PHP中的魔术方法总结 :\_\_construct, \_\_destruct , \_\_call, \_\_callStatic,\_\_get, \_\_set, \_\_isset, \_\_unset , \_\_sleep, \_\_wakeup, \_\_toString, \_\_set\_state, \_\_clone and \_\_autoload

**1、\_\_get、\_\_set**  
这两个方法是为在类和他们的父类中没有声明的属性而设计的  
\_\_get( $property ) 当调用一个未定义的属性时访问此方法  
\_\_set( $property, $value ) 给一个未定义的属性赋值时调用  
这里的没有声明包括当使用对象调用时，访问控制为proteced,private的属性（即没有权限访问的属性）  
  
**2、\_\_isset、\_\_unset**  
\_\_isset( $property ) 当在一个未定义的属性上调用isset()函数时调用此方法  
\_\_unset( $property ) 当在一个未定义的属性上调用unset()函数时调用此方法  
与\_\_get方法和\_\_set方法相同，这里的没有声明包括当使用对象调用时，访问控制为proteced,private的属性（即没有权限访问的属性）  
**3、\_\_call**  
\_\_call( $method, $arg\_array ) 当调用一个未定义的方法是调用此访求  
这里的未定义的方法包括没有权限访问的方法

**4、\_\_autoload**  
\_\_autoload 函数，它会在试图使用尚未被定义的类时自动调用。通过调用此函数，脚本引擎在 PHP 出错失败前有了最后一个机会加载所需的类。  
注意: 在 \_\_autoload 函数中抛出的异常不能被 catch 语句块捕获并导致致命错误。

**5、\_\_construct、\_\_destruct**  
\_\_construct 构造方法，当一个对象创建时调用此方法，使用此方法的好处是：可以使构造方法有一个独一无二的名称,无论它所在的类的名称是什么.这样你在改变类的名称时,就不需要改变构造方法的名称  
\_\_destruct 析构方法，PHP将在对象被销毁前（即从内存中清除前）调用这个方法  
默认情况下,PHP仅仅释放对象属性所占用的内存并销毁对象相关的资源.  
析构函数允许你在使用一个对象之后执行任意代码来清除内存.  
当PHP决定你的脚本不再与对象相关时,析构函数将被调用.  
在一个函数的命名空间内,这会发生在函数return的时候.  
对于全局变量,这发生于脚本结束的时候.如果你想明确地销毁一个对象,你可以给指向该对象的变量分配任何其它值.通常将变量赋值勤为NULL或者调用unset.  
  
**6、\_\_clone**  
PHP5中的对象赋值是使用的引用赋值，如果想复制一个对象则需要使用clone方法，在调用此方法是对象会自动调用\_\_clone魔术方法  
如果在对象复制需要执行某些初始化操作，可以在\_\_clone方法实现  
  
**7、\_\_toString**   
\_\_toString方法在将一个对象转化成字符串时自动调用，比如使用echo打印对象时  
如果类没有实现此方法，则无法通过echo打印对象，否则会显示：Catchable fatal error: Object of class test could not be converted to string in  
此方法必须返回一个字符串

在PHP 5.2.0之前，\_\_toString方法只有结合使用echo() 或 print()时 才能生效。PHP 5.2.0之后，则可以在任何字符串环境生效（例如通过printf()，使用%s修饰符），但 不能用于非字符串环境（如使用%d修饰符）。从PHP 5.2.0，如果将一个未定义\_\_toString方法的对象 转换为字符串，会报出一个E\_RECOVERABLE\_ERROR错误。  
  
**8、\_\_sleep、\_\_wakeup**  
\_\_sleep 串行化的时候用  
\_\_wakeup 反串行化的时候调用  
serialize() 检查类中是否有魔术名称 \_\_sleep 的函数。如果这样，该函数将在任何序列化之前运行。它可以清除对象并应该返回一个包含有该对象中应被序列化的所有变量名的数组。  
使用 \_\_sleep 的目的是关闭对象可能具有的任何数据库连接，提交等待中的数据或进行类似的清除任务。此外，如果有非常大的对象而并不需要完全储存下来时此函数也很有用。  
相反地，unserialize() 检查具有魔术名称 \_\_wakeup 的函数的存在。如果存在，此函数可以重建对象可能具有的任何资源。  
使用 \_\_wakeup 的目的是重建在序列化中可能丢失的任何数据库连接以及处理其它重新初始化的任务。  
  
**9、\_\_set\_state**  
当调用var\_export()时，这个静态 方法会被调用（自PHP 5.1.0起有效）。  
本方法的唯一参数是一个数组，其中包含按array(’property’ => value, …)格式排列的类属性。  
  
**10、\_\_invoke**  
当尝试以调用函数的方式调用一个对象时，\_\_invoke 方法会被自动调用。  
PHP5.3.0以上版本有效  
**11、\_\_callStatic**  
它的工作方式类似于 \_\_call() 魔术方法，\_\_callStatic() 是为了处理静态方法调用，  
PHP5.3.0以上版本有效  
PHP 确实加强了对 \_\_callStatic() 方法的定义；它必须是公共的，并且必须被声明为静态的。同样，\_\_call() 魔术方法必须被定义为公共的，所有其他魔术方法都必须如此。