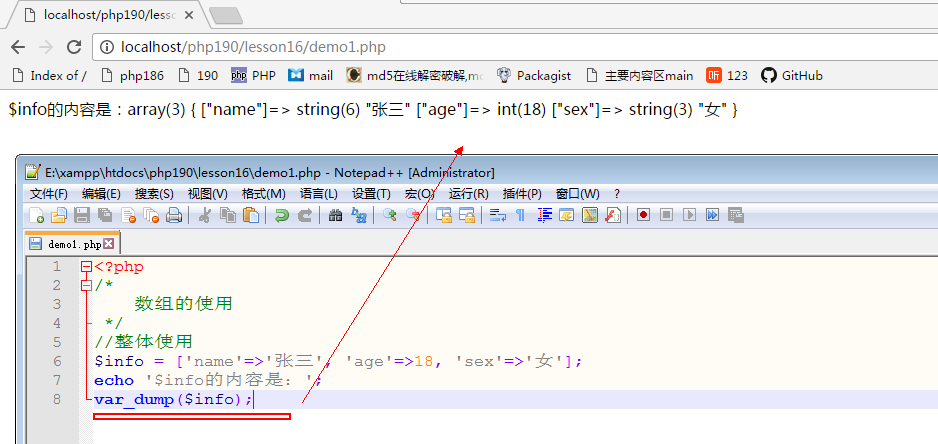
* 函数
  + 匿名函数
    - 没有名字的函数就是匿名函数，闭包
    - 通常情况下，把函数的定义赋值给变量
    - 调用的时候以可变函数的方式调用
  + 回调函数
    - 把某个函数当成另外一个函数的参数
  + 递归函数
    - 特点：自己调用自己的函数
    - 注意：必须有个临界条件，满足条件才自我调用，不满足就退出，否则死循环
    - 常用做：求斐波那契数列，递归统计目录大小，递归列出目录的文件信息，无限极分类等
  + 函数的函数
    - 函数的检测
      * Function\_exists(),返回bool
    - 已定义的函数
      * Get\_defined\_functions()
        + 返回数组
        + 两个键值对，系统预定义（1333），用户自定义
* 数组
  + 一组数，数的组合
  + 数组定义方式
    - 3种
      * $arr = array()
      * $arr[] = ‘’;
      * $arr = [];
  + 键值对
    - 键：下标，key
    - 值：value
    - 注意：
      * 1，如果不指定下标，默认从0开始，依次递增+1
      * 2，键值对之间使用逗号隔开，最后一个键值对后面的逗号可写，可不写
      * 3，键值的指向关系是：key=>value
      * 4,键是不能重复的，如果有重复，后面的值会覆盖前面的值
      * 5，键只能是标量，值是任意类型
      * 6，键如果不定义，就是前面最大的键的值加1，注意的是：不会找补空缺的值
  + 数组分类
    - 按照下标类型来分：
      * 索引数组
        + 下标从0开始，依次递增且连续的数组
      * 关联数组
        + 不是索引数组就是关联数组
    - 按照数组的层级来分
      * 一维数组
      * 二维数组
      * 多维数组
      * 注意：不是数array的数目，而是层级

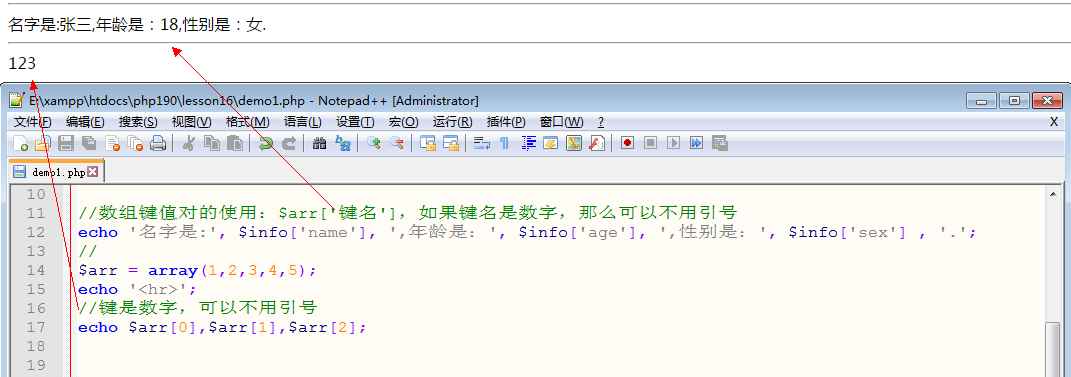
# 数组

## 数组的使用

* 整体使用
  + Var\_dump($arr)



* + Print\_r($arr)
* **使用具体的键值对**
  + **$arr[‘键名’]**

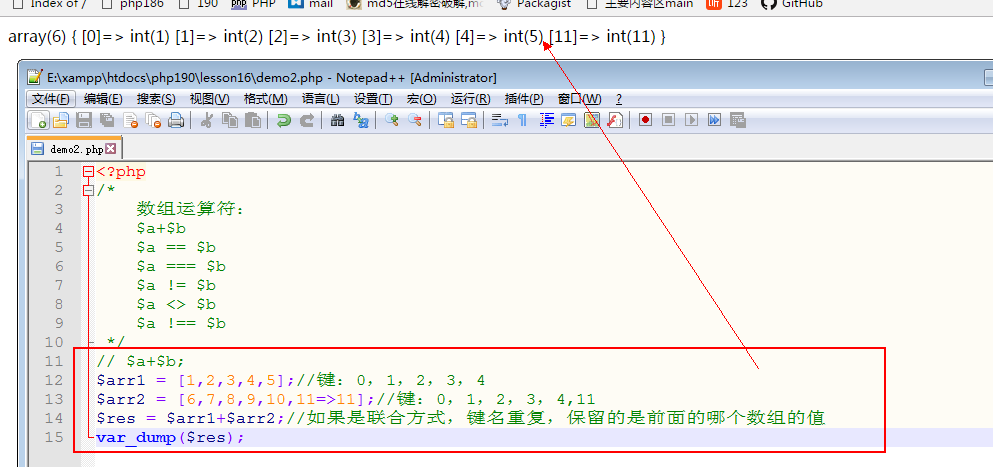


## 数组运算符



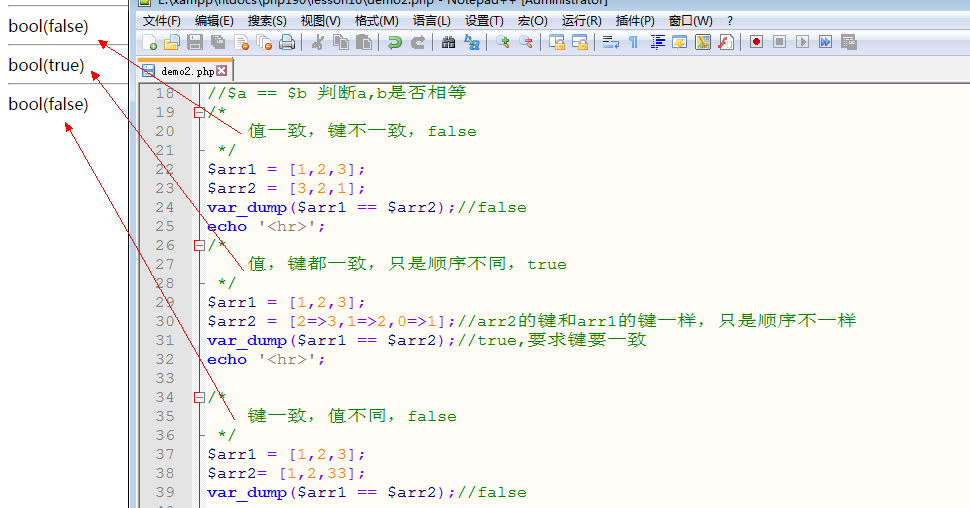
### $a+$b

* 作用：合并数组
* **注意：如果键名重复，最终的结果，重复的键对应的值保留的是前面的数组的值，而不是后面的数组的值。这一点，区别与：定义数组的时候，键名重复，保留后面的值**



### $a == $b

* 作用：判断$a,$b是否相等
* 结果是true的要求：
  + 1，键值都得一致
  + 2，顺序无所谓
  + 3.值的类型无所谓



### $a <> $b | $a != $b

* 作用：判断a,b是否不等
* 结果是true的要求：
  + 1，键值有一个不同就ok
  + 2，顺序无所谓

### $a === $b

* 作用：判断a,b是否全等
* 结果为true的要求：
  + 值，键，顺序必须都一致，包括值的类型
  + 键的类型无所谓，本身键也没有类型



### $a !== $b

* 作用：判断a,b不全等
* True的要求：
  + 只要键，值，顺序有一个不同，值的类型也不同，就是true

### 数组运算符要注意的有：

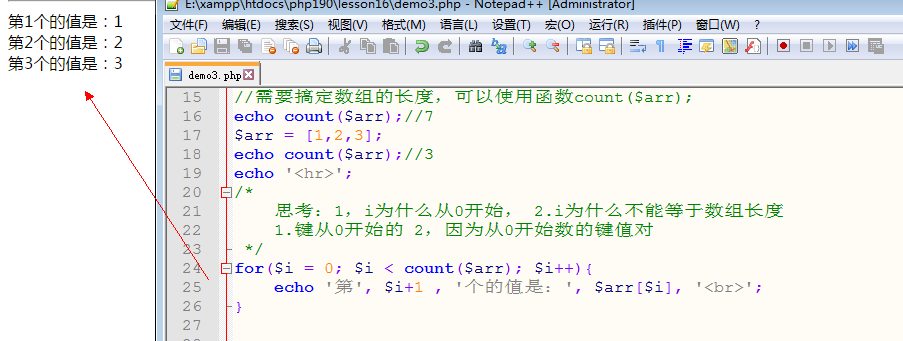
* 1.如果是结合|联合，键重复的话，保留前面的值
* 2.相等：判断键、值的大小、值的类型、键值对的顺序无所谓
* 3.全等：判断键、值的大小、值的类型、键值对的顺序都得一致

## 数组的遍历

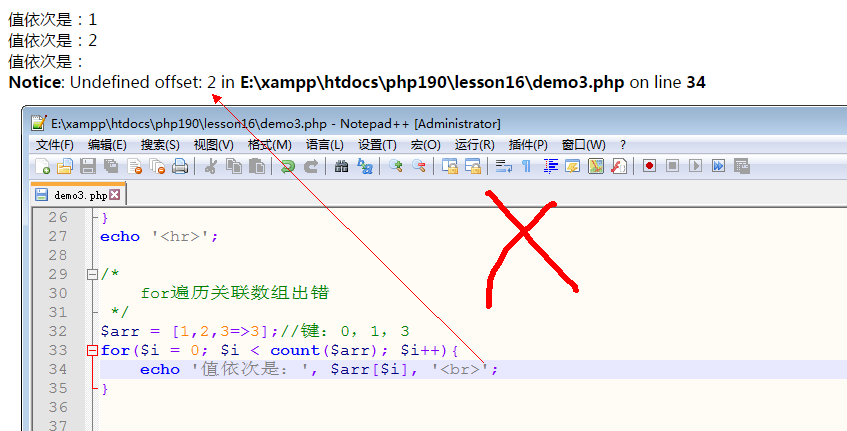
* 什么是遍历？说白了就是把数组中的键值对，挨个拿出来就是遍历。其实就是循环数组的键值对

### For

* 注意：
  + **1.i从0开始数**
  + **2，i不能等于数组长度**
  + **3，for只能遍历索引数组**



* 错误示例：for遍历关联数组



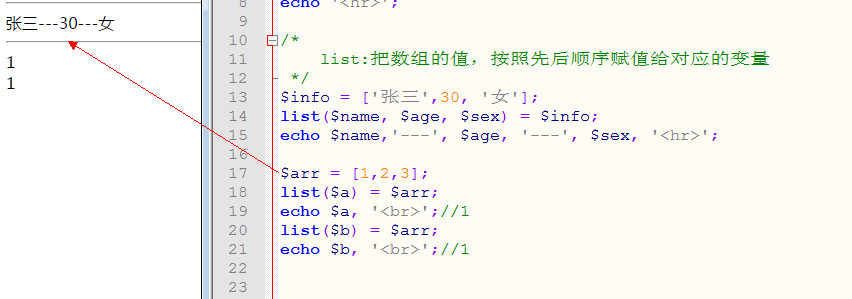
* **注意：for遍历索引数组的时候，效率最高**

### While...each...list组合

#### List:



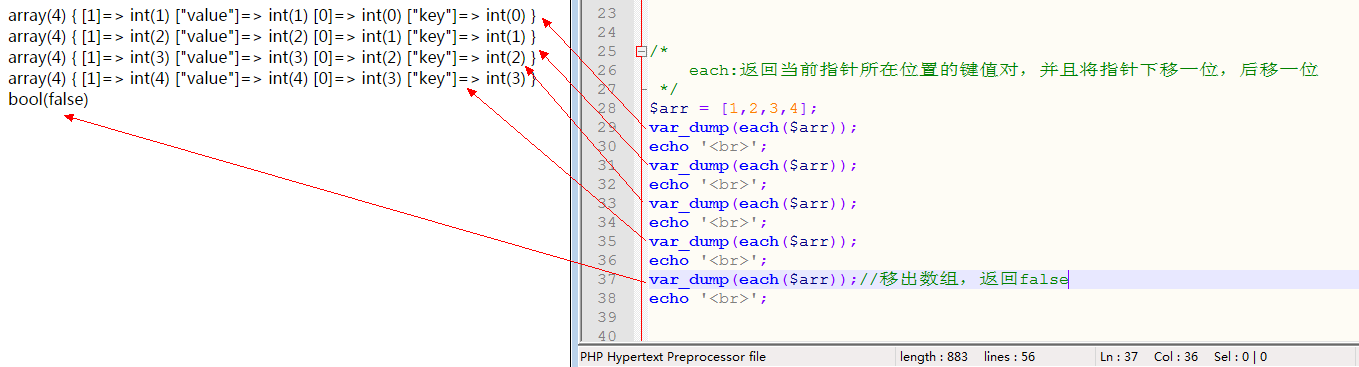
示例：



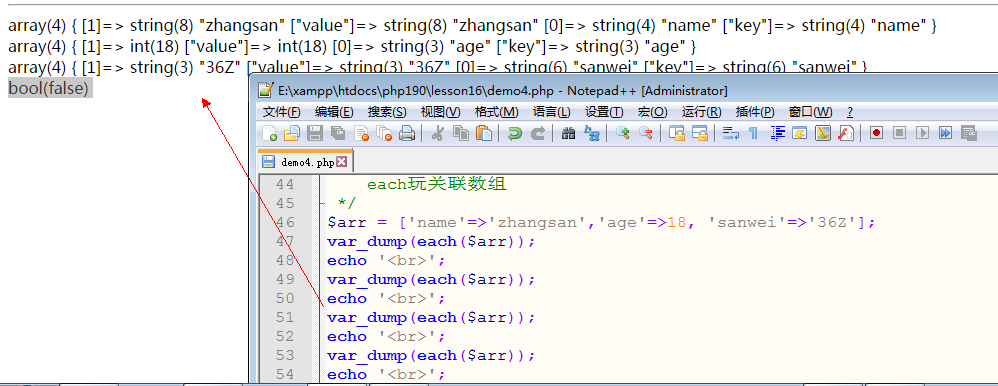
#### Each



示例：



示例2：

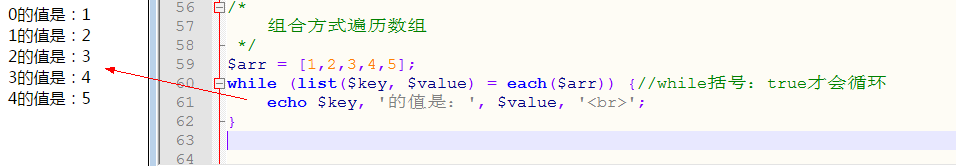


#### 组合方式遍历数组

**注意：**

1. **因为是使用each移动数组指针的方式遍历数组，所以可以遍历所有的数组。**
2. **组合方式遍历数组之后，还想继续玩，记得需要手动重置数组的指针，reset()**

示例1：



示例2:



### Foreach—推荐

* 格式：
* 1

**Foreach ($arr as $k=>$v) {**

**循环体**

**}**

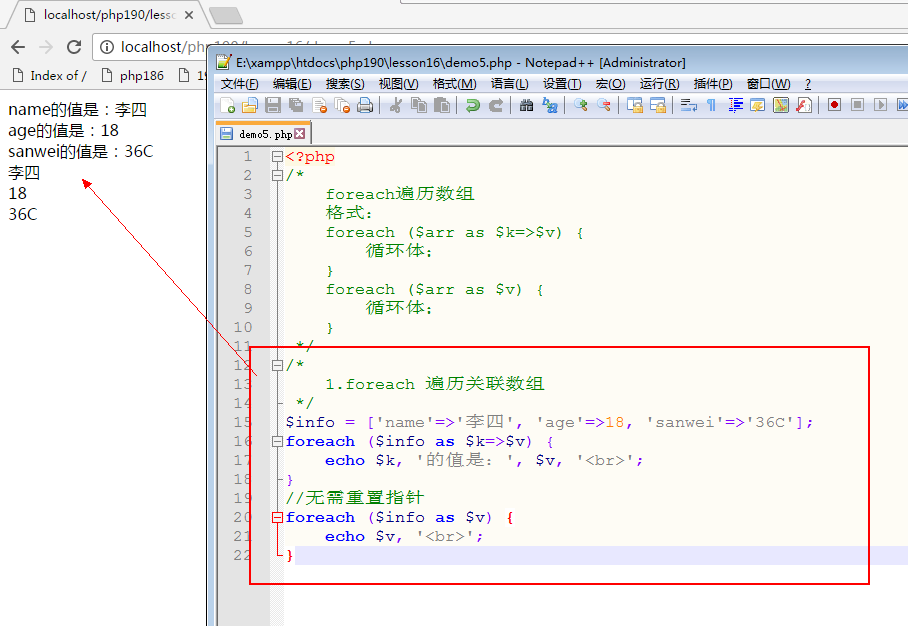
**2.**

**Foreach($arr as $v){**

**循环体**

**}**

* **原理：foreach遍历数组同样是移动数组指针来完成，不过每次foreach都会自动重置数组指针，不需要手动重置**
* **可以遍历所有的数组**
* **Foreach是推荐的遍历数组和对象的方法**



### 手动移动数组指针遍历数组

#### reset():重置指针到开头

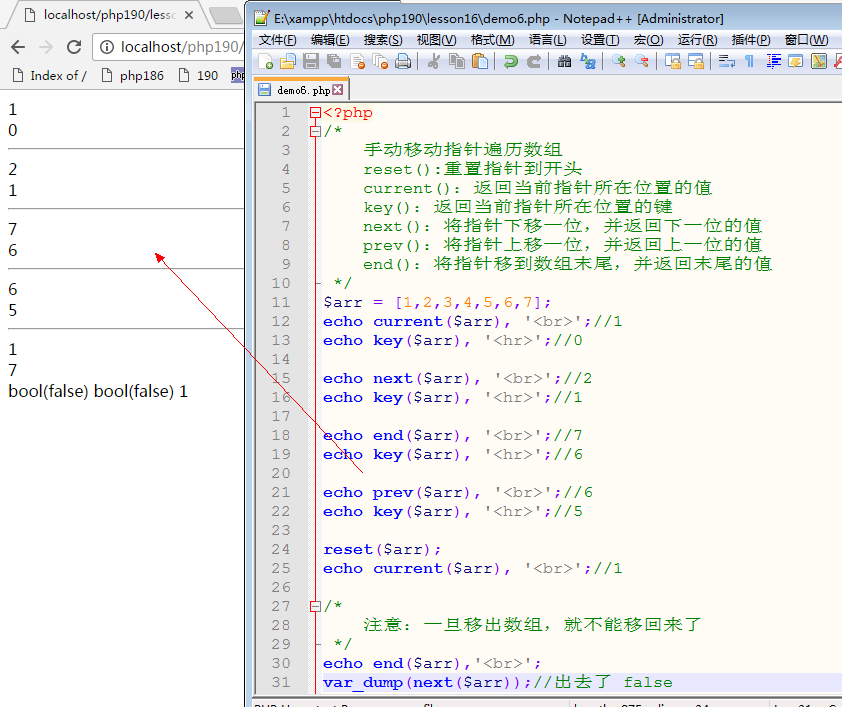
#### current()：返回当前指针所在位置的值

#### key()：返回当前指针所在位置的键

#### next()：将指针下移一位，并返回下一位的值

#### prev()：将指针上移一位，并返回上一位的值

#### end()：将指针移到数组末尾，并返回末尾的值



## 数组的函数

* 键值有关的
* 手动移指针的
* 分段&填充
* 压入&弹出的
* 排序
* 其他

### 小栗子



* **注意：如果是html代码中，想混入php代码，记得，后缀名一定是.php，而不能是.html**

