Magic Methods[¶](https://www.php.net/manual/en/language.oop5.magic.php#language.oop5.magic)

Magic methods are special methods which override PHP's default's action when certain actions are performed on an object.

**Caution**

All methods names starting with \_\_ are reserved by PHP. Therefore, it is not recommended to use such method names unless overriding PHP's behavior.

The following method names are considered magical: [\_\_construct()](https://www.php.net/manual/en/language.oop5.decon.php#object.construct), [\_\_destruct()](https://www.php.net/manual/en/language.oop5.decon.php#object.destruct), [\_\_call()](https://www.php.net/manual/en/language.oop5.overloading.php#object.call), [\_\_callStatic()](https://www.php.net/manual/en/language.oop5.overloading.php#object.callstatic), [\_\_get()](https://www.php.net/manual/en/language.oop5.overloading.php#object.get), [\_\_set()](https://www.php.net/manual/en/language.oop5.overloading.php#object.set), [\_\_isset()](https://www.php.net/manual/en/language.oop5.overloading.php#object.isset), [\_\_unset()](https://www.php.net/manual/en/language.oop5.overloading.php#object.unset), [\_\_sleep()](https://www.php.net/manual/en/language.oop5.magic.php#object.sleep), [\_\_wakeup()](https://www.php.net/manual/en/language.oop5.magic.php#object.wakeup), [\_\_serialize()](https://www.php.net/manual/en/language.oop5.magic.php#object.serialize), [\_\_unserialize()](https://www.php.net/manual/en/language.oop5.magic.php#object.unserialize), [\_\_toString()](https://www.php.net/manual/en/language.oop5.magic.php#object.tostring), [\_\_invoke()](https://www.php.net/manual/en/language.oop5.magic.php#object.invoke), [\_\_set\_state()](https://www.php.net/manual/en/language.oop5.magic.php#object.set-state), [\_\_clone()](https://www.php.net/manual/en/language.oop5.cloning.php#object.clone), and [\_\_debugInfo()](https://www.php.net/manual/en/language.oop5.magic.php#object.debuginfo).

**Warning**

All magic methods, with the exception of [\_\_construct()](https://www.php.net/manual/en/language.oop5.decon.php#object.construct), [\_\_destruct()](https://www.php.net/manual/en/language.oop5.decon.php#object.destruct), and [\_\_clone()](https://www.php.net/manual/en/language.oop5.cloning.php#object.clone), *must* be declared as public, otherwise an [E\_WARNING](https://www.php.net/manual/en/errorfunc.constants.php#constant.e-warning) is emitted. Prior to PHP 8.0.0, no diagnostic was emitted for the magic methods [\_\_sleep()](https://www.php.net/manual/en/language.oop5.magic.php#object.sleep), [\_\_wakeup()](https://www.php.net/manual/en/language.oop5.magic.php#object.wakeup), [\_\_serialize()](https://www.php.net/manual/en/language.oop5.magic.php#object.serialize), [\_\_unserialize()](https://www.php.net/manual/en/language.oop5.magic.php#object.unserialize), and [\_\_set\_state()](https://www.php.net/manual/en/language.oop5.magic.php#object.set-state).

**Warning**

If type declarations are used in the definition of a magic method, they must be identical to the signature described in this document. Otherwise, a fatal error is emitted. Prior to PHP 8.0.0, no diagnostic was emitted. However, [\_\_construct()](https://www.php.net/manual/en/language.oop5.decon.php#object.construct) and [\_\_destruct()](https://www.php.net/manual/en/language.oop5.decon.php#object.destruct) must not declare a return type; otherwise a fatal error is emitted.

Echo and print

echo and print are more or less the same. They are both used to output data to the screen. The differences are small: echo has no return value while print has a return value of 1 so it can be used in expressions. echo can take multiple parameters (although such usage is rare) while print can take one argument. echo is marginally faster than print.

In PHP, both echo and print are used to display output. The main difference is that echo is faster and can take multiple arguments, while print can take only one and returns a value (1), so it can be used in expressions. echo does not return anything. Due to its speed and flexibility, echo is more commonly used.

In PHP, a final class is a class that **cannot be extended** by any other class. This means no other class can inherit from it. A final method is a method that **cannot be overridden** in a child class. The final keyword is used to protect important code from being changed through inheritance or method overriding.

In PHP, there are three types of arrays: Indexed arrays - Arrays with a numeric index. Associative arrays - Arrays with named keys. Multidimensional arrays - Arrays containing one or more arrays.