

## CONSTRUCTORES Y DESTRUCTORES

### Constructores

Las clases que tengan definido un método constructor lo invocan en la creación de cada nuevo objeto.

Un constructor es útil para llevar a cabo la inicialización de propiedades del objeto antes de ser usado.

A partir de PHP5, el método constructor de una clase se define con la función `__construct()` ( En versiones anteriores se tomaba como constructor de clase el método con igual nombre que la clase )

```
void __construct ( [ mixed $args = "" [, $... ]] )
```

Los constructores padres no son llamados implícitamente si la clase hija define un constructor. Para ejecutar un constructor padre, se requiere invocar a `parent::__construct()` desde el constructor hijo. Si la clase hija no define un constructor, entonces se puede heredar de la clase madre como un método de clase normal (si no fue declarada como privada).

```
<?php
class ClaseA
{ function __construct() { print "En el constructor ClaseA\n"; }
}

class ClaseA1 extends ClaseA
{ function __construct()
{ parent::__construct();
  print "En el constructor ClaseA1\n";
}
}

class ClaseA2 extends ClaseA { } // hereda el constructor de ClaseA

$obj1 = new ClaseA(); // En el constructor ClaseA
$obj2 = new ClaseA1(); // En el constructor ClaseA y En el constructor ClaseA1
$obj3 = new ClaseA2(); // En el constructor ClaseA
?>
```

```
<?php
class A
{ function __construct() { $a = func_get_args(); $i = func_num_args();
    if ( method_exists( $this,$f='__construct'.'$i' ) )
        call_user_func_array( array($this,$f) , $a );
    }

    function __construct1($a1) { echo('__construct con 1 param llamado: '.$a1.'<br>'); }
    function __construct2($a1,$a2) { echo('__construct con 2 param llamados: '.$a1.', '.$a2.' <br>'); }
    function __construct3($a1,$a2,$a3) { echo('__construct con 3 param llamados: '.$a1.', '.$a2.', '.$a3.' <br>'); }
}
$o1 = new A('boletus'); // __construct con 1 param llamado: boletus
$o2 = new A('boletus','lepista'); // __construct con 2 param llamados: boletus,lepista
$o3 = new A('boletus','lepista','lactarius'); // __construct con 3 param llamados: boletus,lepista,lactarius
?>
```

## Destructores

Desde PHP 5 es posible definir en una clase un método destructor que se invocará tan pronto como no hayan referencias a un objeto de dicha clase (eliminación del objeto) o en cualquier otra circunstancia de finalización (terminación del script).

void **\_\_destruct** ( void )

Como los constructores, los destructores padre no serán llamados implícitamente. Para ejecutar un destructor padre, se deberá llamar explícitamente a **parent::\_\_destruct()** en el interior del destructor.

También como los constructores, una clase puede heredar el destructor de la clase madre si no implementa uno propio.

```
<?php
class ClaseA { function __construct() { $this->name="claseA"; }
                function __destruct() { print " destruyendo ... " . $this->name .'<br>' ; }
            }

class ClaseA1 extends ClaseA { function __construct() { parent::__construct();
                                                                    $this->email = $this->name. "@gmail.com";
                                                                    }
                                function __destruct() { parent::__destruct();
                                                                    print " destruyendo ... " . $this->email .'<br>' ;
                                                                    }
                                }

$unA= new ClaseA();
$unA1= new ClaseA1();

?>
```