目录

[函数 2](#_Toc487020567)

[匿名函数|闭包 2](#_Toc487020568)

[定义格式： 3](#_Toc487020569)

[回调函数 3](#_Toc487020570)

[递归函数(难点) 4](#_Toc487020571)

[常见的函数的函数 4](#_Toc487020572)

[函数的检测 4](#_Toc487020573)

[Get\_defined\_functions() 6](#_Toc487020574)

[数组array 7](#_Toc487020575)

[定义方式-3种 7](#_Toc487020576)

[$arr = array(); 8](#_Toc487020577)

[$arr[] = ‘’; 8](#_Toc487020578)

[$arr = []; 8](#_Toc487020579)

[特点 9](#_Toc487020580)

[Echo不能输出数组 9](#_Toc487020581)

[Print\_r()输出数组 9](#_Toc487020582)

[键值对 9](#_Toc487020583)

[数组的分类—人工分的，计算机本身不搭理 10](#_Toc487020584)

[按照下标的格式： 10](#_Toc487020585)

[按照数组的层级 11](#_Toc487020586)

[数组的遍历 12](#_Toc487020587)

[数组的函数 12](#_Toc487020588)

[预习 12](#_Toc487020589)

* 函数
  + 分类
    - 系统预定义函数
    - 用户自定义函数
    - 如果某个功能系统已经有函数，就直接用系统函数就可以了，没 必要自己再写
  + 定义格式

Function 函数名 (形参1，形参2，形参N) ｛

函数体；具体的功能

｝

* + 使用
    - 函数名(实参1，实参2，实参N)；
    - 函数名使用的时候不区分大小写，但是一定要和定义的一致
  + 命名规则
    - 命名法：小驼峰
    - 1.可以使用字母，数字，下划线
    - 2，数字不能开头
    - 3，下划线开头慎用
    - 4，命名要有意义
    - 5，函数名不能重复，和常量一样
  + 形参&实参
    - 形参：定义函数的时候用到的参数|变量就是形参
    - 实参：实际调用函数的时候用到的参数|变量就是实参
  + 变量范围
    - 局部变量
      * 函数内部定义的变量
    - 全局变量
      * 函数外部定义的变量
    - 因为函数在执行的时候，会进入额外的空间，相对于全局空间独立的。在函数执行完成后，退出的那一刻，所有的局部变量都会销毁
    - 默认情况下，局部不能使用全局，全局也不能使用局部
  + 局部变量使用全局变量的方法（2种）：
    - 1，以实参的方式传值
    - 2，使用global关键词，
      * 注意：使用了global关键词之后，相当于局部变量是全局变量的引用。也就是说：局部或全局对变量值的修改，会影响的其他的变量
  + 全局变量使用局部变量的方法：
    - Return返回
  + 函数的返回值—return
    - 1.return可以将函数内部的东西返回到全局，抛出的东西需要变量接收
    - 2.return之后，函数就结束，return后面的代码不执行
    - 3，意味着，一个函数内，同一时间只能有一个return生效
  + 形参&实参数目问题
    - 默认情况下，实参和形参数目是相等的，并且顺序是从左到右一一对应.
    - 实参数目>形参
      * 多的实参丢弃
    - 实参= 形参
      * 默认的
    - 实参 < 形参
      * 默认，报错，丢失某个参数
      * 处理方式：
        + 可以给形参设置默认值
        + 注意：使用默认值的形参，必须往后放
    - 实参数目不定
      * 需要借助系统预定义的3个函数
        + Func\_get\_arg():返回某个实参
        + Func\_get\_args()：返回所有实参
        + Func\_num\_args()：返回实参的数目
    - 形参数目为0
  + 可变函数
    - 可变函数：函数的格式是变量名后面加个括号的方式
      * $变量名(实参1，实参2，实参N)；
      * 具体执行哪个函数，取决于变量的值。忘了的，看看昨天的计算器
    - 可变变量：有多个$符号的变量就是可变变量
  + 静态变量
    - 默认情况下，函数执行完成后，局部变量销毁。能否不销毁，留下来？可以，就是静态变量
    - 格式：就是在局部变量的前面加个static关键词

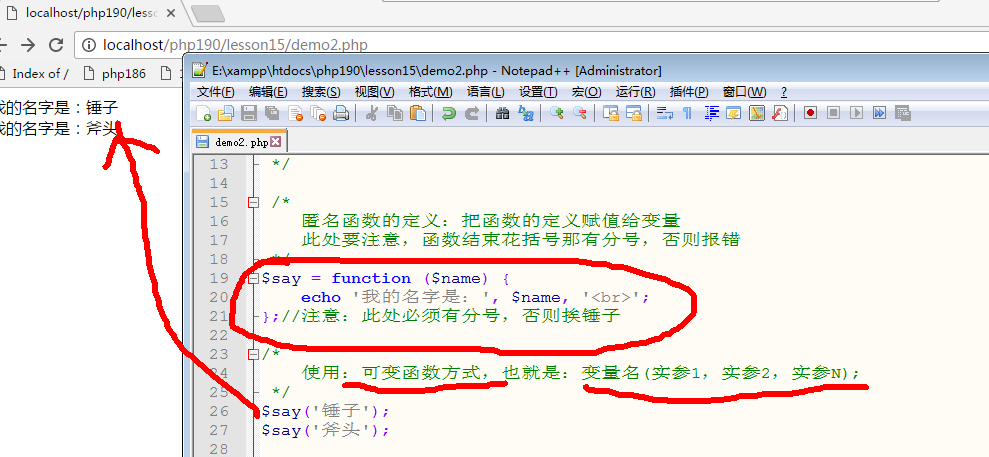
# 函数

## 匿名函数|闭包

* 没有名字的函数就是匿名函数

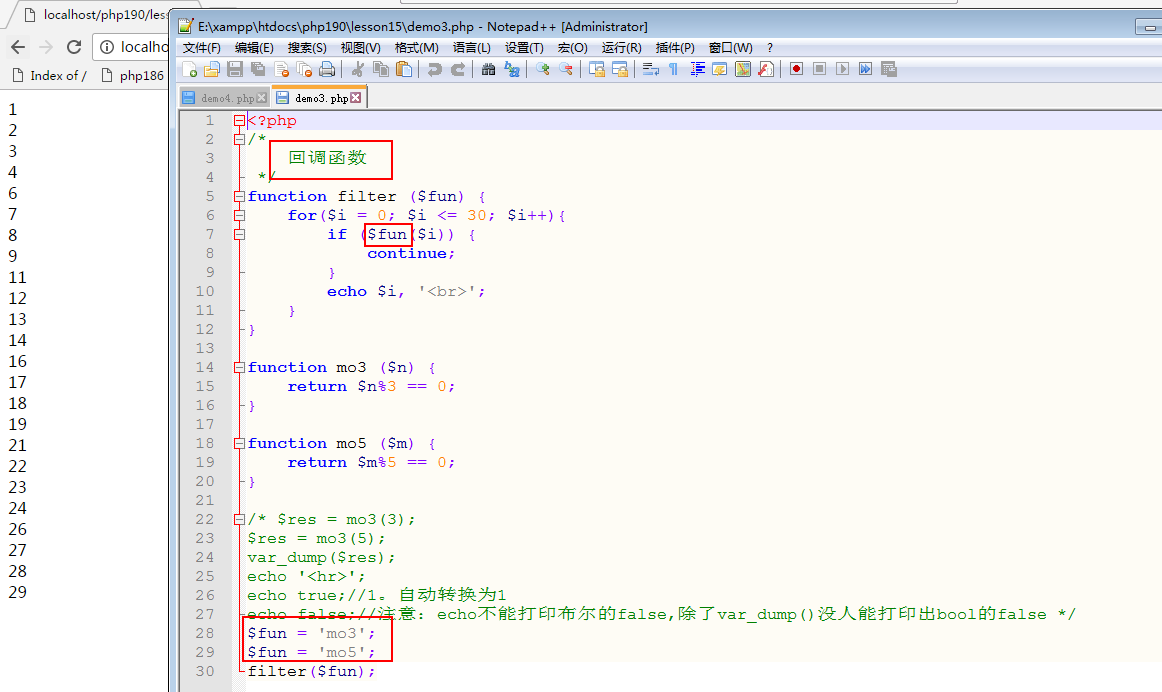
### 定义格式：

* **把函数的定义格式赋值给变量。**
* **注意：函数结束花括号后面必须有分号**



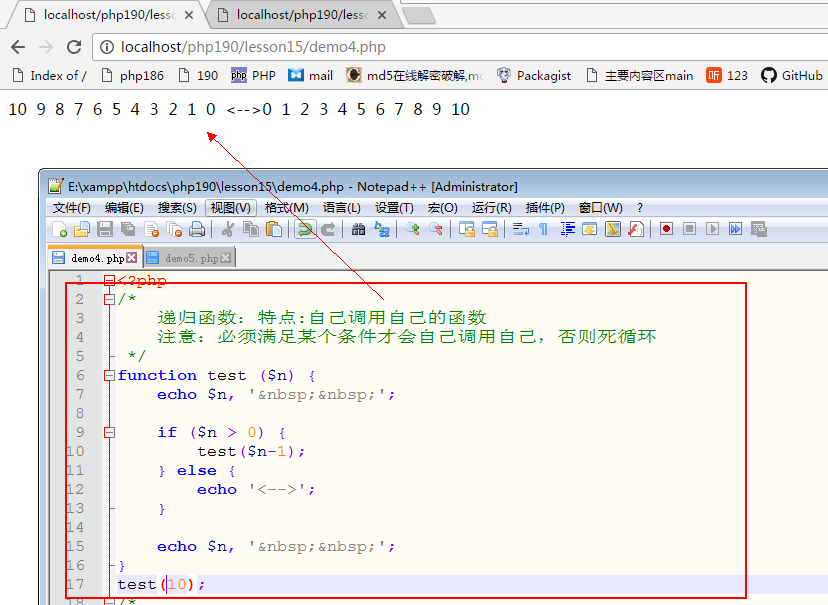
## 回调函数

* 1.可变函数
* 2.将函数的计算结果作为另外函数的值



## 递归函数(难点)

* 特点：自己调用自己的函数
* 条件
  + 满足特定条件下，才会自己调用自己，不满足就退出。否则的话，自己调用到死。类似循环



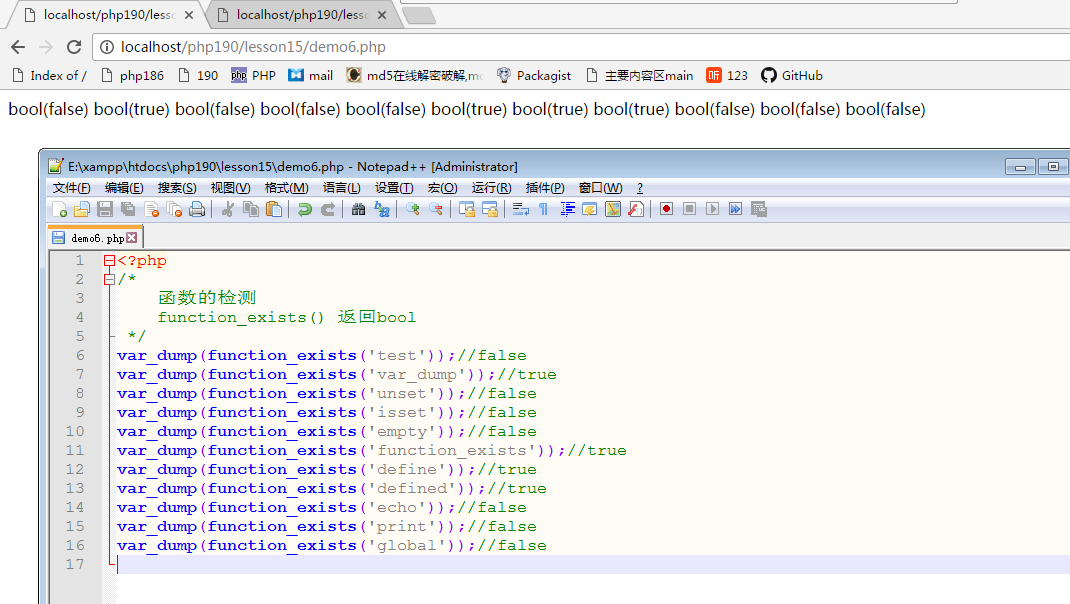
* 具体分解详解demo4,demo5。下来多看看，多交流

## 常见的函数的函数

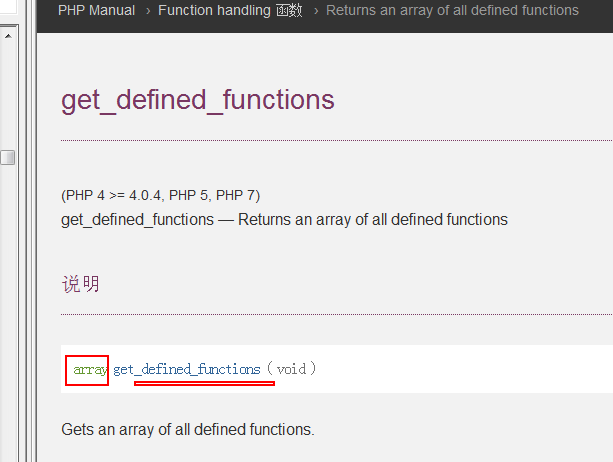
### 函数的检测

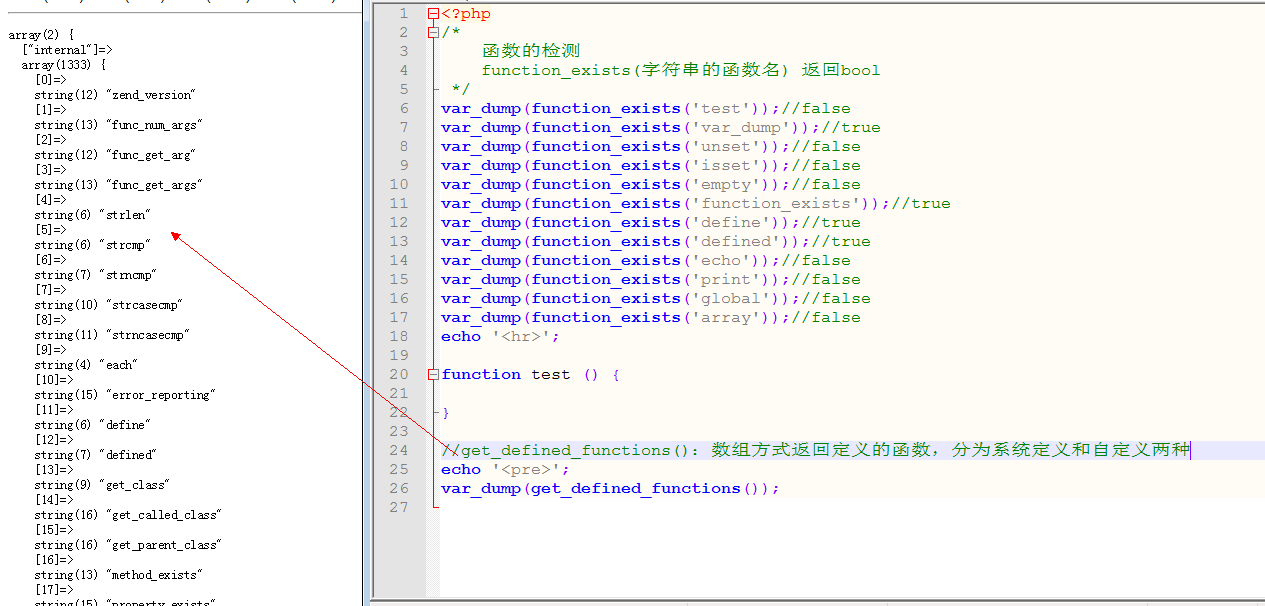
* Function\_exists()
  + 返回bool





### Get\_defined\_functions()



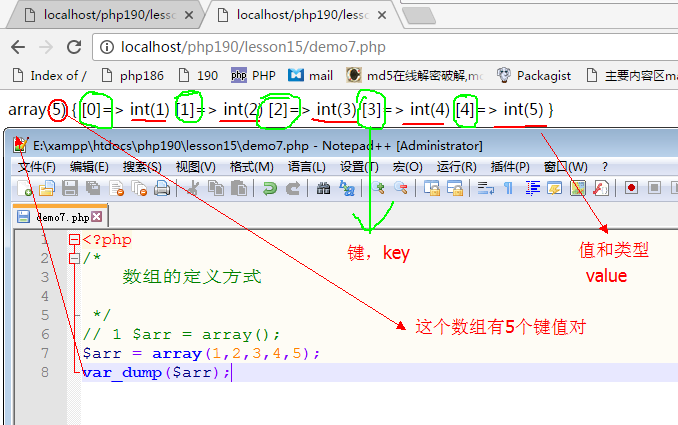


* 扩展：
  + 使用递归实现斐波那契数列，给个N，求出N的值
  + 斐波那契数列特点：
    - 1，1，2，3，5，8，13，21...
    - 从第三个数开始，他的值是前两个数的和

# 数组array

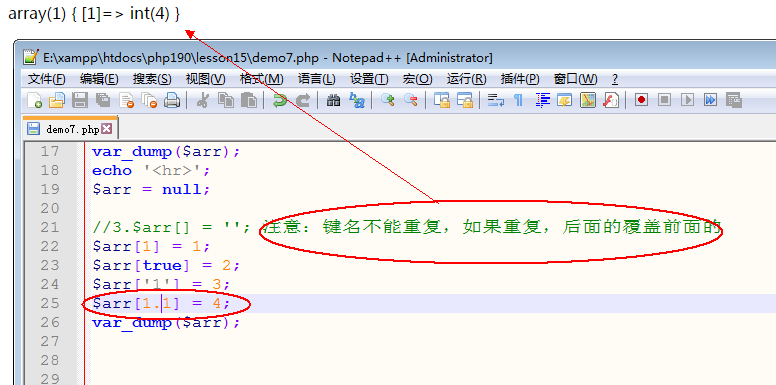
* 一个变量类型，复合类型
* 一组数，就是一系列数据的集合
* 数组由键值对组成

## 定义方式-3种

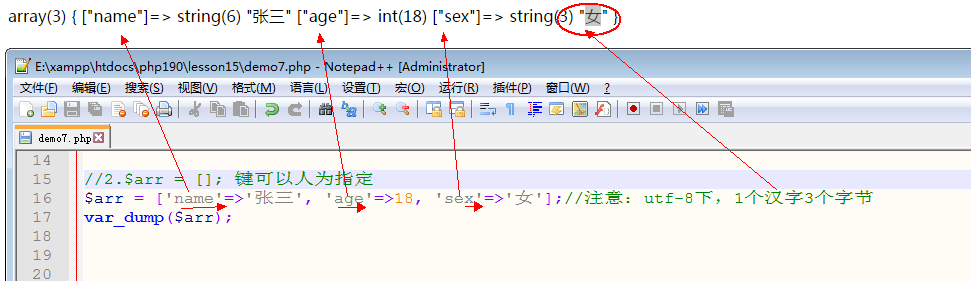


### $arr = array();

### $arr[] = ‘’;

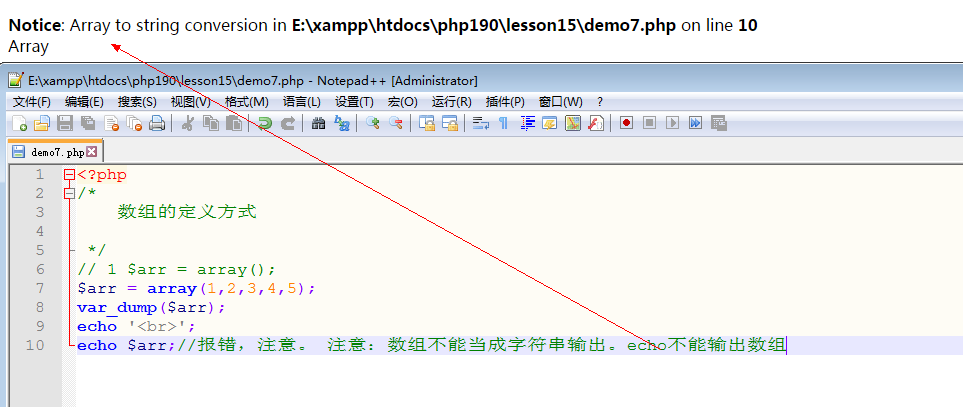


### $arr = [];

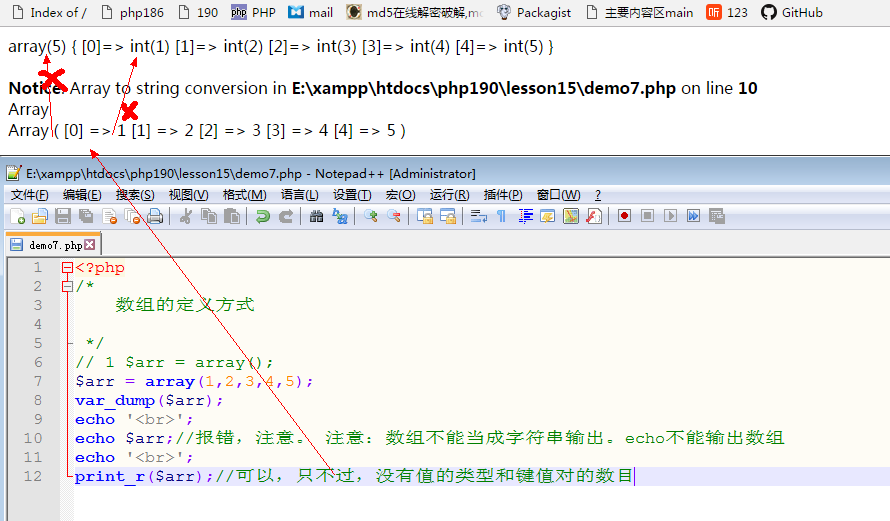


## 特点

### Echo不能输出数组

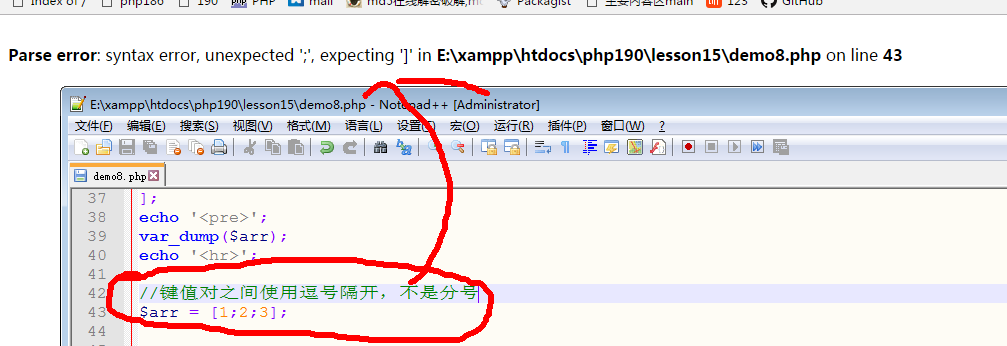


### Print\_r()输出数组

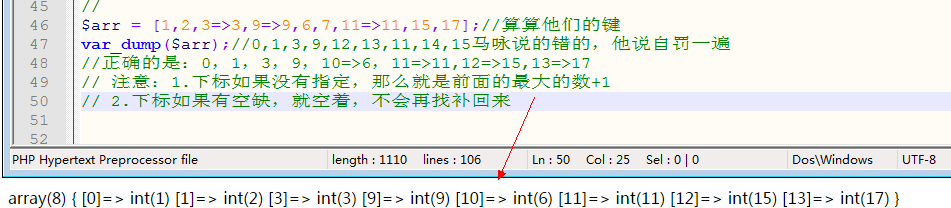


### 键值对

* **键值对之间使用逗号隔开，不是分号**



* 键：下标，key
  + **默认情况下，可以不给键，不给的话，默认从0开始，依次递增加1**
  + **键和值的指向关系是：键=>值**
  + **注意：定义数组的时候键不能重复，如果重复，后面的值生效。**
  + **键只能是标量，布尔，浮点，字符串格式的数字都会转换为整型，键是没有类型的**
  + **如果没有强制指定键，就是前面最大的键+1,中间的空缺不会找补，直接扔了不要**



* 值：值，value

## 数组的分类—人工分的，计算机本身不搭理

### 按照下标的格式：

* 索引数组
  + **下标从0开始，依次递增且连续的数组**



* 关联数组
  + 不是索引数组的，就是关联数组

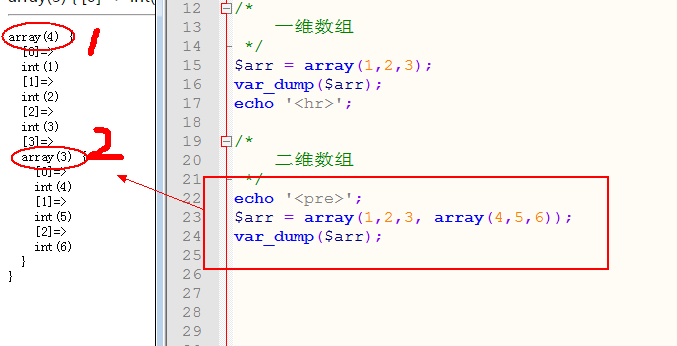


### 按照数组的层级

* 注意：数组的层级不是说array的数目，而是层
* 一维数组



* 二维数组



* 多维数组

