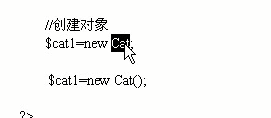
# 1.面向对象的创建



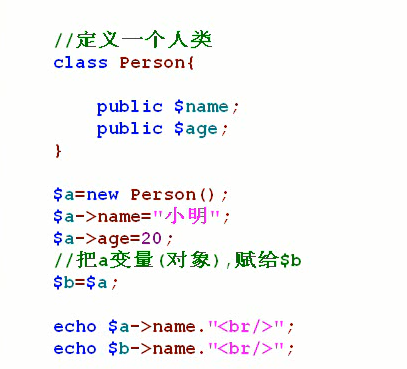
# 2.创建对象是后（）的问题

## 1.可以使用（），也可以不使用

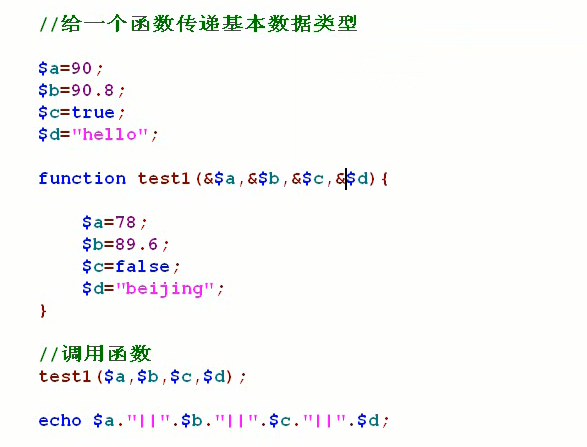


3.

# 3.对象是可以赋值的

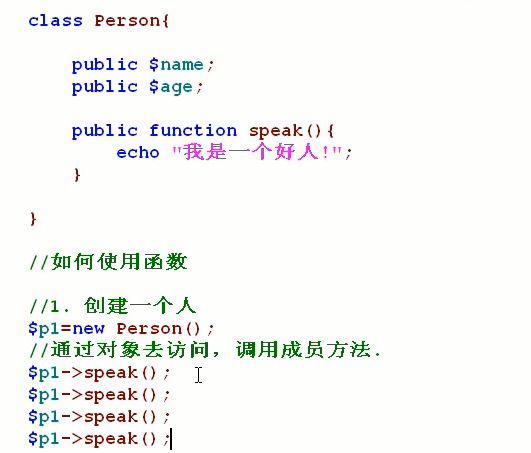


# 4.函数的参数是地址，还是变量，其实都是可以的（默认情况下，传入的是地址）

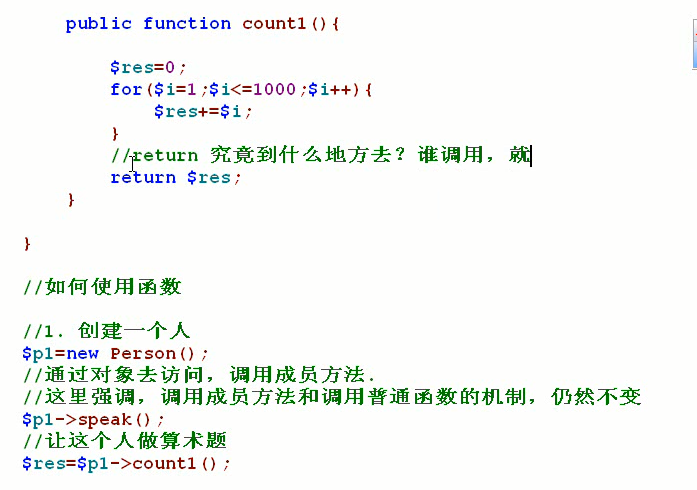


# 5.成员方法，如果将一个函数写到了一个类里面，我们这个函数就是成员方法

## 1.其实就像使用里面的 变量一样，$lei->speak();



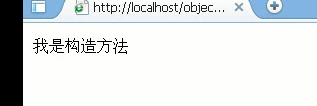
## 2.如果是有一个返回值，我们也是直接得到这个函数的返回值赋值过去就可以了



# 6.构造方法

## 1.php5中（和java中的不一样哦，这个是\_construct）



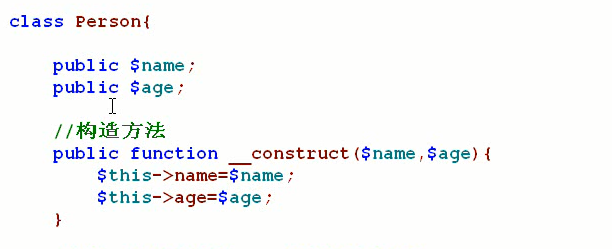
 2.但是在php4中像java中一样使用名字即可，但是优先选择\_construct



## 3.构造方法中参数的使用和java中是一样的

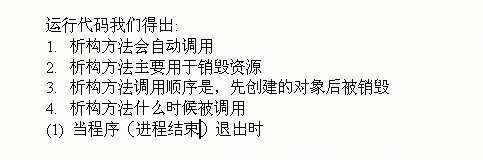
### 3.1比如在初始化类中的变量的时候，我们像java中一样使用this

$this->name= $name

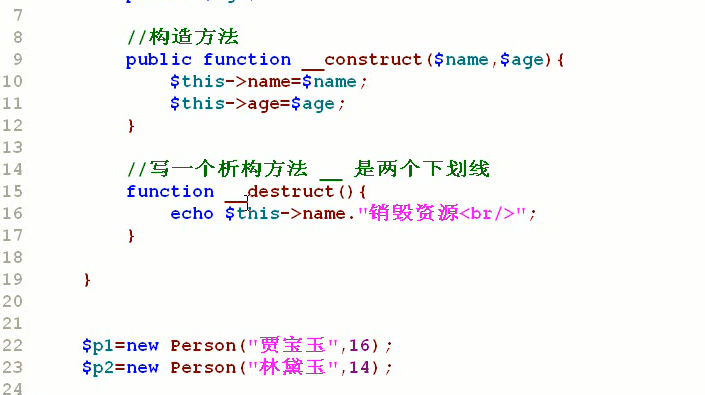


# 7.析构方法

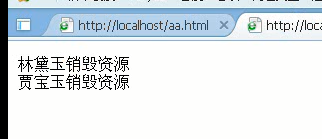
## 1.解释：析构方法主要用于释放资源，只要实例化了这个类，就会调用这个方法，（就是说先创建了对象，然后调用这个析构方法，先创建的对象后面被销毁，



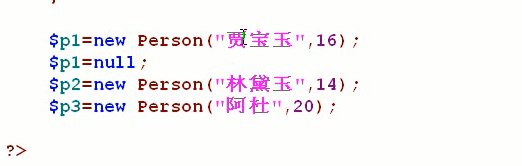
## 2.开始使用析构函数 function \_destrucrt(){，这个事写到了我们的类中的

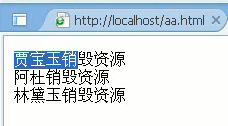


## 3.输出的结果是，注意是贾宝玉在后面，林黛玉在前面，后创建的先消灭

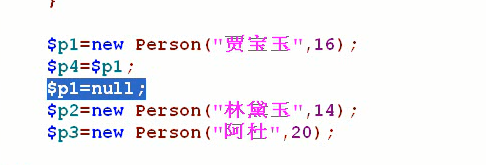


## 3.2但是如果我们先写成下面的东西的时候（就是垃圾对象的时候，马上就会别销毁）贾宝玉先销毁，然后是阿杜，最后是林黛玉



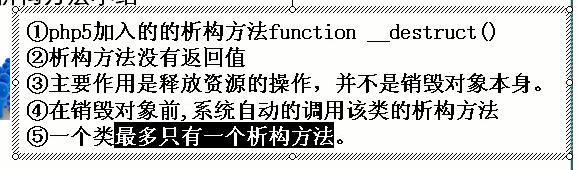


## 3.3但是如果我们写成下面的程序的时候，一定要注意的是，析构函数是对应的对象而言，只要对象还在就不会调用它，不管变量是不是空的，如下，



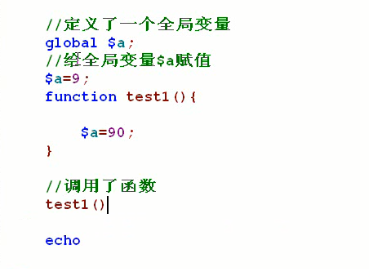


## 4.析构方法的总结，析构方法没有返回值，一个类主要是调用一个方法

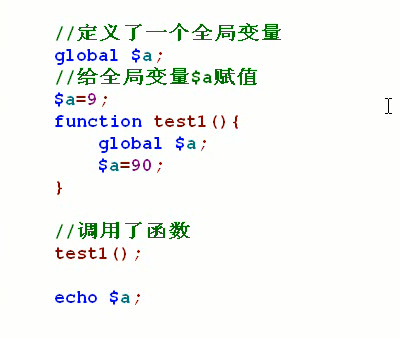


# 8.全局变量

## 1.直接使用（下面这个输出的是9）



## 2.函数中也要加入global



# 9.静态变量

## 一、定义 public static $num = 0;

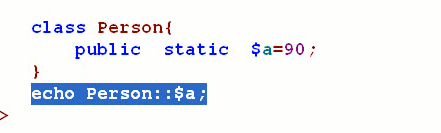
1.类中访问的方法一、使用 self::$cun

2.方法二 Child ::$sum

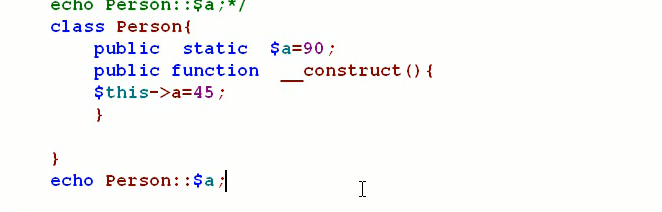
## 二、类外方法的时候只能使用Child ::$sum



## 三、总结，其实这个和 java中是一样的static之后直接就可以访问了，不需要实例化



### 3.2但是我们不可以使用this去访问静态变量

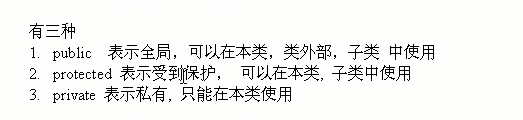


## 四、使用静态的方法

和java中是一样的，直接使用类去调用就可以了

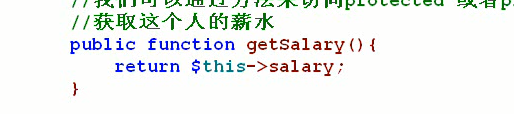


# 10 封装public protected private

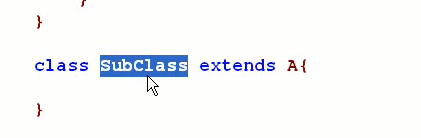


## 1.现在我们要访问pritect和private中的变量

对于protect中的东西，我们可以再放方法中添加一个方法用于得到我们的变量也是可以的



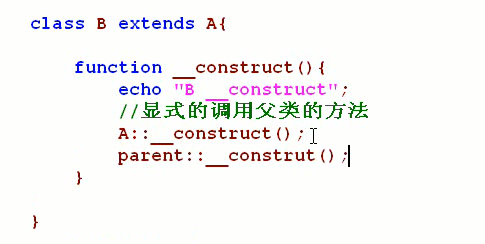
# 11.继承也是使用extends



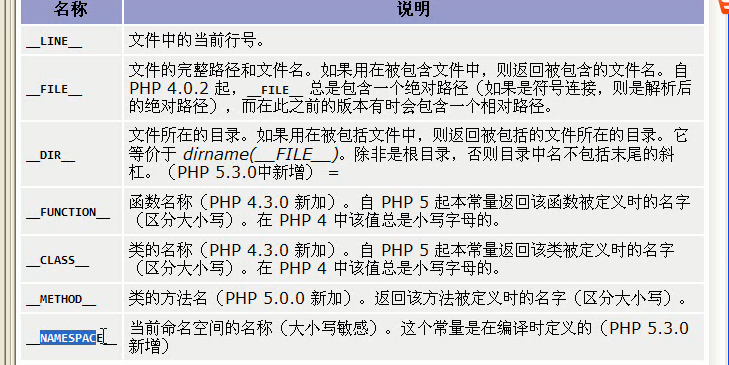
## 1.继承之后调用父类的方法

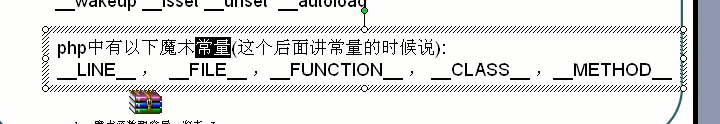
## 1.1A::\_construct()

## 1.2 parent::\_construct()



# 12.php中的一些魔术常量





## 1.然后我们先使用下面的魔术方法\_call,

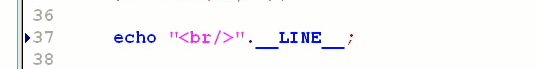
**1.2如果下面的方法，是不存在的下面的东西来就会将$method是一个方法名，$p就是一个我们里面的参数，它这个就是代表的一个数组的大小**



## 2.下面的常量任何的时候都可以使用的

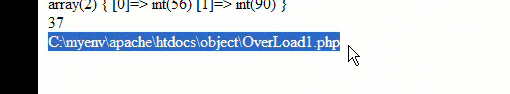


2.1\_LINE\_ 就是一个行号





2.2\_FILE\_表示的就是我们的是在哪个文件中

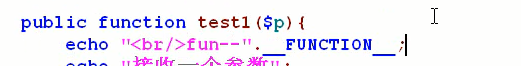


2.3\_\_DIR\_\_表示的就是文件的路径



2.4\_\_CLASS\_\_表示的是当前的类，如果不在当前的类里面，就不会显示了

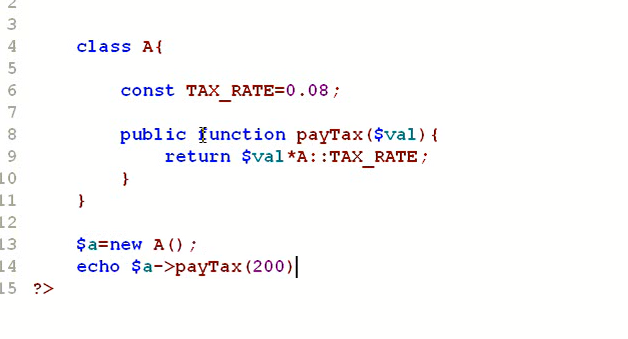
2.5\_\_FUNCTIN\_\_显示函数名



# 13.常量的表示不可以使用$，而且必须给出初始值

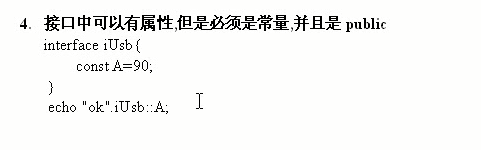
const A = 2





# 14.接口

## 1.下面是直接制作成，接口中变量的使用



2.