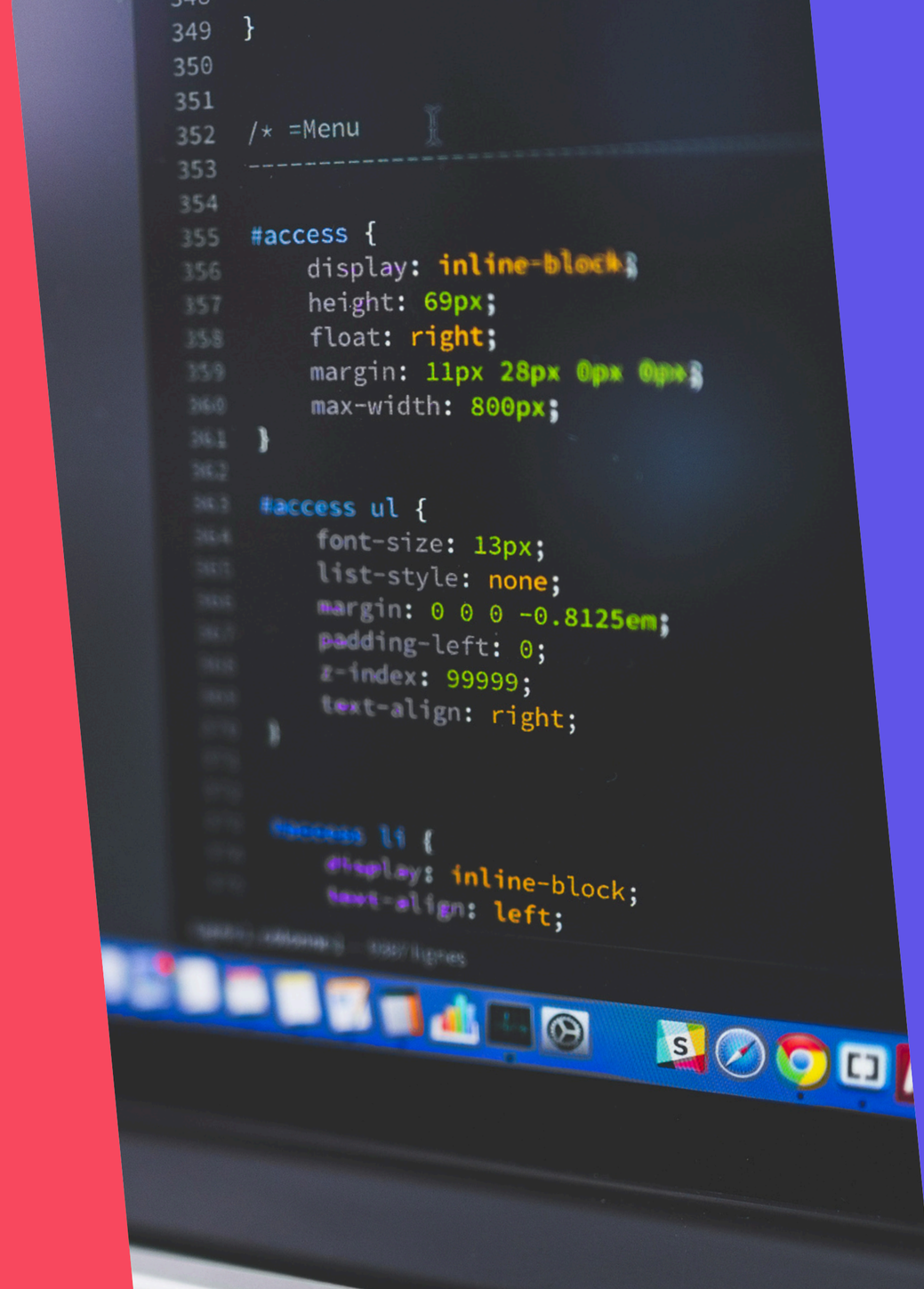


WBS CODING SCHOOL

Intro To Programming

WBS CODING SCHOOL

Agenda

- Introduction to Programming
- Variables and Data Types
- Programing concepts

Learning outcomes

WBS CODING SCHOOL

Skills you will acquire:

1.Understand what Programming is

2.Know the main programming concepts

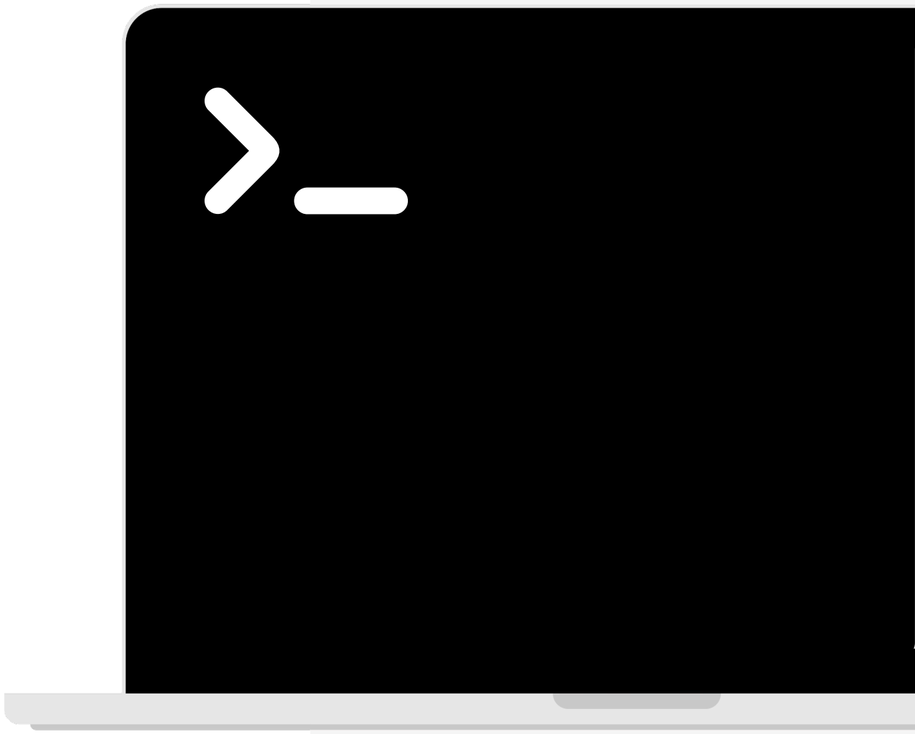
3. Outline the different data types

4. Understand what is pseudo code and why to use it

Chapter 2: Programming

Introduction to Programming

Understanding the Foundations of Writing Code



What is Programming?

Programming is the process of creating and organizing instructions for computers. It enables us to solve problems, automate tasks, and build software.

Goal - creating effective solutions

- Programming enables:
  - Automation
  - Data analysis
  - Interactive applications

Overview of Programming Languages

Languages differ in design, purpose, and structure.

Classifications:

- By level of abstraction
- By purpose
- By paradigm

Classification by Level of Abstraction

- **Machine Language**
  - Binary instructions (0s and 1s)
  - Directly executed by hardware
- **Assembly Language**
  - Symbolic representation of machine code
  - Hardware-specific
- **High-Level Languages**
  - Human-readable
  - Examples: Python, JavaScript, C#

Examples of High-Level Languages

- **Python:** Easy to learn; used in AI, web, and automation.
- **JavaScript**: Core technology for interactive websites.
- **C#:** Common in game development and enterprise software.

Classification by Purpose

- **General-Purpose Languages**
  - Versatile and broad use
  - Example: Python, Java
- **Domain-Specific Languages**
  - Tailored for specific tasks
  - Example: SQL, HTML, CSS

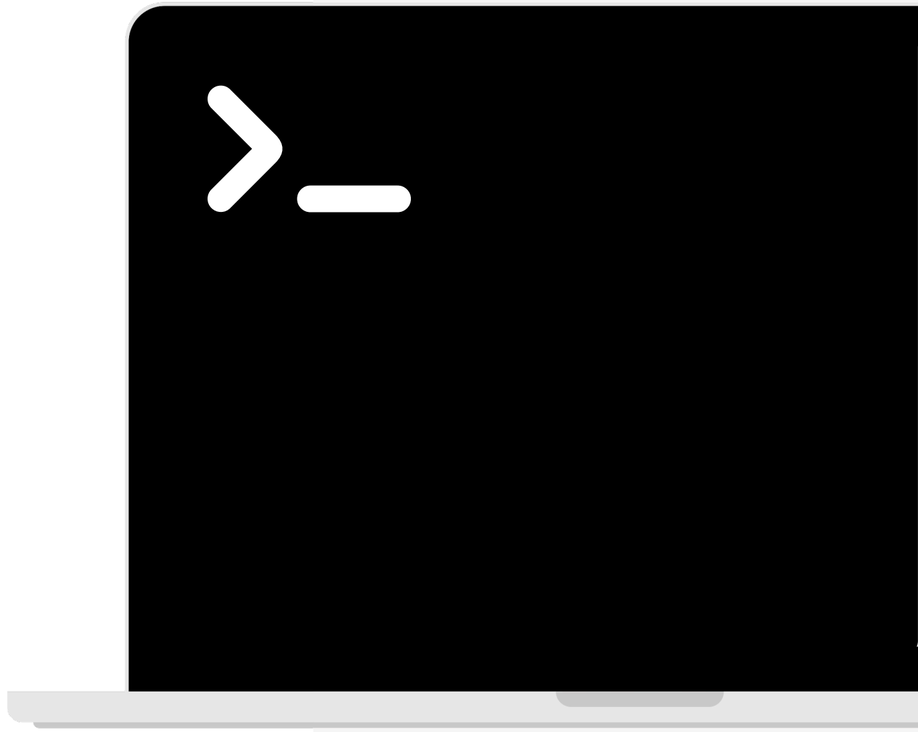
Classification by Paradigm

- **Procedural Programming**
  - Linear sequence of instructions
  - Example: C
- **Object-Oriented Programming (OOP)**
  - Code as reusable objects
  - Examples: Python, JavaScript, C#
- **Functional Programming**
  - Focus on immutability and pure functions
  - Examples: Haskell, Lisp

Chapter 2: Programming

Variables and Data Types

The Foundation of Data Handling in Code



Intro to Variables and Data Types

Variables and data types are core concepts in programming. They define how we store, label, and interact with information.

What is a Variable?

A variable is a container for storing data that can hold different values and be reuse.

Variables make programs dynamic and flexible.

Example Use Cases:

- Storing a user’s name
- Keeping track of the current time
- Holding a calculation result

Key Characteristics of Variables

- **Name** – a label for referencing data. Each language has its own naming conventions.
- **Value** – the actual data stored
- **Type** – the kind of data (e.g., number, text, boolean)

These elements define how a variable behaves in code!!

What are Data Types?

Data types define the kind of value a variable can hold and the operations that can be performed on it. Therefore:

- They help the program know how to store and process data
- Prevents bugs and logic errors

Some languages are strict about types while others allow type coercion.

Dynamic vs. Static Typing

- **Dynamically typed**
  - Languages like Python or Javascript
  - Type can change at runtime (x = 5, then x = "hi")
- **Statically typed**
  - Languages like Java or C#
  - Must declare type upfront (int x = 5)
  - Type can’t change later

Common Data Types

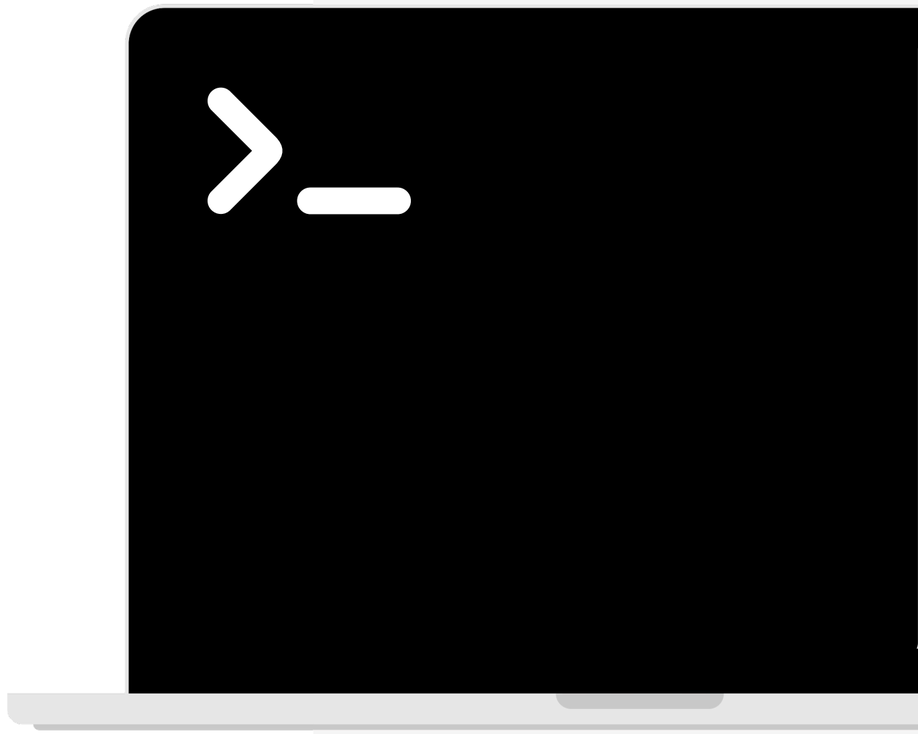
- **Integers:** Whole numbers (1, -3, 42)
- **Floats:** Decimal numbers (3.14, -0.001)
- **Strings:** Text values ("hello", "123", "Welcome!")
- **Booleans:** Truth values (True, False)

Each type has a purpose—use them wisely for correct logic and calculations.

Chapter 2: Programming

Programing concepts

Pseudo code, logic and syntax



Pseudo Code

Pseudocode is a powerful tool that encourages you to stop, break down the problem, and think critically about the solution. It acts as a bridge between understanding the problem and writing actual code.

The point is focusing on logic and structure rather than syntax.

Three steps of the process:

- Understanding the problem we want to solve
- Breaking down the solution into small steps
- Write the pseudo code using the solution steps as a guide

Programing concepts

- **Variables:** To store data or values.
- **Conditions:** To make decisions (e.g., if-else statements).
- **Loops:** To repeat actions (e.g., for or while loops).
- **Control Flow:** The sequence in which actions are executed.
- **Functions:** To organize reusable logic into modular blocks.

This concepts are very useful when writing pseudo code, as they are the basis of the logic in programming.

Syntax and Conventions

Once the pseudo code has been written, we can focus into the language we have choosen and translate it into actual code.

Each language will have its own syntax and specificities, as well as its own conventions, such as how to name variables.

WBS CODING SCHOOL

Lesson completed