函数：

可变函数：

所谓的可变函数就是通过一个变量来调用函数。

形参的默认值

Php中函数的形参可以赋一个值，这个值就是形参的默认值。

形参的引用

静态变量

函数内部使用static声明的变量就是静态变量。

作用域：

php中有很清晰的作用域。函数外部就是函数外部使用，内部就是函数内部使用。

$GLOBALS超全局变量 数组，保存所有的全局变量。没有作用域的限制，所以可以在内部通过$GLOBALS访问全局变量。

global关键字，用于在函数内部声明变量

语法:

global 变量名

time();

date(‘Y年m月 d日 H时i分s秒’【,time】);

mktime(h,i,s,m,d,y);

strtotime(‘’);

一、字符串

1、定义

使用单引号或双引号括起来的0个或多个字符。

单引号：

不有解析变量的值，能够被转义的 \\ \’

双引号：

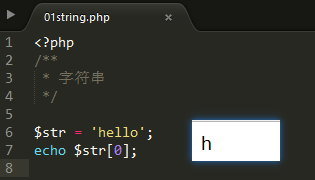
能够解析变量的值，都能被转义

heredoc 本质就是使用双引号定义大段的文本，只是以另一种方式进行书写。

nowdoc 本质就是使用单引号定义大段的文本，只是以另一种方式进行书写。

字符串也可以当作一个由多个字符组织的系列

示例：



2、字符串长度

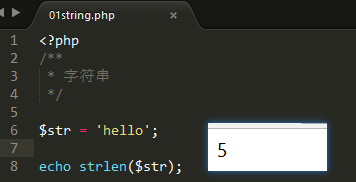
①、strlen

语法：

strlen(变量)

说明：

用于获取字符串的一个字节数

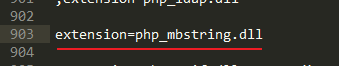


②、多字节字符的支持

默认字母在任何字符住在中占据的1个字节保存一个字符。

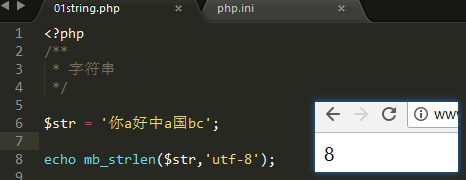
但例如：汉字，一个字符可能占据多个字节，所以php提供了对多字节字符的支持。

需要在php.ini中开启多字节字符的支持。



开启多字节字符的支行后，我们就可以使用多字节字符的操作函数。

mb\_strlen(变量，存储编码)



3、字符串相关函数

①、输出函数

echo

print

print\_r

var\_dump

②、查找并截取函数

strstr(str,substr)

说明：

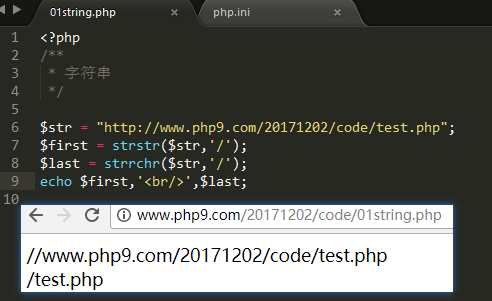
用于在字符串str中查询子字符串substr首次出现的位置，并截取到最后

strrchr(str,substr)

说明：

用于在字符串str中查询子字符串substr最后一次出现的位置，并截取到最后

示例：



③、查找

strpos(str,substr)

说明：

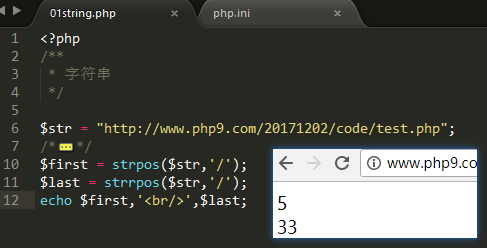
用于在字符串str中查询子字符串substr首次出现的位置

str**r**pos(str,substr)

说明：

用于在字符串str中查询子字符串substr最后次出现的位置

示例：



④、分割

explode(分隔符,str)

说明：

指定指定的分隔符，将字符串str进行分割，并将每一部分组织成数组，并返回

示例：



⑤、替换

str\_replace(search,rep, str);

说明：

在字符串str中，查找search表示的内容，并替换为rep代表的内容



⑥、大小写转换

strtolower()

strtoupper()

示例：



⑦、去除指定字符

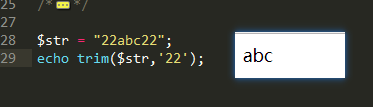
trim(str【,substr】)

说明：

用于将字符串str两侧的子字符串substr去除。

substr可以省略，如果省略表示去除空格。

示例：



ltrim(str【,substr】);

rtrim(str【,substr】)

⑧、pathinfo

语法：

pathinfo(path【,option】);

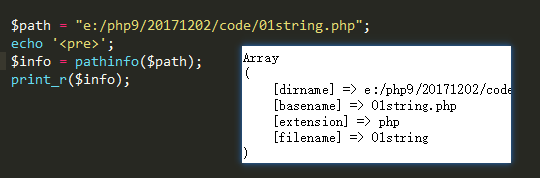
说明：

path 是一个文件路径的字符串

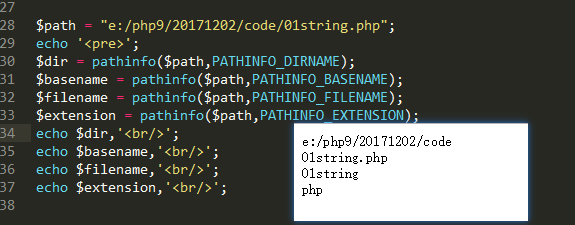
用于获取一个文件的路径信息(文件名、文件夹、文件名、扩展名)

option参数用于获取路径信息中指定的部分

示例：



示例：

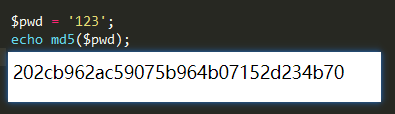


⑨、md5()

语法：

md5(str);

说明：  
 用于对str字符串进持加密。对任何长度的字符串进行md5处理得到的都是32位长度的字符串。



⑩、htmlspecialchars

语法：

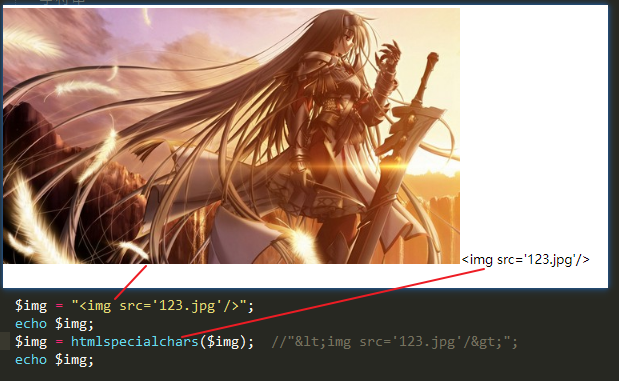
htmlspecialchars(str)

说明：

用于将字符串str中的大于号小于号转换为相应的字符实体。

< &lt; > &gt;

示例：



htmlspecialchars\_decode(str)

二、数组初步

1、概念

数组是一种数据的集合。

数组主要是用于存储具有行列特征(表格)的数据。

2、PHP数组的分类

①、索引数组

数组的下标是整数。这样的数组就是索引数组

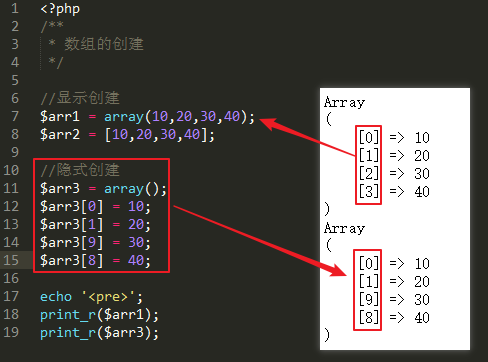
②、关联数组

数组的下标是字符串。这样的数组就是关联数组。

3、PHP数组的创建

①、索引数组的创建

示例：



说明：

在php中数组的下标可以不连续。

②、关联数组的创建

语法：

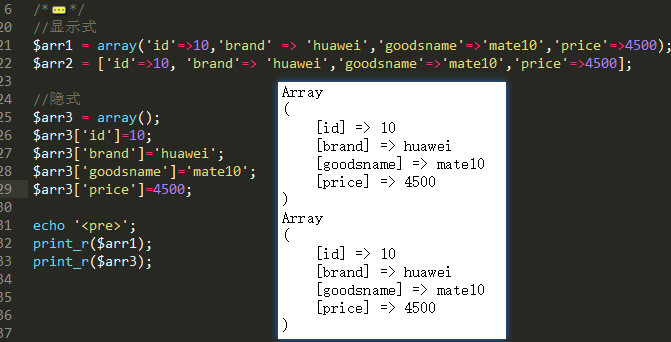
$arr = array(键名=>键值, 键名=>键值,….);

$arr = [键名=>键值, 键名=>键值,….];

说明：

php中的数组元素由两部分组成，键名(下标)，键值

示例：



4、多维数组

php中支持多维数组，如果一个数组的元素又是一个数组，那么这就是多维数组。



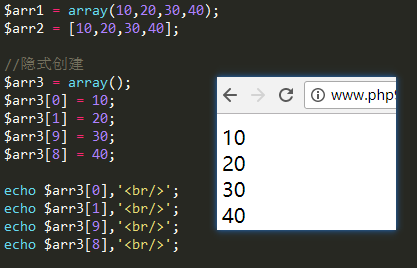
5、数组元素的访问

①、一维数组元素的访问

语法：

$数组名[下标/键名]

**示例：**

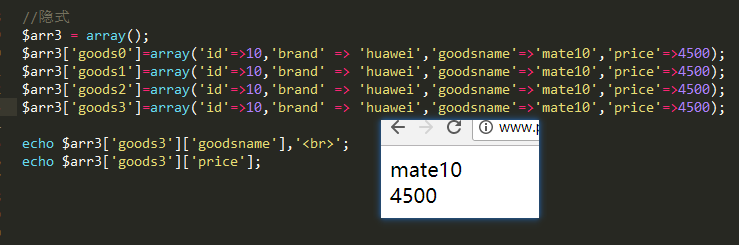


**②、二维数组元素的访问**

**语法：**

**$数组名[行下标][列下标]**

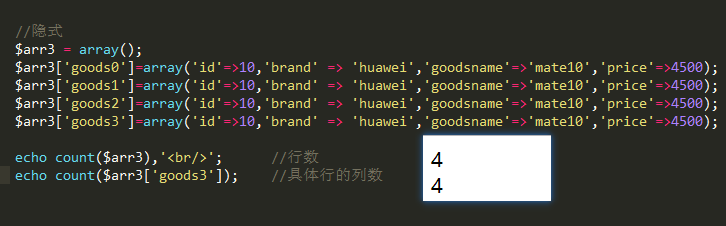
**示例：**



6、数组的长度

count()

用于获取数组的长度



三、数组的指针

数组指针用于表示当前所关注的有元素。

current($arr) 用于当前指针所指向的元素的键值

key($arr) 用于当前指针所指向的元素的键名

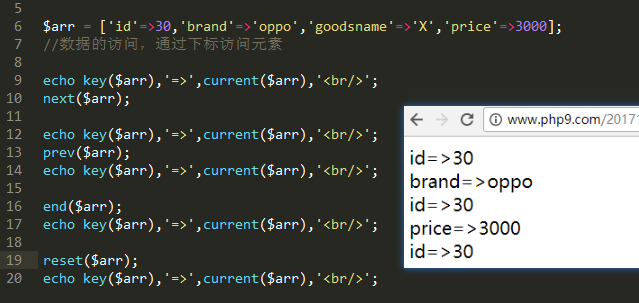
next($arr) 用于将数组的指针下移。

prev($arr) 用于将数组的指针上移。

reset($arr); 用于将数组的指针重置(归位，数组的指针默认位于第1个元素)。

end($arr); 用于将数组的指针移到最后一个元素。

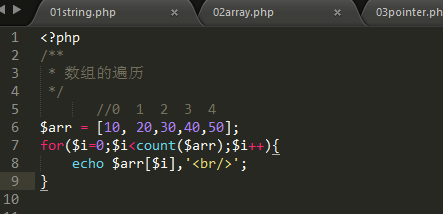
示例：



四、数组的遍历

1、for

for循环是使用循环控制变量来模拟下标的方式来遍历数据，只能遍历下连续或有规则



但不PHP中数组下标可以不连续且不规律，还可能是关联数组。对于关联数组如何来遍历。

2、foreach

①、语法  
 foreach($arr as 【$key=>】$value){

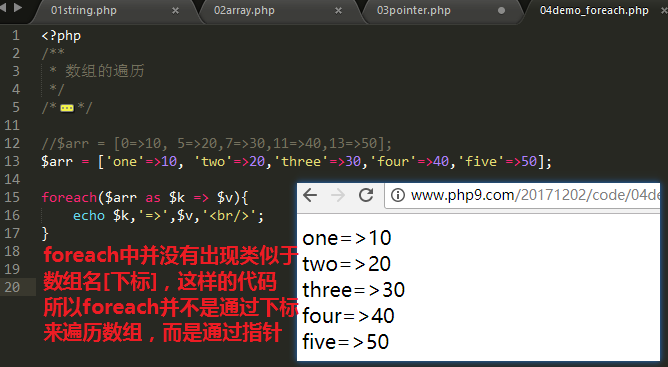
//循环体

**}**

说明：

$arr是所要遍历的数组，$key、$value就是一个变量，变量名可以自定义。

示例：



②、foreah原理

首先对数组的指针进行重置。

读取当前指针所指向的数组元素，并将元素的键名赋值变量$key，将键值赋值给变量$value，

同时会将数组的指针下移一行(下移指针是为了下一次循环)。

foreach不需要我们控制循环的次数，它会自动判断什么时候结束循环。

3、while-each-list遍历(了解)

使用while循环，及each()函数，list语法结构联合来遍历数组。

①、each

each(数组)

说明：

用于获取当前指针所指向的元素键名与键值，并以索引元素与关系元素两种形式表示，同时将指针下移一行。

示例：



②、list

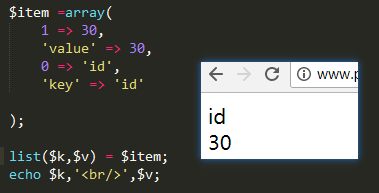
语法：

list(变量列表)=$arr

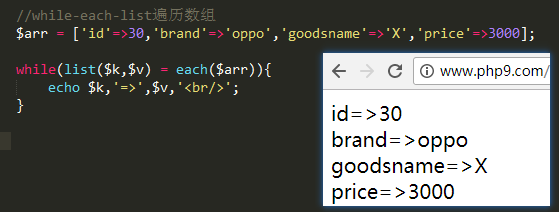
说明：

将数组$arr中的索引元素赋值给变量列表中的变量。

示例：



③、while-each-list遍历数组



③、foreach与list连用

语法：

list(变量列表)=$arr

说明：

list语句的右侧必须是数组

示例：



五、数组操作常用的函数

1、数组的长度

count(数组名)

2、获取数组元素的键名与键值

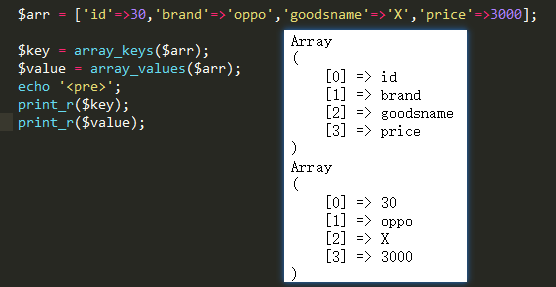
①、array\_keys()

获取数组元素所有的键名

②、array\_values()

获取数组元素所有的键值

示例：



3、判断键名与键值是否存在

①、array\_key\_exists(key,arr)

用于判断某个键名是否存在于数组中，如果存在则返回true，否则 返回false

②、in\_array(value,arr);

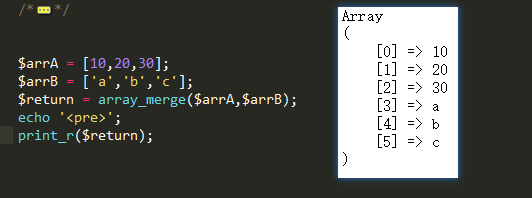
用于判断某个键值是否存在于数组中，如果存在则返回true，否则 返回false

示例：



4、数组的合并

array\_merge(数组1,数组2…)



5、数组的排序

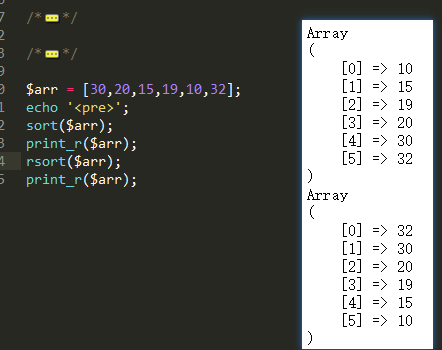
①、sort()

对数组按键值进行升序排序

②、rsort()

对数组按键值进行降序排序

示例：

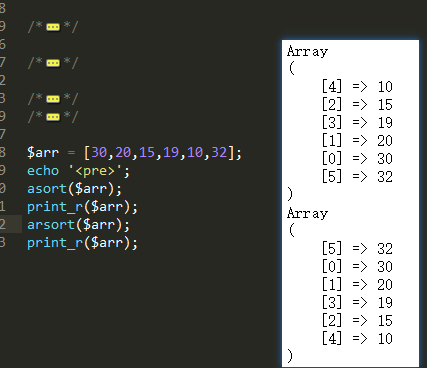


③、asort();

对数组按键值进行升序排序，但原下标不会变

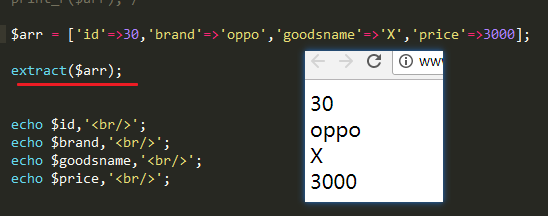
④、arsort()

对数组按键值进行降序排序，但原下标不会变



6、extract

用于解压数组，将关联元素转换为以键名为变量名的变量。

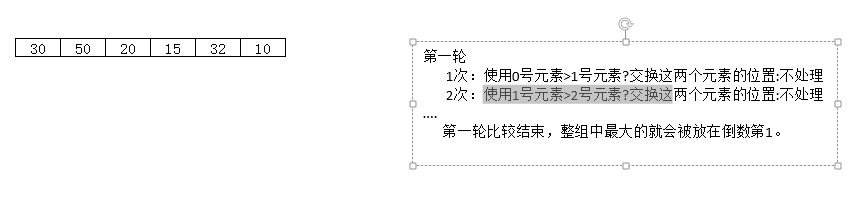


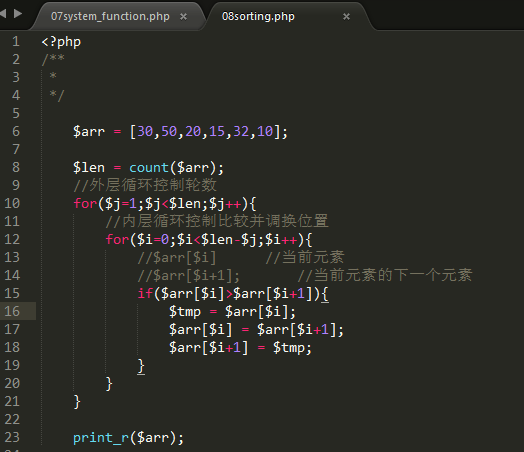
六、数组的相关算法

1、排序算法

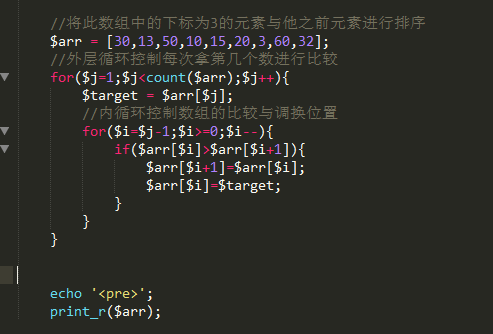
①、冒泡排序法

原理：



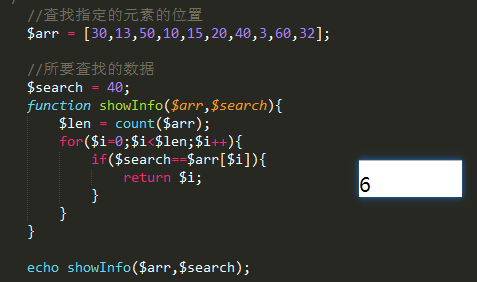


②、插入排序法



2、查找算法

①、顺序查找法



②、二分查找法

前提：

数组一定是有序，且元素不能重复。

