



# Magic Method

## • `__get ( $property )`

- هي Magic Method بتننفذ تلقائياً لما تحاول توصل لخاصية (Property) مش موجودة أو خاصة (private/protected) في ال object.
  - تتيح لك التحكم في الوصول للخصائص بطريقة معينة، وغالباً تستخدم مع encapsulation لحماية البيانات.
- مثال :

```
1 class Car {
2     private $model = "Toyota";
3
4     // Magic method __get to access private/protected properties
5     public function __get($property) {
6         // Check if the property exists within the class
7         if (property_exists($this, $property)) {
8             return $this->$property;
9         }
10        return "Property does not exist";
11    }
12 }
13
14 $myCar = new Car();
15 echo $myCar->model; // Output: Toyota
16
```

- هنا، لما نحاول نوصل للخاصية \$model وهي \_\_get، private، هتتنفذ تلقائياً، وتجيب قيمة model.

## • \_\_set ( \$property , \$value )

- هي Magic Method بتننفذ تلقائياً لما تحاول تضبط قيمة خاصية مش موجودة أو خاصة (private/protected).
- تتيح لك التحكم في تغيير الخصائص بطريقة معينة.

مثال :

```
1 class Car {
2     private $model;
3
4
5     // Magic method __set to set values to private/protected properties
6     public function __set($property, $value) {
7         // Check if the property is "model" to allow setting its value
8
9         if ($property == "model") {
10             $this->model = $value;
11         }
12     }
13
14     // Method to retrieve the model value
15     public function getModel() {
16         return $this->model;
17     }
18 }
19
20 $myCar = new Car();
21 $myCar->model = "Honda"; // Using __set to set the model property
22 echo $myCar->getModel(); // Output: Honda
23
```

هنا، لما نحاول نغير قيمة \$model وهي \_\_set، private، هتتنفذ تلقائياً وتضبط القيمة بطريقة معينة.

## • ( ) \_\_clone

- بتنفيذ لما نعمل نسخة من كائن موجود باستخدام clone.
- بتسمح لك بتخصيص عملية النسخ، وتقدر تستخدمها لضبط القيم في ال object الجديد بطريقة معينة.  
مثال:

```
1 class Car {
2     public $model;
3
4     public function __construct($model) {
5         $this->model = $model;
6     }
7
8     // Magic method __clone to customize cloning behavior
9     public function __clone() {
10        // Modify the model value in the cloned object
11        $this->model = "Cloned " . $this->model;
12    }
13 }
14
15 $myCar = new Car("Toyota");
16 $clonedCar = clone $myCar; // Clone the object
17 echo $clonedCar->model; // Output: Cloned Toyota
18
19
```

هنا، عند نسخ \$myCar باستخدام \_\_clone، clone، هتغير قيمة \$model في النسخة الجديدة وتضيف كلمة Cloned قبلها.

## • ( ) \_\_toString

- \_\_toString بتنفيذ لما نحاول نطبع كائن كأنه string.
- مفيدة لو عايز تحدد رسالة معينة أو قيمة مخصصة عند طباعة ال object.

مثال:

```
1 class Car {
2     private $model;
3
4     public function __construct($model) {
5         $this->model = $model;
6     }
7
8
9     // Magic method __toString to specify what to display when the object is treated
10    as a string
11    public function __toString() {
12        return "Car model is: " . $this->model;
13    }
14 }
15
16 $myCar = new Car("Toyota");
17 echo $myCar; // Output: Car model is: Toyota
```

لما نستخدم echo على كائن \$\_\_toString، myCar بتنفيذ وترجع string توضح موديل السيارة.

## • ( ) \_\_invoke

- بتسمح لك تستخدم ال object كأنه دالة.
- مفيدة لو عايز الكائن يقوم بعملية معينة عند استدعائه كدالة مباشرة.

مثال:

```
1
2 class Car {
3     private $model;
4
5     public function __construct($model) {
6         $this->model = $model;
7     }
8
9     // Magic method __invoke to make the object callable like a function
10    public function __invoke() {
11        return "Starting the car model: " . $this->model;
12    }
13 }
14
15 $myCar = new Car("Toyota");
16 echo $myCar(); // Output: Starting the car model: Toyota
17
```

- لما نستخدم \$myCar() كدالة، \_\_invoke بتننفذ وتعرض رسالة توضح بدء تشغيل السيارة.