

Day 7

Public & Private

In PHP classes, `public` and `private` are visibility modifiers for properties and methods.

- `Public` members can be accessed from anywhere, including outside of the class.
- `Private` members, on the other hand, can only be accessed from within the class itself.

```
class Car
{
    // Set default value for id
    private $id = "1";

    public $make;
    public $model;

    public function printData()
    {
        echo "Make: " . $this->make . "\n";
        echo "Model: " . $this->model . "\n";
        // Accessing private property within the class
        echo "Id: " . $this->id . "\n";
    }
}

// Creating an object of the Car class
$myCar = new Car();

// Setting public properties
$myCar->make = "BMW";
$myCar->model = "M3";

// Attempting to set private property will result in an error
//$myCar->id = "2";

// Printing car data
$myCar->printData();
```

Getters & Setters

Are methods used to **access (get)** and **modify (set)** the value of private or protected properties of a class. A getter method returns the value of a property, while a setter method updates the value of a property.

```
class Car
{
    // Default value for modelYear
    private $modelYear = 2023;

    // Getter for modelYear
    public function getModelYear() {
        return $this->modelYear;
    }

    // Setter for modelYear
    public function setModelYear($year) {
```

```

        $this->modelYear = $year;
    }
}

$car = new Car();
echo $car->getModelYear() . "\n";
// Outputs: 2023

// Updating modelYear using the setter method
$car->setModelYear(2024);

// Accessing the updated value using the getter
echo $car->getModelYear(); // Outputs: 2024

```

Constructors

A constructor is a special method within a class that is automatically called when an object of the class is created. It is commonly used to initialise class properties or perform startup procedures necessary for the class to function properly

- The constructor method is named `__construct()`.

```

class Car {
    public $color;
    public $model;

    // Constructor
    public function __construct($color, $make, $model) {
        $this->color = $color;
        $this->model = $make;
        $this->model = $model;
    }

    public function getCarInfo() {
        return $this->color . " " . $this->make . " " . $this->model;
    }
}

// Creating an object of the Car class
$myCar = new Car("Red", "BMW", "M3");

// Accessing a method to get car information
// Outputs: Red BMW M3
echo $myCar->getCarInfo();

```