languages are completely different. But the truth is, the logic is mostly the same. Only the writing style, called syntax, changes from one language to another. This book is designed with a simple idea: understand the concept once, then see how it works in four languages.

2 What This Book Teaches

- Variables and data types
- Input and output statements
- Operators and expressions
- Conditions and decision making
- Loops and repetition
- Arrays and storing multiple values

3 Learning Method

Each topic follows a simple classroom-style pattern:

Concept Meaning → Real-Life Example → Code in 4 Languages → Output → Common Mistake → Memory Tip

4 Why This Book is Useful

- Helps beginners compare four languages easily
- Builds strong programming logic
- Makes coding less confusing
- Supports students, teachers, and self-learners
- Encourages problem-solving and step-by-step thinking

5 Final Message

This book is not just for memorizing code. It is designed to help learners think like programmers. Once the logic is clear, learning any language becomes easier.

Prepared by 369 Tesla | Kalviyogi

Powered by Edunivra Pvt Ltd

Programming Foundation Series

About Kalviyogi

Modern Learning Initiative

1 Who is Kalviyogi?

Kalviyogi is a modern learning initiative created to make education simple, visual, practical, and future-ready for students. It helps learners understand concepts clearly through smart notes, visual explanations, coding lessons, exam preparation, podcasts, AI-based learning methods, and step-by-step teaching content.



2 Our Learning Vision

> *"Learn Beyond Limits"*

Every student should get access to powerful learning methods, whether they are from a city, town, or rural background. Kalviyogi aims to bring high-quality digital education to every learner in a simple and understandable way.



3 What Kalviyogi Provides

- Visual notes
- Programming learning materials
- NEET / JEE / TRB / TNPSC support
- School subject explanations
- AI-integrated lesson design
- Podcast-based learning
- Mind maps and concept maps
- Practice questions and MCQs
- Modern digital learning resources

4 Teaching Method

Concept Meaning → Real-Life Example → Step-by-Step Explanation → Practice → Memory Tip

5 Why Kalviyogi is Different

Kalviyogi combines traditional teaching strength with modern learning technology. The goal is to make difficult subjects easy by using visuals, examples, comparison methods, storytelling, and practical learning flow.

- Student-friendly
- Visual learning
- Concept clarity
- Modern technology
- Useful for students, teachers, and self-learners



6 Final Message

Kalviyogi is not just a learning name. It is a mission to make education more clear, creative, and powerful for the next generation.



Learn Today Lead Tomorrow

4 Languages in One Programming Notes

Programming Introduction

What is Programming?

1 Simple Meaning

- Programming means giving instructions to a computer.
- A computer cannot think on its own; it follows exact steps.
- A programmer writes those steps clearly.



2 Real-Life Analogy

Teacher instruction example:
Go to class -> take notebook -> write date -> start homework.



→ Computer also works step by step like this.

3 What is a Program?

- A program is a set of instructions written in a programming language.
- Logic first, language next.
- Core idea: Problem -> Logic -> Code.



4 Same Output in 4 Languages

JavaScript:

```
console.log("Hello World");
```

C++:

```
#include <iostream>
using namespace std;
int main(){
   cout << "Hello World";
}
```

Python:

```
print("Hello World")
```

PHP:

```
<?php echo "Hello World"; ?>
```

All four give the same output: Hello World

5 Key Point

- Language may change.
- Logic stays the same.

6 Memory Tip

Programming = Problem + Logic + Language

Why Many Languages Use the Same Logic

1 Same Logic, Different Syntax

- Each language has its own syntax.
- Syntax means rules for writing code.
- Logic means the thinking used to solve a problem.

Language	Example to Show Output
JavaScript	→ console.log()
C++	→ cout
Python	→ print()
PHP	→ echo

Different syntax, same goal: show output.

2 What is Logic?



Find whether a student passed or failed.

Logic:
- If mark >= 35, show Pass.
- Otherwise, show Fail.



3 Pass / Fail in 4 Languages

JavaScript:

```
let mark = 50;
if (mark >= 35) {
    console.log("Pass");
} else {
    console.log("Fail");
}
```

C++:

```
int mark = 50;
if (mark >= 35) {
    cout << "Pass";
} else {
    cout << "Fail";
}
```

Python:

```
mark = 50
if mark >= 35:
    print("Pass")
else:
    print("Fail")
```

PHP:

```
$mark = 50;
if ($mark >= 35) {
    echo "Pass";
} else {
    echo "Fail";
}
```

Output: Pass

4 Real Understanding

- Logic does not change.
- Only the writing style changes.
- First understand the idea, then the syntax.

5 Memory Tip

Logic is the brain.
Syntax is the dress.

4 Languages in One Programming Notes

Programming Introduction

Programming Thinking Method

1 How to Think Like a Programmer

1. Understand the problem
2. Break it into small steps
3. Create logic
4. Write code
5. Run the program
6. Check output
7. Fix errors



2 Example Problem

Problem: Add two numbers and show the answer.

Logic steps:

- Take first number
- Take second number
- Add both numbers
- Store the result
- Display the answer





3 Same Logic in 4 Languages

JavaScript:

```
let a = 10;
let b = 20;
let sum = a + b;
console.log(sum);
```

C++:

```
int a = 10, b = 20;
int sum = a + b;
cout << sum;
```

Python:

```
a = 10
b = 20
sum = a + b
print(sum)
```

PHP:

```
$a = 10;
$b = 20;
$sum = $a + $b;
echo $sum;
```

Output: 30

4 Common Concepts in All Languages

Output	show result
Variable	store value
Input	get value
If	decision making
Loop	repeat task
Array	store many values

5 Beginner Tips

- Learn logic first.
- Start with small programs.
- Test the output.
- Do not fear errors.

6 Final Memory Formula

Input → Process → Output

Learn logic first.
Language becomes easy.

Four Languages Overview
JavaScript, C++, Python, PHP

Meet the 4 Languages

1 Why 4 Languages?

All four are programming languages.
Each one is used for different jobs.
Learning all four gives you more power and more choices.



2

JavaScript

- Purpose: Make web pages interactive.
- Used in: Websites (front-end).
- Why learn: It brings web pages to life.

C++

- Purpose: Build fast and efficient programs.
- Used in: Games, apps, systems, devices.
- Why learn: It builds strong programming fundamentals.

Python

- Purpose: Write simple and powerful programs.
- Used in: AI, data, automation, web, more.
- Why learn: Easy to learn and very useful.

PHP

- Purpose: Build dynamic websites.
- Used in: Web apps, servers, websites.
- Why learn: It works with databases and web forms.

3 Quick Comparison

Language	Main Use	Beginner Note
JavaScript	Make websites interactive (front-end)	Start with small web projects.
C++	Build fast programs and systems	Learn logic and problem solving.
Python	Build apps, automation, AI, data work	Very beginner friendly.
PHP	Build websites and web applications	Easy to use with HTML and forms.

4 One Important Idea



The job may differ,
but the logic is common.
Learn the idea once,
and you can use it anywhere.

5 Memory Tip

Different tools, same programming thinking.

Four Languages Overview

Basic Purpose and Usage

1 JavaScript



- Used to make websites interactive.
- Runs in the browser.
- Can also work on servers (Node.js).
- Examples: buttons, forms, alerts, menus.

Everyday Analogy

JavaScript is like the remote control that makes things respond.



2 C++



- Used for fast programs and high performance.
- Used to build software, games, and system-level tasks.
- Works close to the computer hardware.
- Examples: games, compilers, drivers.

Everyday Analogy

C++ is like the engine of a car – powerful and fast.



3 Python



- Easy to learn and very friendly.
- Used for automation, AI, data science, and simple apps.
- Great for beginners and professionals.
- Examples: data analysis, AI, scripts.

Everyday Analogy

Python is like a smart helper that makes work easy.



4 PHP



- Used on the server side of websites.
- Creates dynamic web pages.
- Works with databases.
- Examples: login systems, websites, online stores.

Everyday Analogy

PHP is like the kitchen behind the website that cooks and serves the data.



Where Each Language is Commonly Used

Language	Browser (Front-end)	Software / System	AI / Data / Automation	Website Backend (Server)
JavaScript	✓	–	–	✓ (Node.js)
C++	–	✓	–	–
Python	–	–	✓	–
PHP	–	–	–	✓

✓ = Commonly Used — = Not Common

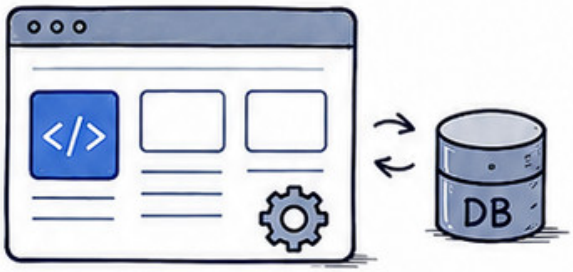
Quick Understanding

Before you start coding, know WHAT each language is mainly used for.
Right tool for the right job makes you a smart programmer!

How These Languages Help Us

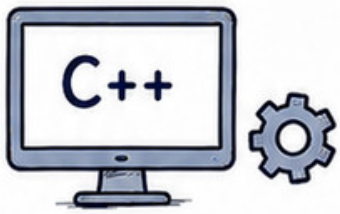
1 If you build websites

- JavaScript: makes websites interactive.
  Handles clicks, forms, menus, animations.
- PHP: works on server.
  Saves data, login system, processes forms.



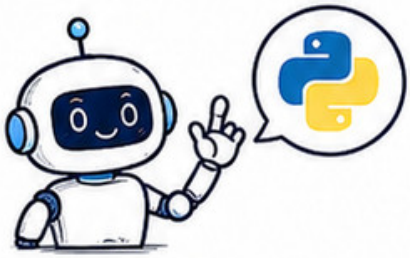
2 If you build software

- C++: builds fast and powerful applications.
  Games, tools, systems, and big programs.



3 If you want easy learning and AI

- Python: simple to learn and write.
  Used in AI, data science, automation, web apps, and more.

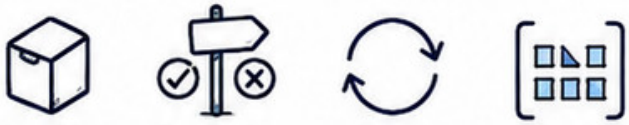


4 Common Learning Point

No matter which language you use, the core ideas are always the same:

Variables • Conditions • Loops • Arrays

These building blocks are in all languages.



5 Best Known For

Language	Best Known For
JavaScript	Making websites interactive and dynamic.
C++	High performance and system-level programming.
Python	Easy programming, AI, data science, automation.
PHP	Server-side web development and databases.



6 Memory Formula

Language changes → Purpose changes a little → Logic still stays same.

Same Logic, Different Syntax

Same Idea, Different Writing

1 What is Logic?



Logic means the idea or thinking used to solve a problem.

2 What is Syntax?



Syntax means the writing rules of a programming language.

3 One Goal, 4 Ways

Goal: Print the message "Welcome" on the screen.

JavaScript

```
console.log("Welcome");
```

Used in web pages (front-end) and servers (Node.js).

C++

```
cout << "Welcome";
```

Used for fast programs, games, systems, and applications.

Python

```
print("Welcome")
```

Used for simple scripts, AI, data science, automation, and more.

PHP

```
echo "Welcome";
```

Used to build dynamic websites and web applications.

4 Quick Difference

Concept	Logic (Thinking)	Syntax (Writing Style)
	The idea or plan to solve a problem.	The way you write that idea in code.
	Same for everyone.	Different in every programming language.
	Example: Print "Welcome"	Written differently in each language.

5 Real-Life Example

Four teachers say the same message "Welcome to class!" in different languages—Hindi, English, Tamil, and French.



Words are different, but the meaning is the same!



6 Memory Tip

Logic stays same. | Syntax changes.

Same Logic, Different Syntax

Variable Example in 4 Languages

1 Concept Meaning

A variable is a named storage box that keeps data.



Variable Analogy

Think of a variable as a labeled box that stores a value.



2 Same Idea

We store two pieces of information:

- name = "Ravi"
- age = 15



Ravi



3 Code in 4 Languages

JavaScript

```
let name = "Ravi";
let age = 15;
```

C++

```
string name = "Ravi";
int age = 15;
```

Python

```
name = "Ravi"
age = 15
```

PHP

```
$name = "Ravi";
$age = 15;
```

4 What Changes?

What Differs		How It Changes
let	Keyword	let (JS), string/int (C++), none (Python), $ (PHP)
int	Datatype	Needed in C++, not needed in others
=	Assignment Style	All use = but format differs slightly
$	Special Symbol	Only PHP uses $ before variable name

5 What Stays Same?

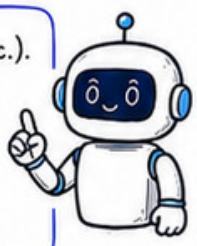
All four are storing the same information:

- name = Ravi
- age = 15



6 Beginner Note

- C++ usually needs a datatype (string, int, etc.).
- Python is simpler — no datatype needed.
- PHP uses $ before variable names.
- JavaScript often uses let for variables.



7 Memory Tip

Variable meaning is same in all languages.

Same Logic, Different Syntax

If Condition in 4 Languages

1 Problem

mark = 50.
If mark is 35 or more, show Pass.
Otherwise, show Fail.



2 Logic Steps

- Take the mark.
- Check if mark is greater than or equal to 35.
- Display "Pass" if true, otherwise "Fail".



3 Code in 4 Languages

JavaScript

```javascript
let mark = 50;
if (mark >= 35) {
    console.log("Pass");
} else {
    console.log("Fail");
}
```

C++

```cpp
int mark = 50;
if (mark >= 35) {
    cout << "Pass";
} else {
    cout << "Fail";
}
```

Python

```python
mark = 50
if mark >= 35:
    print ("Pass")
else:
    print ("Fail")
```

PHP

```php
$mark = 50;
if ($mark >= 35) {
    echo "Pass";
} else {
    echo "Fail";
}
```

4 Output



5 Syntax Change Box

{ } JavaScript and C++ use braces { }.

Python uses indentation instead of braces.

PHP uses $ before variable names.

6 Final Understanding

Same decision, different writing style.

7 Memory Formula

Idea same → syntax different → output same.

4 Languages in One Programming Notes

Program Structure

Basic Program Structure

1 What is Program Structure?



- Program structure means the arrangement of lines and parts in a program.
- It helps us know where the program starts, what it does, and how it ends.

2 Common Flow

Start → Instructions → Process → Output → End

All four languages follow this same basic idea.

3 Basic Skeleton in 4 Languages

JavaScript

```javascript
let message = "Hello";
console.log(message);
```

C++

```cpp
#include <iostream>
using namespace std;
int main() {
    string message = "Hello";
    cout << message;
    return 0;
}
```

Python

```python
message = "Hello"
print(message)
```

PHP

```php
<?php
$message = "Hello";
echo $message;
?>
```

4 What to Notice

- JavaScript usually uses semicolons.
- C++ needs #include, main(), and return 0.
- Python is simple and direct.
- PHP starts with <?php and uses $ for variables.

5 Quick Understanding

JavaScript	→	flexible web language
C++	→	formal program structure
Python	→	easiest structure
PHP	→	server-side web structure

6 Memory Tip

Different structure, same programming idea.

Program Structure

Important Parts of Each Program

1 JavaScript Structure

```
let name = "Ravi";
console.log(name);
```

- Code can start directly.
- console.log() shows output.
- Semicolons are commonly used.

2 C++ Structure

```
#include <iostream>
using namespace std;
int main() {
    cout << "Ravi";
    return 0;
}
```

- #include brings library support.
- using namespace std; is commonly used.
- main() is the starting function.
- return 0; ends the program.

3 Python Structure

```
name = "Ravi"
print(name)
```

- Python code starts directly.
- print() gives output.
- Indentation is important in blocks.

4 PHP Structure

```
<?php
$name = "Ravi";
echo $name;
?>
```

- PHP begins with <?php.
- Variables use the $ symbol.
- echo shows output.
- PHP is common in website backend.

5 Quick Comparison

Language	Starts With	Output Command	Special Note
JavaScript	direct code	console.log()	semicolon common
C++	#include and main()	cout	formal structure
Python	direct code	print()	very simple
PHP	<?php	echo	$ for variables

6 Memory Tip

Learn the parts first; writing programs becomes easy.

Program Structure

One Program, Four Structures

1 Goal

Print the message "Welcome Student".

2 Code in 4 Languages

JavaScript

```javascript
console.log("Welcome Student");
```

C++

```cpp
#include <iostream>
using namespace std;
int main() {
  cout << "Welcome Student";
  return 0;
}
```

Python

```python
print("Welcome Student")
```

PHP

```php
<?php
echo "Welcome Student";
?>
```

3 Line-by-Line Structure

Language	Start	Main Output Line	End
JavaScript	direct	console.log(...)	semicolon style
C++	#include and main()	cout << ...	return 0;
Python	direct	print(...)	no special ending
PHP	<?php	echo ...	?>

4 What Stays Same?

- All four show the same message.
- The goal is the same.
- Only the writing style changes.

5 Beginner Note

- First learn structure.
- Next learn variables.
- Then learn if, loops, and arrays.
- Strong basics make coding easy.

6 Memory Formula

Structure changes -> Output idea stays same.

4 Languages in One Programming Notes

First Program

Hello World in 4 Languages

1 What is Hello World?



Hello World is the first small program beginners learn.

It is used to test that the language and output are working.

2 Why First Program?

- Easy to understand.
- Tests output.
- Gives confidence to beginners.



3 Code in 4 Languages

JavaScript

```javascript
console.log("Hello World");
```

C++

```cpp
#include <iostream>
using namespace std;
int main() {
  cout << "Hello World";
  return 0;
}
```

Python

```python
print("Hello World")
```

PHP

```php
<?php
echo "Hello World";
?>
```

4 Output



5 Quick Understanding

JavaScript	→ web output
C++	→ formal structure
Python	→ easiest first program
PHP	→ web backend output

Different languages, same first output.

4 Languages in One Programming Notes

First Program

Important Parts of Hello World

1 JavaScript Hello World

```
console.log("Hello World");
```

- Code starts directly.
- console.log() shows output.
- Semicolon is common.

2 C++ Hello World

```
#include <iostream>
using namespace std;
int main() {
   cout << "Hello World";
   return 0;
}
```

- #include adds library.
- main() is starting point.
- cout gives output.
- return 0 ends program.

3 Python Hello World

```
print("Hello World")
```

- Starts directly.
- print() gives output.
- Very simple for beginners.

4 PHP Hello World

```
<?php
echo "Hello World";
?>
```

- Begins with <?php
- echo shows output.
- Often used in websites.

5 Quick Comparison

Language	Starts With	Output Command	Special Note
JavaScript	direct code	console.log()	simple
C++	#include and main()	cout	formal structure
Python	direct code	print()	easy
PHP	<?php	echo	web use

6 Memory Tip

Learn the first program first; coding becomes easy.

4 Languages in One Programming Notes

First Program

One Output, Four Ways

1 Goal

Print the message "Hello World".

2 Code in 4 Languages

JavaScript

```javascript
console.log("Hello World");
```

C++

```cpp
#include <iostream>
using namespace std;
int main() {
  cout << "Hello World";
  return 0;
}
```

Python

```python
print("Hello World")
```

PHP

```php
<?php
echo "Hello World";
?>
```

3 Line-by-Line Structure

Language	Start	Main Output Line	End
JavaScript	direct	console.log(...)	semicolon style
C++	#include and main()	cout << ...	return 0;
Python	direct	print(...)	no special ending
PHP	<?php	echo ...	?>

4 What Stays Same?

- all four show the same message.
- the goal is the same.
- only the writing style changes.

5 Beginner Note

- first learn the first program.
- then learn variables.
- then conditions and loops.
- Strong basics help coding.

6 Memory Formula

Structure changes -> Output idea stays same.

4 Languages in One Programming Notes

Output Statement

Output Statement in 4 Languages

1 What is Output Statement?



Output statement means showing information or result on the screen.

2 Why Output is Important?

- Shows result to the user.
- Helps to check if program works.
- Helps in learning and debugging.



3 Code in 4 Languages

JavaScript

```
console.log("Hello World");
```

C++

```
cout << "Hello World";
```

Python

```
print("Hello World")
```

PHP

```
echo "Hello World";
```

4 Output



5 Quick Comparison

JavaScript	→	console.log
C++	→	cout
Python	→	print
PHP	→	echo

Different commands, same output idea.

4 Languages in One Programming Notes

Output Statement

Important Parts of Output Commands

1 JavaScript Output

```
console.log("Hello World");
```

- Code starts directly.
- console.log() shows output.
- Commonly used in JavaScript.

2 C++ Output

```
cout << "Hello World";
```

- cout gives output.
- << sends data to output.
- Often used with main().

3 Python Output

```
print("Hello World")
```

- Simple command.
- Beginner friendly.
- Directly shows output.

4 PHP Output

```
echo "Hello World";
```

- echo shows output.
- Common in web pages.
- Used inside PHP code.

5 Quick Comparison

Language	Output Command	Special Note
JavaScript	console.log()	web scripting
C++	cout	standard output
Python	print()	easiest
PHP	echo	web use

6 Memory Tip

Learn the output command first; coding becomes easy.

4 Languages in One Programming Notes

Output Statement

One Output, Four Commands

1 Goal

Print the message "Welcome Student".

2 Code in 4 Languages

JavaScript

```
console.log("Welcome Student");
```

C++

```
cout << "Welcome Student";
```

Python

```
print("Welcome Student")
```

PHP

```
echo "Welcome Student";
```

3 Line-by-Line Structure

Language	Start	Main Output Line	End
JavaScript	direct	console.log(...)	semicolon style
C++	direct	cout << ...	semicolon style
Python	direct	print(...)	no special ending
PHP	direct	echo ...	semicolon style

4 What Stays Same?

- all four show the same message
- goal is same
- only command name changes

5 Beginner Note

- first learn output
- then variables
- then conditions and loops
- strong basics help coding

6 Memory Formula

Command changes -> Output idea stays same.

4 Languages in One Programming Notes



Variables Introduction



What is a Variable?

1 What is Variable?

A variable is a named place used to store data or a value in a program.



2 Why Do We Need Variables?

- Store information
- Reuse values
- Make programs easy to understand
- Change values whenever needed



3 Real-Life Box Analogy

Think of a variable like a box with a name on it. Inside the box, we keep some value.



name = Ravi | age = 15 | mark = 95

4 Examples of Stored Values

Variable (Name)	Stored Value (Content)
name	Ravi
age	15
city	Chennai
isPass	true

5 Quick Understanding



Variable name = label



Stored value = content inside the box



Program uses that value when needed

Variable = Named box for storing data.



Variables Introduction



Variable Examples in 4 Languages

1 Same Idea

All languages use variables to store values. The idea is the same, only the writing style changes.



2 Store Name and Age

The common idea: `name = Ravi, age = 15.`

3 Code in 4 Languages

JavaScript

```
let name = "Ravi";
let age = 15;
```

C++

```
string name = "Ravi";
int age = 15;
```

Python

```
name = "Ravi"
age = 15
```

PHP

```
$name = "Ravi";
$age = 15;
```

4 What Changes?



- JavaScript uses `let`
- C++ uses data types like `string` and `int`
- Python is simple and direct
- PHP uses `$` before the variable name

5 What Stays Same?

All four are storing the same information: Ravi and 15.
Only the syntax changes.



Meaning same. Syntax changes.





Variables Introduction



Rules and Tips for Variables



1 Variable Naming Basics

A variable name should tell what value it stores.

Examples:
studentName, totalMarks, age

2 Simple Rules

- Use meaningful names
- Do not start with numbers
- Avoid spaces in variable names
- Symbols are usually not allowed except special cases like $ in PHP
- Use easy names beginners can understand

3 Good vs Bad Examples

Good	Bad
studentName	1name
age	total marks
totalMarks	@age

4 Real-Life Analogy

If many boxes are in a room, each box needs the correct label. In programming too, each variable needs a clear name.



5 Beginner Note



First understand what a variable stores. Then learn how each language writes it.



Clear variable names make coding easy.





Variable Declaration



Variable Declaration Basics

1 What is Variable Declaration?

Variable declaration means creating a variable and getting it ready to store a value.

Some languages need a keyword or datatype while some are direct.



2 Why Declaration Matters



- Store data
- Identify variable names
- Help the program understand values
- Improve readability

3 Same Example in 4 Languages

JavaScript

```
let name = "Ravi";
let age = 15;
```

C++

```
string name = "Ravi";
int age = 15;
```

Python

```
name = "Ravi"
age = 15
```

PHP

```
$name = "Ravi";
$age = 15;
```

4 Quick Comparison



Language	Declaration Style
JavaScript	uses let
C++	uses datatype
Python	direct assignment
PHP	uses $ before variable name

5 Memory Tip

Declaration means telling the program what value we want to store.



Same idea, different declaration style.



4 Languages in One Programming Notes

Variable Declaration



How Each Language Declares Variables

1 JavaScript Declaration

JS

```
let name = "Ravi";
let age = 15;
```

- JavaScript often uses let for variables.

2 C++ Declaration

C++

```
string name = "Ravi";
int age = 15;
```

- C++ needs data types such as string and int.

3 Python Declaration

```
name = "Ravi"
age = 15
```

- Python is direct and simple, no datatype needed here.

4 PHP Declaration

php

```
$name = "Ravi";
$age = 15;
```

- PHP uses the $ symbol before variable names.

5 Quick Comparison Table



Language	Declaration Style	Special Note
JS JavaScript	let	simple web language
C++	datatype	formal structure
Python	direct assignment	easiest style
php PHP	$ variable	web backend style

6 Beginner Tip



First understand variable meaning.
Then learn how each language writes it.



Meaning stays same. Syntax changes.





4 Languages in One Programming Notes



Variable Declaration



One Idea, Four Declaration Styles

1 Goal

The common idea: store the values

`name = Ravi, age = 15`



2 Code in 4 Languages

JavaScript

```
let name = "Ravi";
let age = 15;
```

C++

```
string name = "Ravi";
int age = 15;
```

Python

```
name = "Ravi"
age = 15
```

PHP

```
$name = "Ravi";
$age = 15;
```

3 What Changes?



- JavaScript uses `let`
- C++ uses data types
- Python writes directly
- PHP uses the `$` symbol

4 What Stays Same?

All four store the same information:

`Ravi and 15.`



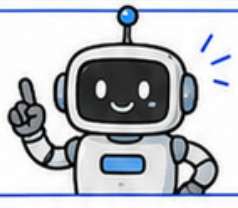
5 Line-by-Line Understanding

Language	First Part	Variable Name	Value
JavaScript	`let`	name, age	"Ravi", 15
C++	`string / int`	name, age	"Ravi", 15
Python	— (none)	name, age	"Ravi", 15
PHP	`$`	$name, $age	"Ravi", 15

6 Memory Formula



Declaration style changes → stored idea stays same.





Declare clearly, code easily.







4 Languages in One Programming Notes

Data Types Basic

What are Data Types?

1 What is Data Type?

A data type tells what kind of value a variable stores.



2 Why Data Types Matter

- Store correct values
- Avoid confusion
- Help the program work properly
- Improve readability



3 Four Common Types

123 Number	abc Text	3.14 Decimal	T/F Boolean
10, 25	"Ravi", "Hello"	99.5, 3.14	true, false

4 Real-Life Understanding

Example	Value		Data Type
age	15	→	Number
name	"Ravi"	→	Text
price	99.5	→	Decimal
isPass	true	→	Boolean

5 Quick Summary

- Number means whole value.
- Text means words/characters.
- Decimal means point value.
- Boolean means yes/no or true/false.



Data type = kind of value stored.



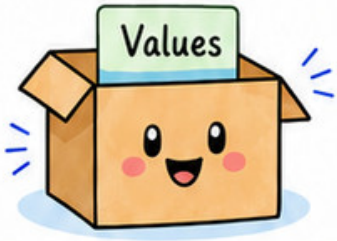
4 Languages in One Programming Notes

Data Types Basic



Data Types in 4 Languages

1 Same Values



age	=	15
name	=	"Ravi"
price	=	99.5
isPass	=	true

- age is a whole number.
- name is text.
- price is a decimal number.
- isPass is True or False.

2 Code in 4 Languages

JavaScript

```
let age = 15;
let name = "Ravi";
let price = 99.5;
let isPass = true;
```

C++

```
int age = 15;
string name = "Ravi";
float price = 99.5;
bool isPass = true;
```

Python

```
age = 15
name = "Ravi"
price = 99.5
isPass = True
```

PHP

```
$age = 15;
$name = "Ravi";
$price = 99.5;
$isPass = true;
```

3 What Changes?



- JavaScript uses number for whole and decimal values.
- C++ uses int/float/string/bool.
- Python is simple and direct.
- PHP uses $ before variable names.

4 Quick Comparison Table



Type	JavaScript	C++	Python	PHP
Number	number	int	direct number	number
Text	string	string	direct string	string
Decimal	number	float	direct decimal	decimal number
Boolean	boolean	bool	True or False	true or false

5 What Stays Same?



The meaning of the value is the same in all four languages.
Only the way we write it (syntax) changes.



Meaning same. Syntax changes.



4 Languages in One Programming Notes

Data Types Basic

Rules and Tips for Data Types

1 Number vs Decimal

- Number means whole values like 10, 25.
- Decimal means values with a point like 99.5, 3.14.

2 Text Rules

Text values are written inside quotes.

Examples: "Ravi", "Hello", "369 Tesla"

3 Boolean Rules

- Boolean has only two values: true and false.
- It is used for yes/no or pass/fail type decisions.

true false

4 Good Examples

123	$	HELLO MY NAME IS	✓
age = 15	price = 99.5	name = "Ravi"	isPass = true

5 Common Beginner Confusion

Concept	Correct ✓	Wrong / Confusing ✗
Number	age = 15 (This is a number)	age = "15" (This is text, not number)
Text	name = "Ravi" (Correct text)	name = Ravi (Wrong for beginner notes)
Boolean	isPass = true (This is boolean)	isPass = "true" (This is text, not boolean)

6 Beginner Tip

First identify what kind of value it is, then choose the data type.

Value decides the data type.



4 Languages in One Programming Notes

Variable Naming Rules



Basic Rules for Variable Names

1 What is a Variable Name?

A variable name is the label used to identify stored data in a program.



2 Main Rules

- Use letters, numbers, and underscore
- Do not start with a number
- Do not use spaces
- Avoid special symbols like @, #, %
- Choose meaningful names



3 Good vs Bad Examples

Good	Bad
✓ studentName	✗ 1name
✓ totalMarks	✗ total marks
✓ user_age	✗ @age
✓ price1	✗ price-value

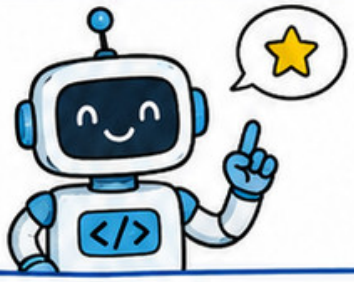
4 Real-Life Analogy

Like labeling boxes in a room, variable names should clearly tell what is inside.



5 Quick Tip

A clear name saves time while coding.





Good names make coding easy.



4 Languages in One Programming Notes

Variable Naming Rules



Common Naming Mistakes

1 Mistake 1: Starting with a Number

Wrong: `2marks` Correct: `marks2`

2 Mistake 2: Using Spaces

Wrong: `student name` Correct: `studentName or student_name`

3 Mistake 3: Unclear Names

Wrong: `a, x, data1` Correct: `studentAge, totalMarks, isPass`

Meaningful names are easier to understand.

4 Mistake 4: Special Symbols

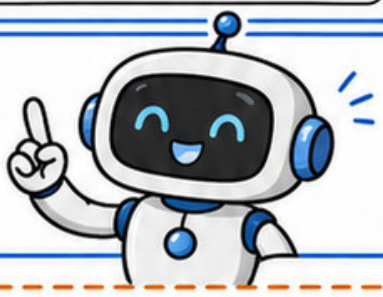
Wrong: `name@, total#, price%` Correct: `name, totalMarks, price`

5 Quick Correction Table

Wrong Name	Correct Name	Why
1name	name1	cannot start with number
total marks	totalMarks	no spaces
@age	age	no special symbols
x	studentAge	unclear meaning

6 Beginner Tip

When you read the variable name,
you should quickly understand what it stores.



Avoid mistakes, choose clear names.





4 Languages in One Programming Notes

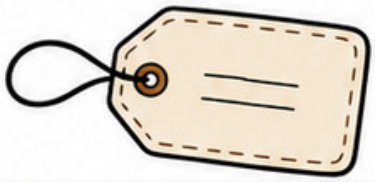
Variable Naming Rules



Good Variable Name Ideas

1 What Good Names Do

A good variable name tells the purpose of the value.



2 Examples by Category

Name	Age	Marks	Price	Boolean	Count
⇩	⇩	⇩	⇩	⇩	⇩
studentName	studentAge	totalMarks	bookPrice	isPass	totalStudents

3 Short vs Meaningful

Short	→	Meaningful
x	→	studentMarks
n	→	studentName
p	→	productPrice
f	→	isFinished

★ Meaningful names are better for learning and teamwork.

4 Boolean Name Style

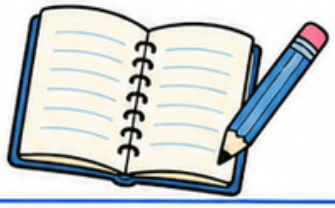
Boolean variables often start with is, has, or can.

isPass hasLicense canVote isLoggedIn



5 Mini Practice

1. Store student city → studentCity
2. Store item price → itemPrice
3. Store pass or fail → isPass



6 Final Summary

- Use simple words
- Avoid confusion
- Keep names meaningful
- Follow the rules in every language



Clear names = clear coding.



Assignment Operator



What is Assignment Operator?

1 What is Assignment Operator?

The assignment operator = is used to store a value in a variable.



2 What Does It Do?

- stores values
- connects variable and value
- helps the program remember data
- used in almost every program



3 Simple Examples

x = 10	name = "Ravi"	price = 99.5	isPass = true
Number	Text	Decimal	Boolean

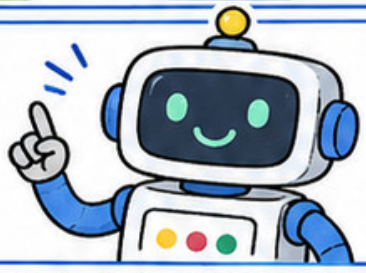
4 Real-Life Analogy

It is like writing a label on a box and placing something inside it.



5 Quick Understanding

- Left side = variable name
- Right side = value
- = means store the value





variable = value

Assignment means storing data.





Assignment Operator



Assignment Operator in 4 Languages

1 Same Idea

All languages use = to assign a value, but syntax style may differ slightly.



2 Code in 4 Languages

JavaScript

```
let age = 15;
let name = "Ravi";
```

C++

```
int age = 15;
string name = "Ravi";
```

Python

```
age = 15
name = "Ravi"
```

PHP

```
$age = 15;
$name = "Ravi";
```

3 What Changes?

- JavaScript often uses let
- C++ uses datatype
- Python is direct
- PHP uses $ before variable names



4 What Stays Same?

The meaning is the same in all four languages.

age gets 15
name gets Ravi



5 Quick Comparison Table

Language	Example 1	Example 2
JavaScript	`let age = 15;`	`let name = "Ravi";`
C++	`int age = 15;`	`string name = "Ravi";`
Python	`age = 15`	`name = "Ravi"`
PHP	`$age = 15;`	`$name = "Ravi";`

Meaning same. Syntax changes.





Assignment Operator



Rules and Tips for = Operator

1 Important Rule

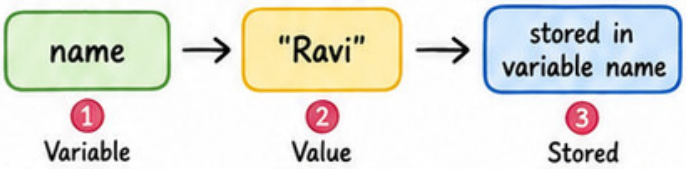
Assignment operator = stores a value.

Do not confuse it with ==, which is used for comparison.

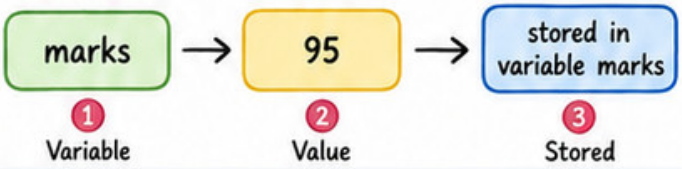
=	==
store	compare

2 Step-by-Step Example

Example 1: name = "Ravi"



Example 2: marks = 95



3 Good Examples

- age = 15
- totalMarks = 480
- price = 99.5
- isPass = true

4 Common Beginner Confusion

Correct	Meaning
x = 10	store 10 in x
name = "Ravi"	store text in name

Wrong idea	Why wrong
10 = x	value cannot store variable
x == 10	this checks, it does not store

5 Beginner Tip



Read it as: variable gets value.

6 Final Summary

- Left side should be variable
- Right side should be value
- = is for storing
- Practice with simple examples

Read x = 10 as: x gets 10.

Runtime Value Meaning

What is Runtime Value?

1 What is Runtime Value?

Runtime value means the value a variable gets while the program is running.

2 Why It Is Called Runtime

- Program runs.
- Value is stored.
- Value is used.
- Output is shown.

3 Real-Life Analogy



Imagine a student writes his marks on the board during class time.

Marks exist only while class is going on.

That written marks are like runtime value.

4 Simple Example

```
age    = 15
name   = "Ravi"
price  = 99.5
isPass = true
```

These values are known only when the program runs.

5 Quick Understanding

Left Side (Variable) → Name given to store data.

Right Side (Value) → Current value available during execution.

MEMORY TIP

Runtime value = value available while program runs.

Runtime Value Meaning

Runtime Value in 4 Languages

Example Concept: Store student name and age at runtime.

1 Same Idea

All these languages allow us to store values (like name and age) while the program is running. That is called **Runtime Value**.

4 What Stays Same?

All four languages store the same data:

`Name = Ravi, Age = 15`

at runtime.

2 Code in 4 Languages

JavaScript

```
1 let age = 15;
2 let name = "Ravi";
```

C++

```
1 int age = 15;
2 string name = "Ravi";
```

Python

```
1 age = 15
2 name = "Ravi"
```

PHP

```
1 $age = 15;
2 $name = "Ravi";
```

3 What Changes?

- JavaScript uses `let`.
- C++ uses datatype (`int`, `string`).
- Python is direct and simple.
- PHP uses `$` before variable names.

5 Quick Comparison Table

Language	Runtime Example Style
JavaScript	`let variable = value;`
C++	`datatype variable = value;`
Python	`variable = value`
PHP	`$variable = value;`

Meaning same. Syntax changes.

Runtime Value Meaning

Runtime Value Tips and Examples

1 How Runtime Value Changes

Runtime value can change while the program is running.

At one time:
mark = 90
(runtime value is 90)

→

Later in program:
mark = 95
(runtime value is 95)

Same variable, different runtime values at different times.

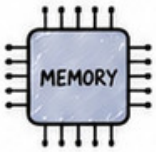
2 Good Examples

These are runtime values (during execution):

```
name   = "Ravi"   → text value
count  = 10       → integer value
total  = 99.5     → decimal value
isPass = true     → boolean value
```

3 Common Beginner Confusion

Runtime Value (changes during execution)




```
mark = 90
```
↑ this value is in memory during program run

It can change to 95, 100, etc.

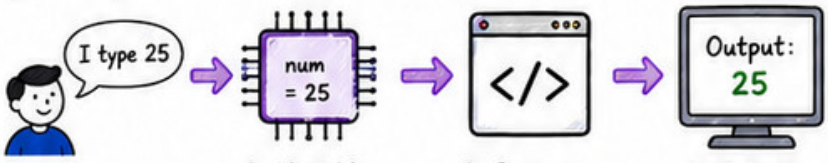
Fixed Text in Code (does not change)

```
print("Student mark is:")
```
↑ this text is fixed in code. It does not change.



Runtime value appears in memory and output during execution.

4 Input to Runtime Value



1. User Input
2. Variable stores value
3. Program uses value
4. Output shown

5 Beginner Tip

Read it as:

variable gets value while the program runs.

6 Final Summary

- Runtime value is temporary during program execution.
- It may change many times.
- It helps the program do work and produce output.

When program runs, variables get runtime values.

4 Languages in One Programming Notes

User Input Basics

What is User Input?

1 What is User Input?

User input means data given by the user while the program is running.



2 Why Do We Need Input?

- Makes programs interactive
- Accepts values from users
- Helps produce different outputs
- Useful in real programs

3 Real-Life Analogy



Just like a teacher asks a student for their name or mark, a program **asks the user for input** so it can work with that data.



4 Simple Examples of Input

name = Ravi

age = 15

mark = 95

price = 99.5

5 Quick Understanding

User → Input → Program → Output

MEMORY TIP!

Input = data given by the user while the program runs.



Input in 4 Languages



Example concept: Get student name and age from the user.

Code in 4 Languages

JavaScript

```
let name = prompt("Enter your name:");
let age = prompt("Enter your age:");
console.log(name);
console.log(age);
```

C++

```
string name;
int age;
cin >> name;
cin >> age;
cout << name;
cout << age;
```

Python

```
name = input("Enter your name: ")
age = input("Enter your age: ")
print(name)
print(age)
```

PHP

```
$name = readline("Enter your name: ");
$age = readline("Enter your age: ");
echo $name;
echo $age;
```

What Changes?

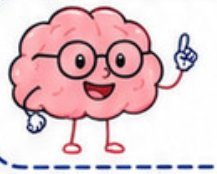
- JavaScript often uses `prompt()`.
- C++ uses `cin`.
- Python uses `input()`.
- PHP can use `readline()`.

What Stays Same?

- All four languages take values from the user.

Quick Comparison

Language	Input Command
JavaScript	`prompt("text")`
C++	`cin >> variable`
Python	`input("text")`
PHP	`readline("text")`



Memory Tip: Meaning same. Syntax changes.



User Input Basics

Input Rules and Tips

1 Basic Rule

Input usually comes as **text** first.



2 Common Beginner Confusion

If the user types **25**, the program may read it as **text** unless converted.



3 String vs Number Input

Example	What User Types	How Program Reads	Type	Good 👍	Wrong ✖
name = "Ravi"	Ravi	"Ravi"	Text (String)	✔	Using text in calculations may cause errors!
age = 15	15	"15" (as text)	Number (Integer)	✔	
price = 99.5	99.5	"99.5" (as text)	Decimal (Float)	✔	

4 Need for Conversion

Convert text input to number when needed.

JavaScript
```
Number(prompt("Enter age: "));
```

C++
```
int age; cin >> age;
```

Python
```
age = int(input("Enter age: "))
```

PHP
```
$age = (int) readline("Enter age: ");
```

5 Input Flow


User types value

↓


Program receives it

↓


Store in variable

↓


Use in calculation / output

6 Beginner Tips



- Use clear prompts (tell user what to enter).
- Store input in meaningful variables.
- Convert number inputs when needed.
- Test with simple examples.

Practice

Task: Write a program that takes input.

Enter name, age, and mark



Memory Formula

User types → program stores → output shows.



4 Languages in One Programming Notes

Runtime Addition Program

What is Runtime Addition Program?



Add two numbers using user input while the program is running.

1 What is Runtime Addition?



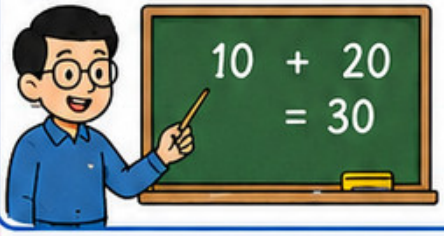
The user enters two numbers while the program runs. The program adds them and then shows the result.

2 Why It Is Called Runtime

- User gives input.
- Values are stored.
- Addition happens.
- Output is shown.



3 Real-Life Analogy



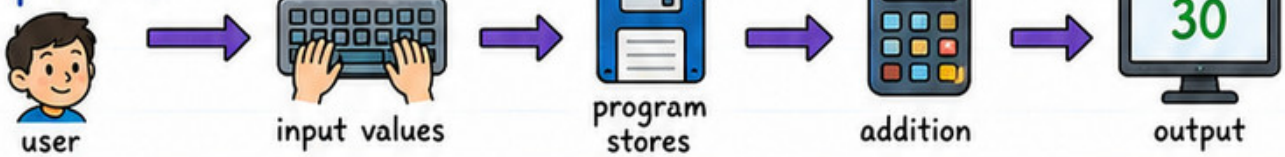
Like a teacher asks two marks and immediately adds them on the board.

4 Simple Formula

firstNumber + secondNumber = sum

Example: 10 + 20 = 30

5 Input Flow



6 Quick Understanding

Left Side = Input Numbers

15 25

Middle = Process

Right Side = Result

40



MEMORY TIP: Runtime addition = add values entered while program runs.

4 Languages in One Programming Notes

Runtime Addition Program

Add Two Numbers in 4 Languages



Get two numbers from the user and show their sum.

1 Same Idea



- All four languages take two inputs.
- They store the values.
- They add the values.
- They display the result.

2 Code in 4 Languages

JavaScript

```
let a = Number(prompt("Enter first number: "));
let b = Number(prompt("Enter second number: "));
let sum = a + b;
console.log(sum);
```

C++

```
int a, b, sum;
cin >> a;
cin >> b;
sum = a + b;
cout << sum;
```

Python

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
sum = a + b
print(sum)
```

PHP

```
$a = (int) readline("Enter first number: ");
$b = (int) readline("Enter second number: ");
$sum = $a + $b;
echo $sum;
```

3 What Changes?



- JavaScript uses prompt() and Number().
- C++ uses cin and cout.
- Python uses input() and int().
- PHP uses readline() and $.

4 What Stays Same?



All four languages ask for two numbers and show the same total.

5 Quick Comparison Table



Language	Input + Add Style
JavaScript	Uses prompt() for input, Number() to convert, + to add, console.log() to show.
C++	Uses cin to take input, + to add, cout to show.
Python	Uses input() to take input, int() to convert, + to add, print() to show.
PHP	Uses readline() to take input, (int) to convert, + to add, echo to show.

MEMORY TIP: Meaning same. Syntax changes.

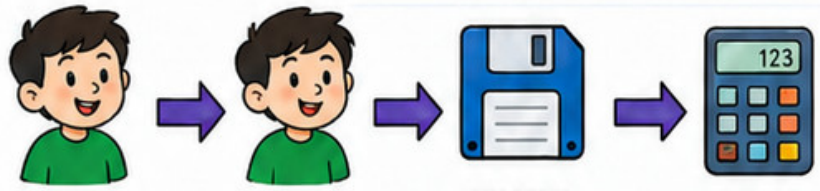
Runtime Addition Program

Runtime Addition Tips and Output



Learn simple tips to add two numbers using user input.

1 Input to Output Flow



user enters first number → user enters second number → program stores values → sum is calculated

2 Good Example

firstNumber	=	10
secondNumber	=	20
sum	=	30

3 Sample Input and Output



```
Enter first number: 12
Enter second number: 8
Output: 20
```

4 Common Beginner Confusion



If input is read as text, addition may join values like 12 and 8 incorrectly in some cases.

So, number conversion is important.

5 Beginner Tip



Read a + b as add the values stored in a and b.

6 Final Summary



- Program takes input.
- Stores the values.
- Adds the values.
- Shows the result.

MEMORY TIP: When program runs, user input becomes runtime values.

Compile Time vs Runtime

What is Compile Time and Runtime?

1 What is Compile Time?

Compile time is the stage before the program runs. During this stage, some languages check the code for errors and convert it into machine code.



2 What is Runtime?

Runtime is the stage when the program is actually running. In this stage, the program uses values (input) and gives output.



3 Classroom Example



★ Teacher checks your homework before class.
→ That is Compile Time.

☆ You write the answer in class during the test.
→ That is Runtime.



4 Quick Difference



- Compile Time → before running
- Checks code for errors and prepares it



- Runtime → while running
- Uses values (input) and shows output

5 Simple Example

C++ (Compile First, Then Run)

Step 1: Compile

```
g++ prog.cpp -o prog
```

↓

Step 2: Run

```
./prog
```

Python (Run Directly)

Run Directly

```
python prog.py
```

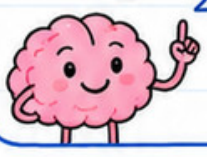
↓

Hello, World!

6 Memory Tip



Compile Time = before run,
Runtime = during run.











Compile Time vs Runtime

Compile Time vs Runtime in 4 Languages

1 Same Idea

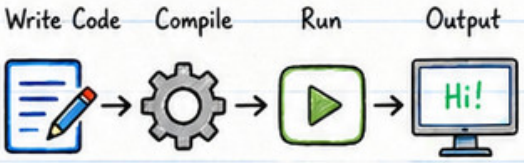
All languages follow the same logic, but some are compiled and some are interpreted or run more directly.



2 C++ Example

C++ usually has compile time and runtime clearly.

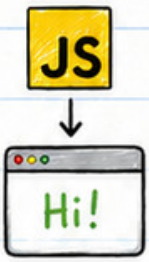
```
#include <iostream>
int main() {
  std::cout << "Hi!";
  return 0;
}
```



3 JavaScript Example

JavaScript runs and gives output at runtime.

```
console.log("Hi!");
```



4 Python Example

Python usually runs directly.

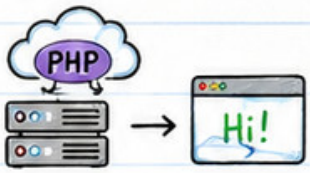
```
print ("Hi!")
```



5 PHP Example

PHP runs on the server and gives output at runtime.

```
<?php
  echo "Hi!";
?>
```



6 What Changes?

Compiled Style: code is first checked/converted (compile time) then executed (runtime).

Direct Runtime Style: code is run and output is shown directly at runtime (no separate compile step).

7 What Stays Same?



All languages finally run and produce output for the user.



8 Quick Comparison Table

Language	Compile Time Idea	Runtime Idea
JavaScript	No separate compile step.	Runs in browser at runtime and shows output.
C++	Code is compiled first (compile time).	Executable runs and shows output.
Python	No separate compile step (usually).	Interpreter runs code at runtime and shows output.
PHP	No separate compile step by programmer.	Runs on server at runtime and shows output.



Meaning same. Process style changes.



Compile Time vs Runtime

Rules, Tips, and Classroom Understanding

1 Classroom Story

Before class, the teacher checks your notebook for mistakes. That is **Compile Time**.

During class, you answer questions and use values. That is **Runtime**.



2 Good Examples

Compile Time

- Syntax error check
- Code translation (Source → Machine Code)

Runtime

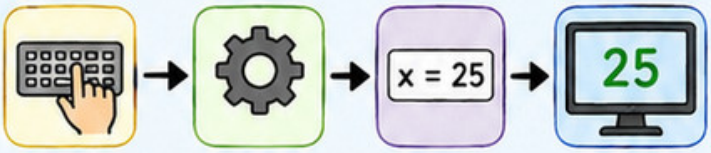
- User input
- Calculation / Value used
- Output shown — Hello!

3 Common Beginner Confusion



Do NOT confuse code checking before run (Compile Time) with values getting used during run (Runtime).

4 Input to Output Flow



5 Mini Comparison

Compile Time	Runtime
Happens before execution starts.	Happens during execution.
Checks for syntax errors.	Works with variable values and data.
Converts code from source to machine code.	Performs operations, uses values, and produces results.
Mostly seen by programmers (e.g., C++).	Seen by the user as output.

6 Beginner Tip

Read it as:

"Compile first, run next."

7 Final Summary

- Compile Time prepares the program.
- Runtime uses the program.
- One checks, the other executes.
- Both are important!

Memory Formula — Compile Time = before execution | Runtime = during execution

Interpreted Languages

4 Languages in One Programming Notes

What are Interpreted Languages?

1 What is an Interpreted Language?



An interpreted language usually runs line by line or directly while the program is running.

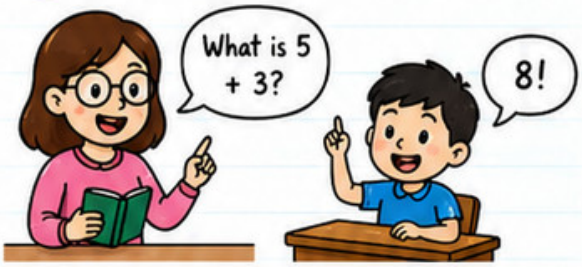
It does not usually need a separate compile step by the beginner.

2 Why It Is Called Runtime Execution



- Code runs.
- Values are used.
- Output appears immediately.
- Easy to test small programs.

3 Real-Life Classroom Example



Teacher asks a question in class and student answers immediately. This is like runtime / direct execution.

4 Languages in This Group

JavaScript

Used in websites for making pages interactive.

Python

Easy language for beginners. Used in many applications.

PHP

Used for web development (server-side).

5 Simple Hello Example

JavaScript	Python	PHP
console.log ("Hello");	print ("Hello")	<?php echo "Hello"; ?>

6 Quick Understanding



Memory Tip

Interpreted language = code runs directly and gives output at runtime.

Interpreted Languages

4 Languages in One Programming Notes

JS, Python, and PHP at Runtime

1 Same Idea

All three run code and give output at runtime.

2 JavaScript Runtime

```
console.log("Hello");
```

- Example:

```
console.log("Hello");
```

- JavaScript often runs in the browser.

3 Python Runtime

```
print("Hello")
```

- Example:

```
print("Hello")
```

- Python executes directly and is beginner-friendly.

4 PHP Runtime

```
<?php
  echo "Hello";
?>
```

- Example:

```
<?php echo "Hello"; ?>
```

- PHP runs on the server and sends output to the browser.

→ Hello

5 What Changes?

The command names and where they run are different.

6 What Stays Same?

All three execute instructions and produce output at runtime.

Quick Comparison

Language	Runtime Command Example	Where It Runs
JavaScript	`console.log("Hello");`	In the browser
Python	`print("Hello")`	On your computer (directly)
PHP	`<?php echo "Hello"; ?>`	On the server (sends to browser)

Memory Tip

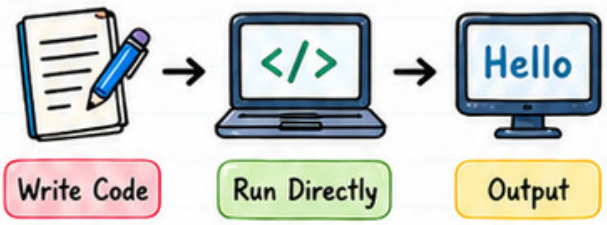
Different commands, different places, same result — output at runtime!

Interpreted Languages

4 Languages in One Programming Notes

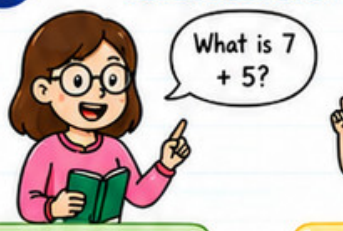
Tips, Flow, and Beginner Understanding

1 Runtime Flow



In interpreted languages, you write code, run it directly, and see the output immediately.

2 Classroom Example



Classroom
Teacher asks, student answers immediately.

=

Direct Runtime Execution
You run the program, and it gives output immediately.

3 Good Examples

JavaScript
```
console.log ("Hello");
```
Hello

Python
```
print("Hello")
```
Hello

PHP
```
<?php
echo "Hello";
?>
```
Hello

All show output "Hello" at runtime.

4 Common Beginner Confusion



Interpreted does not mean no rules!

- It still follows syntax rules.
- It usually runs directly without a separate beginner compile step.

5 Beginner Tip



1. Write a small program.
2. Run the code.
3. Check the output.
4. If error comes, fix it.
5. Run again and see the result!

6 Final Summary



JavaScript, Python, and PHP are common interpreted languages. They execute and show output at runtime.

Mini Comparison

Interpreted
= direct runtime execution.

VS

Compiled
= code is prepared first then run.

Memory Formula

Write → Run → Output

Interpreted language = direct runtime execution.

Operators Introduction

What are Arithmetic Operators?

1 What is an Operator?

An operator is a symbol used to do an action in programming.



2 What are Arithmetic Operators?

Arithmetic operators are used for mathematical calculations.



3 Five Main Arithmetic Operators

+	−	*	/	%
Addition	Subtraction	Multiplication	Division	Modulus (remainder)

4 Real-Life Understanding

10 + 5	10 - 5	10 * 5	10 / 5	10 % 3
= 15	= 5	= 50	= 2	= 1

5 Why We Use Them

- calculate values
- solve problems
- build logic
- use in almost every program



6 Quick Flow

Number (10, 5, 3, 20...) → Operator (+,-,*,/,%) → Result (15, 5, 50, 2, 1...)

7 Memory Tip

Arithmetic operators help us calculate values in a program.

+ add, − subtract, * multiply, / divide, % remainder

Operators Introduction

Rules, Tips, and Practice

1 Operator Meaning

+	add
−	subtract
*	multiply
/	divide
%	remainder

2 Good Examples

8 + 2 = 10

8 − 2 = 6

8 * 2 = 16

8 / 2 = 4

8 % 3 = 2

3 Common Beginner Confusion

- / means division, not a fraction line.
- % gives the remainder, not percentage.
- Do not confuse the symbols with each other!

4 Mini Practice

Solve these expressions:

1. 7 + 3 = ____
2. 9 − 4 = ____
3. 6 * 5 = ____
4. 15 % 4 = ____

5 Input to Output Flow

Number (Left) + Operator (+, -, *, /, %) + Number (Right)

↓

Result (Output)

6 Beginner Tip

Read expressions like this:

10 + 5

as "10 plus 5".

This helps you understand better!

7 Final Summary

- Operators perform actions on numbers.
- Use the correct operator for the correct job.
- Understand the result carefully.
- Practice regularly to become strong!

Practice makes perfect!
Try solving more problems every day.

Answers (Check Yourself)

1. 10
2. 5
3. 30
4. 3

Memory Formula:

+ add | − subtract | * multiply | / divide | % remainder

Operators Introduction

Arithmetic Operators in 4 Languages

1 Example Values

a = 10
b = 5



2 Same Idea

All 4 languages use arithmetic operators to perform calculations on numbers.

3 Operators Used

- `+` Addition
- `-` Subtraction
- `*` Multiplication
- `/` Division
- `%` Modulus (remainder)

4 Code in 4 Languages

JavaScript

```
let a = 10;
let b = 5;

console.log(a + b); // 15
console.log(a - b); // 5
console.log(a * b); // 50
console.log(a / b); // 2
console.log(a % b); // 0
```

C++

```
int a = 10;
int b = 5;

cout << a + b << endl;   // 15
cout << a - b << endl;   // 5
cout << a * b << endl;   // 50
cout << a / b << endl;   // 2
cout << a % b << endl;   // 0
```

Python

```
a = 10
b = 5

print(a + b)   # 15
print(a - b)   # 5
print(a * b)   # 50
print(a / b)   # 2.0
print(a % b)   # 0
```

PHP

```
$a = 10;
$b = 5;

echo $a + $b;   // 15
echo $a - $b;   // 5
echo $a * $b;   // 50
echo $a / $b;   // 2
echo $a % $b;   // 0
```

5 What Changes?

- Syntax (how we write code)
- Print statements
- Comments style
- Declaration style

But meaning is the same!

6 What Stays Same?

- Operators: +, -, *, /, %
- Logic: do math on numbers
- Results: same values

Meaning is same in all 4 languages!

Quick Comparison (using a = 10, b = 5)

Operation	Expression	Result
+ Addition	a + b	15
- Subtraction	a - b	5
* Multiplication	a * b	50
/ Division	a / b	2
% Modulus (Remainder)	a % b	0

8 Memory Tip

Meaning same. Syntax changes.