## 1.定义一个变量

1.1php中定义一个变量要以$符号打头

1.2$a =890; //我定义了一个变量，这个变量的名叫$a，然后给这个变量赋值为890

1.3变量的数据类型是可以变化 的

1.4显示变量的信息 var\_dump($a)

**echo** "zhang";

**echo** "hello" ;

$a=890;

var\_dump($a); //可以显示变量的信息

$a=1.1; //这样我们的额$a就编程了小小数类型

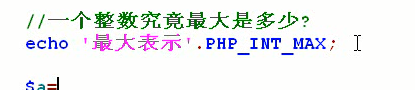
**echo** "<br>";

var\_dump($a); //可以显示变量的信息

?>

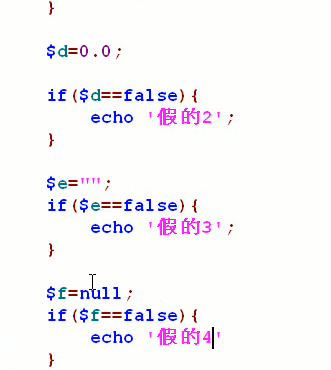
### 2.1php中的最大的值是，同时注意我们下面的字符加变量，如果超出了int的最大的数字，将会变成float

‘’.变量

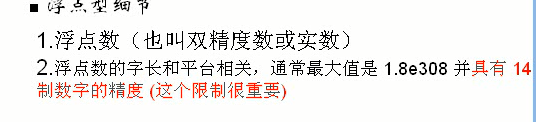


### 3.1布尔类型，可以把整形为0的值可以当最一个假来对待

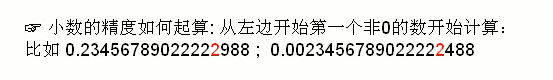




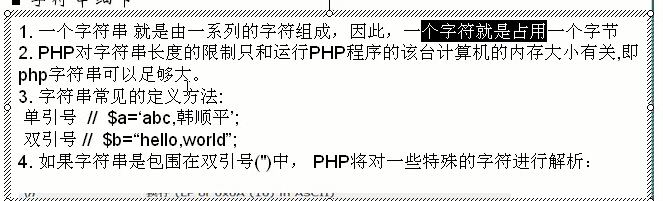
### 4.1浮点型，如果超出了14位 的时候是不会，也就是说从第一个非0的数字开始，只能输出14位之内的数字



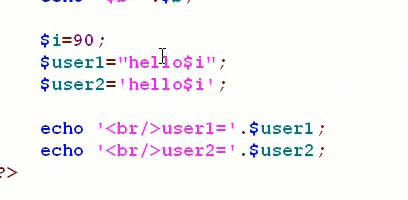
置于精度



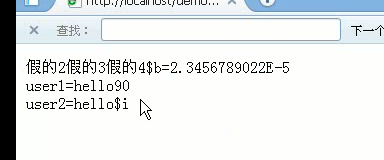
### 5.1字符串



#### 1.字符串 常见的定义方法有两种，一种是单引号，一种是双引号但是他们是有区别的，单引号什么都不用考虑，双引号的时候会考虑到变量，但是单引号可以使用‘’.变量



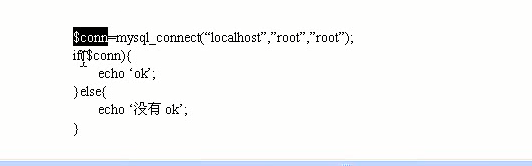
输入的结果是



# 2.php是可以和html混合使用的

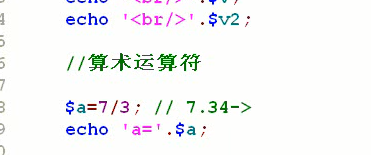
**echo** "<br>";

# 3.连接数据库（下面这个相当于是使用了布尔类型）



# 4.表达式

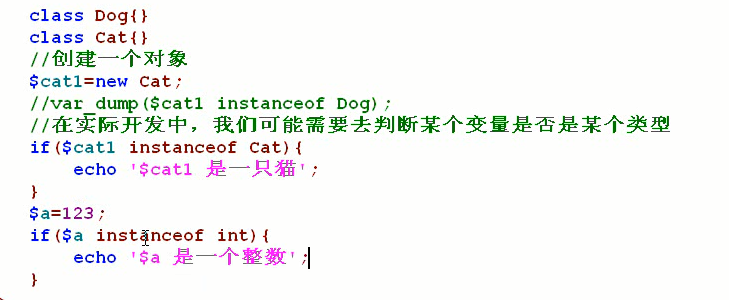
## 1. 除号，这样的话算出来的就是浮点型了

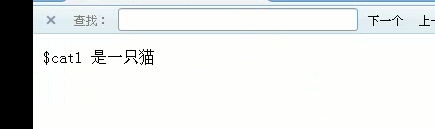




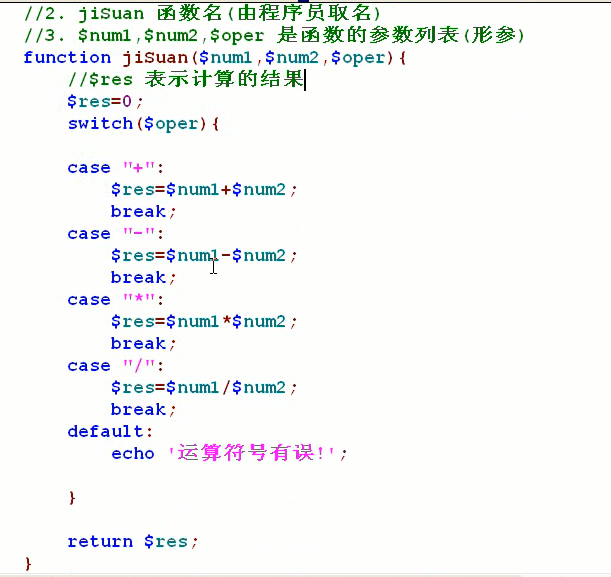
# 5.对象的判断，instanceof只能判断是不是某一类不能够判断其他的，下面的int是不能够判断的





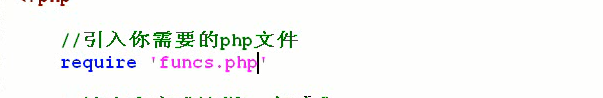


# 6.函数

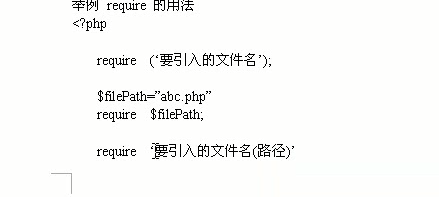


# 7. requrie引入另一个php文件(双引号和单引号都行，里面的其实是一个路径)

1.直接引用 require ‘funcs.php’



## 2.使用变量来引用（）



## 3.使用requrie\_once和requrie

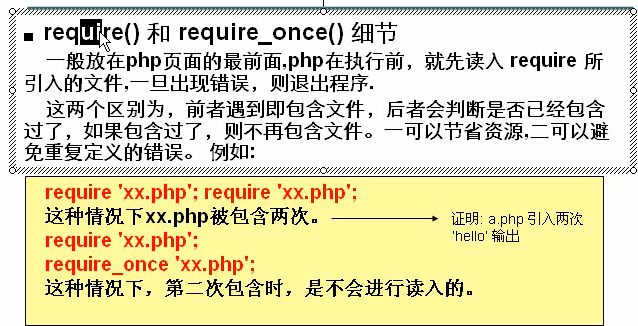
## 3.1相同点，引入文件后，一旦出现了错误，就会退出程序

## 3.2不同点，requrie不判断是不是引入了，requrie\_once判断是不是引入了，如果有了就不会再引入了，这样就节约了资源

如果我们使用了两个 require ‘b.php‘

require ‘b.php‘

这样会出现了错误，相当于我们将b.php中的东西引入了两次，（比如有两个相同的函数来到了这个文件中）



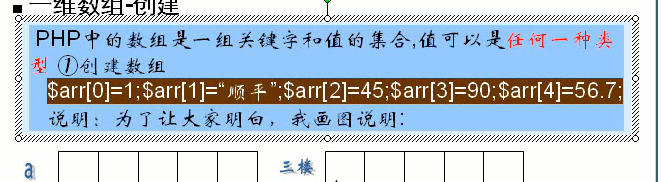
# 8.include和include\_once是一个文件包含在另一个文件中

其实这个和上面的require是是一样的了，只不过include遇到错误还会进行下去，相同点和不同点和require也是一样的

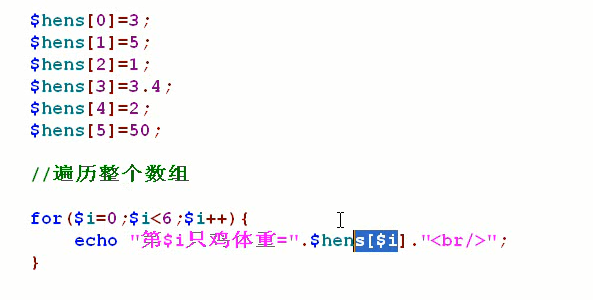
# 7和8的总结

显示中我们一般使用俄require\_once，因为我们使用了这个既节约了资源，而且也有利于错误的发现，如果使用include错误就不会检查了

# 9.数组$a[0],真尼玛活，先定义，后使用数组的大小



## 1.第一种创建数组方式



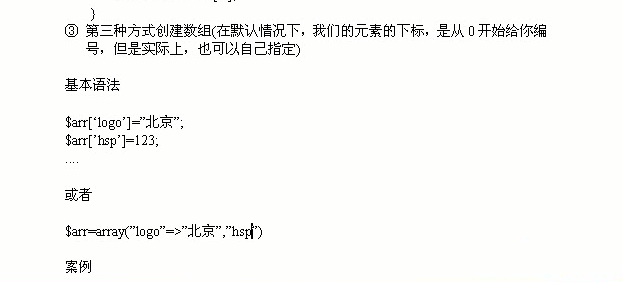
### 1.1数组的大小count($srr) ,count(数组名)，这样就是说明了数组是可以动态增长的



## 2.第二中创建数组的方式array（里面加东西）



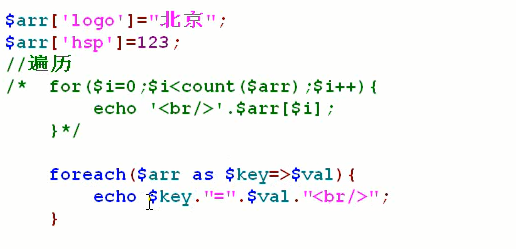
## 3.第三种创建数组的方式，数组下标自己定义，但是自己定义 的是不可以使用上面的for打印出来的我们不得不是用下面的方法forreach($arr as $val)



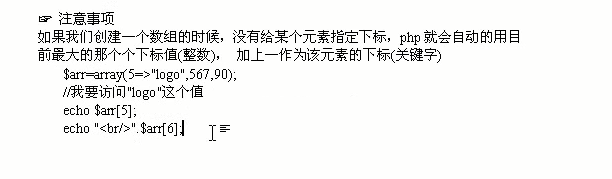


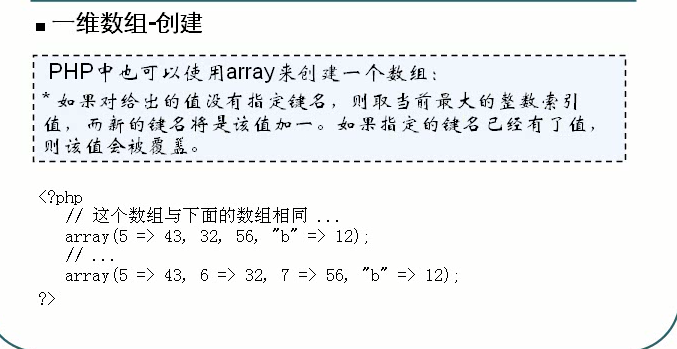


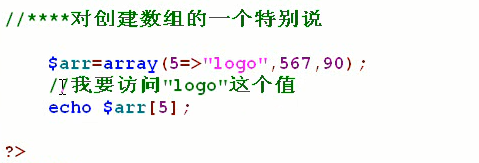
我们同时也想将我们自己定义的下标也打印出来，所以我们可以使用$key=>$val



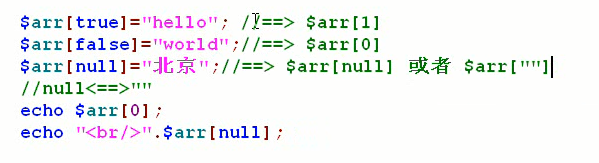
# 10数组的创建，5=>43,这样就是表示arr【5】，如果没有给定 数组的下标，按照





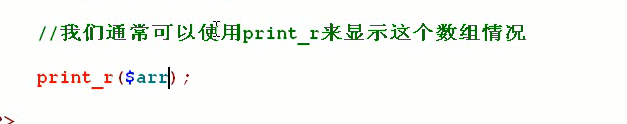


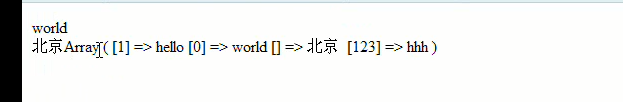
# 11数组中使用布尔关键字



# 12.使用函数来观察数组中的使用情况

## 1.使用print\_r($arr)



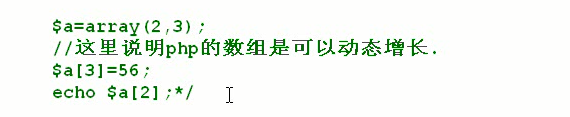


## 2.使用var\_dump($arr)，这个可以显示更多的信息

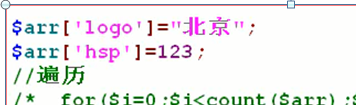


# 13.数组的注意事项

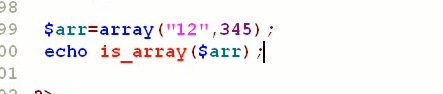
## 1.数组是可以动态增长的，观察下图

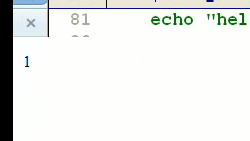


## 2.自己定义数组的时候，一定要用双引号，或者单引号否则有的时候会和变量可能重复，在变量没有的时候，php会自动它当成字符串

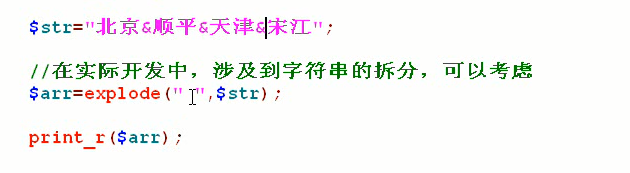


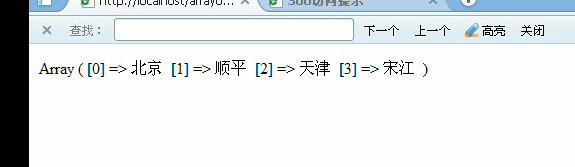
## 3. is\_array($atrr)判断是不是数组,如果是 数组的话，会打印输出1，否则什么也不输出



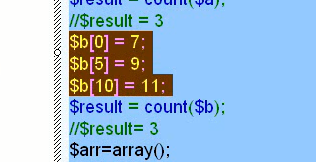


## 4.explode（“”，$str）拆分字符串，得到数组





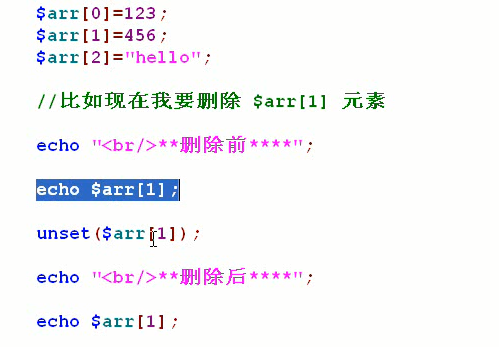
## 5.数组是以实际的数据来统计的，有几个是几个数组的大小

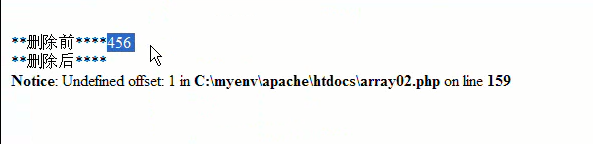


## 6.使用 forreach($arr as $val)，也可以输入正常的序列

具体操作和 自己定义的是一样的

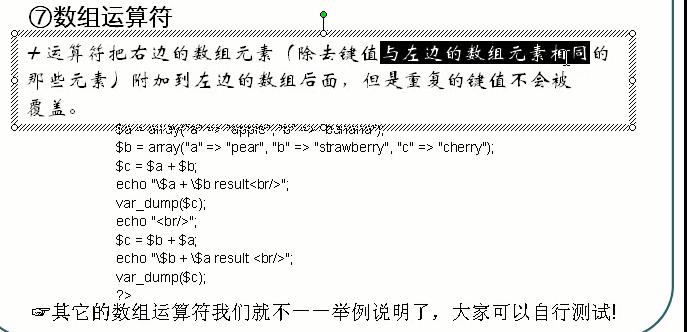
## 6.删除 unset($atrr[2]);



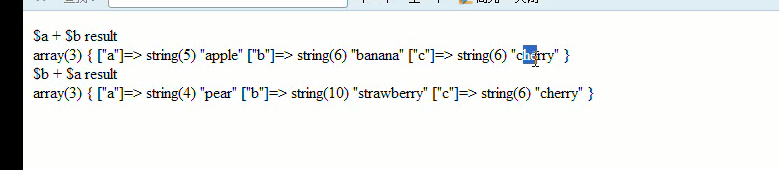


# 14. +数组运算符，只会将后面的东西夹到前面去

## 比如 $c = $a +$b;我们就是讲b中有的 但是a中没有的 添加到a中去，让他构成c，但是a中和b中一样的是不会改变的

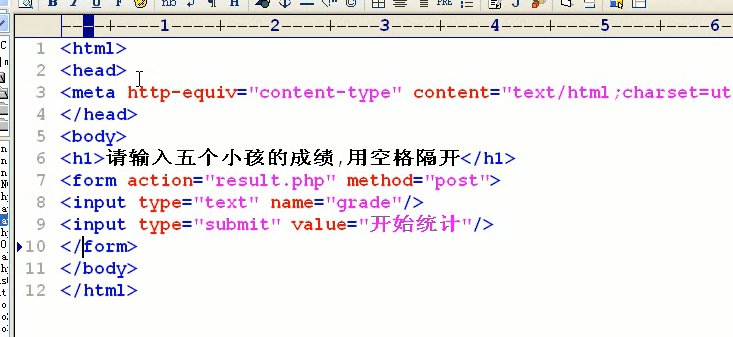




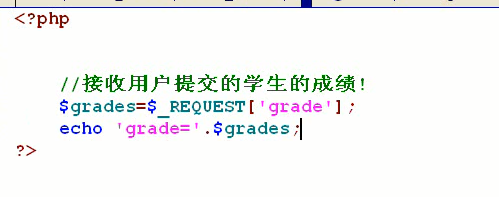


# 15.将一个文件的html和另一个php文件结合在一起，并且将html中的表单接收

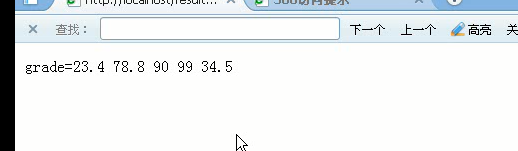
## 1.下面这个是html文件（其实也是一个php文件，这两个在同一个目录中），注意里面的action中写的是 result.php



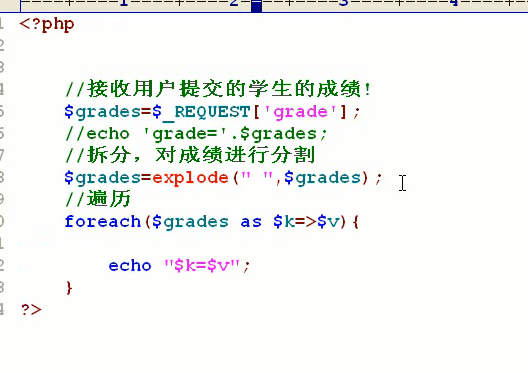
## 2.然后使用 $attr = $\_REQUEST(‘grade’);进行接收

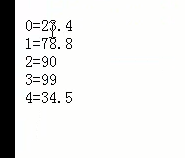


## 2.2.输出的结果就是我们输入的东西

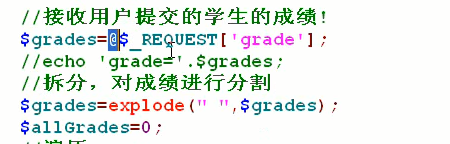


## 3.现在我们希望将它变成一个数组，使用explode

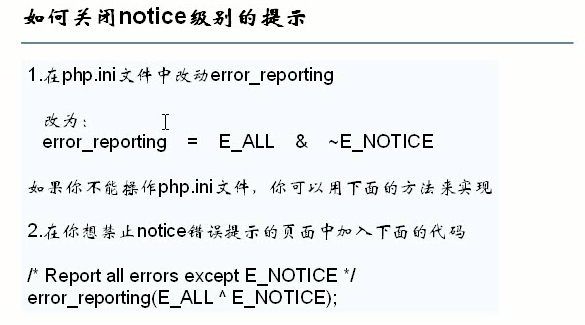




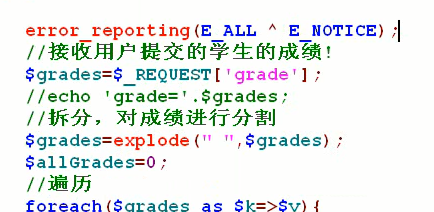
# 15.@运算符。出现错误不提示（只要不是致命的）



## 第二种，我希望所有的界面中显示级别低的错误



## 第三种想在某一个界面中，不显示级别低的错误



# 16.我们希望在html文件中显示php语句的内容

## 1.其实和jsp基本是一样的

