

list() “函数” 的使用：

list()函数用于一次性取得一个数组中从 0 开始的数字下标的多个单元的值！

形式：

list(\$变量 1, \$变量 2, \$变量 3, ..) = \$数组；

作用：

上述一行代码相当于如下代码：

\$变量 1 = \$数组[0];

\$变量 2 = \$数组[1];

\$变量 3 = \$数组[2];

.....

注意：这里变量的个数，要跟该数组的数字下标的单元对应，如果某个变量对应的该数字下标不存在，就会报错！

数组变量的一些细节

- foreach 也是正常的循环语法结构，可以有 break 和 continue 等操作。
- 遍历过程中**值变量**默认的传值方式是值传递。
- 遍历过程中**值变量可以人为设定为引用传递**：foreach(\$arr as \$key => &\$value){ ... }
- foreach 默认是原数组上进行遍历。但如果在遍历过程中对数组进行了某种修改或某种指针性操作，则会复制数组后在复制的数组上继续遍历循环（原数组保持不变）。
- foreach 中如果值变量是引用传递，则无论如何都是在原数组上进行。

数组排序

冒泡排序算法

演示数组为：

\$a = array(9, 3, 5, 8, 2, 7); //下标为 0,1,2,3,4,5

演算过程描述：

从数组的左边开始，依次两两比较**相邻的 2 个数据**的大小，如果发现左边的比右边的大，则将他们进行交换。这样进行“一趟”之后，必然可以确定最大的一个数据放在最右边。

按此方式，对“剩余的数据”继续进行下一趟，则有会确定这些剩余数据耳朵最大值放在剩余位置的最右边。

演示：

原始数组：	9	3	5	8	2	7
第 1 趟后：	3	5	8	2	7	9
第 2 趟后：	3	5	2	7	8	9
第 3 趟后：	3	2	5	7	8	9
第 4 趟后：	2	3	5	7	8	9
第 5 趟后：	2	3	5	7	8	9

规律描述：

- 1，假设数组的数据有 n 个。
- 2，要进行比较的“趟数”为 $n - 1$ ；
- 3，每一趟要比较的数据个数都比前一趟少一个，第一趟要比较 n 个（即比较 $n - 1$ 次）
- 4，每一次比较，如果发现“左边数据”大于“右边数据”，就对这两者进行交换位置。

选择排序：

演示数组为：

`$a = array(9, 3, 5, 8, 2, 7);` //下标为 0,1,2,3,4,5

演算过程描述：

求得一个数组的最大值的下标，并将这个最大值下标的单元跟最后一个单元进行交换；

然后，继续从剩余数据中取得最大值的下标，并将这个最大值下标的单元跟剩余的最后一个单元交换以此类推，直到只剩下一个数据，就不用找了。

演示：

原始数组：	9	3	5	8	2	7
第 1 趟后：	7	3	5	8	2	9
第 2 趟后：	7	3	5	2	8	9
第 3 趟后：	2	3	5	7	8	9
第 4 趟后：	2	3	5	7	8	9
第 5 趟后：	2	3	5	7	8	9

规律描述：

- 1，假设数组的数据有 n 个。
- 2，要进行查找最大值单元并进行交换的“趟数”为 $n - 1$ ；
- 3，每一趟都要求出“剩余数据”中的最大值单元，并且，剩余数据的数量每一趟都少 1 个，第一趟有

n 个。

4, 每一趟找出最大值单元后, 都要进行交换: 最大值单元, 跟剩余数据中的最后一个单元交换。