**📝 PHP Notes**

**INITIAL SETUP (Oncer per new project)**

**Variable Types**

**📝 string →** A sequence of characters (e.g. "Hello")

**🔢 int →** Whole number (e.g. 42)

**💧 float →** Decimal number (e.g. 3.14)

**✅ bool →** true / false values

**📦 array →** Holds multiple values (e.g. ["red", "blue"])

**🧱 object →** An instance of a class (e.g. new Car();)

In PHP, **variables** are declared using the **$ dollar sign**, followed by the variable name.

php ● ● ●

$variableName = value;

**Scalar Types (Single Values)**

php ● ● ●

$string = "Mark"; // A sequence of characters (text)

$int = 12345; // An integer (whole number)

$float = 2.56; // A decimal number (floating point)

$bool = true; // A boolean (true or false)

📌 **Note:** If you wrap a number in quotes (e.g., "123"), it becomes a **string**, not a number.

**Array Types (Multiple Values)**

Arrays store **multiple values** in a single variable.

php ● ● ●

// Using array() function

$names = array("Jianxin", "Calcharo", "Encore");

// Using square bracket syntax (PHP 5.4+)

$names = ["Jianxin", "Calcharo", "Encore"];

⚠️ Square bracket syntax only works in **PHP 5.4 and above**.

In PHP, it's **good practice** to initialize variables with **default values** to avoid unexpected behavior or runtime errors — especially when you're not assigning them right away.

**🔐 Why Initialize with Defaults?**

* Prevents "undefined variable" errors
* Makes code predictable and safer
* Helps with debugging and conditional logic

**🧠 Tip from Dani Crossing:**

“Always initialize your variables to something — even if it’s an empty string, zero, or null. It prevents errors and makes your code more reliable.”

php ● ● ●

<?php

  //Scalar Types (Contains One Value)

  $string = "";

  $int = 0;

  $float = 0;

  $bool = false;

  $array = [];

  $object = null;

?>