

数组函数。

array_keys();

- 语法：
 - array_keys(array arr);
- 功能：
 - 以数组的形式返回arr数组中的键名。
- 应用场景：Model类的insert方法中获取字段名。

```
$data = [  
    ['name' => '小红', 'gender' => 0],  
    ['name' => '小明', 'gender' => 1]  
]  
$re = array_keys($data);  
var_dump($data);  
<!--  
    输出:  
    array(2) {  
        [0]=>  
        string(4) "name"  
        [1]=>  
        string(6) "gender"  
    }  
-->  
$sql = $this->buildInsert(array_keys($data[0]), count($data));
```

array_values();

- 语法：
 - array_values(array arr);
- 功能：
 - 以数组的形式返回arr数组中所有的元素值。（过滤掉重复的元素值）

array_fill();

- 语法：
 - array_fill(int start_key,int length,value);
- 功能：
 - 创建一个数组，初键为start_key，长度为length，每个元素的值都为value。
- 应用场景：
 - Model类的buildInsert方法中生成(?,...,?)。

```
$values = array_fill(0, count($fields), '?');
```

array_reduce();

- 语法：
 - array_reduce(array \$array, callable \$callback, mixed \$initial = null): mixed
- 功能：
 - 将回调函数 callback 迭代地作用到 array 数组中的每一个单元中，从而将数组简化为单一的值。
 - 关键字：迭代、简化。
- 实现：

```

function arrayReduce($arr,$callBack,$init){
    foreach($arr as $k => $v){
        // carry携带上次迭代的返回值; 如果本次迭代是第一次, 那么这个值是 initial.
        if($k == 0){
            $carry = $init;
        }
        $carry = $callBack($carry,$v);
    }
    return $carry;
}

function merge($carry,$item){
    // $carry = array_merge(array_values($item),$carry);
    $carry = array_merge($carry,array_values($item));
    /*
        注意这里的顺序。
        如果carry在前, item在后则是顺序。
        (item附加到carry上)
        如果item在前, carry在后则是倒序。
        (carry附加到item上)
    */
    return $carry;
}

// 测试
$arr = [
    [
        'name'=>'ccc',
        'gender'=>'1',
        'email'=>'ccc@myframe.test',
        'mobile'=>'15828338094'
    ],
    [
        'name'=>'abc',
        'gender'=>'1',
        'email'=>'abc@myframe.test',
        'mobile'=>'15828837894'
    ]
];

$re = arrayReduce($arr,'merge',[]);
var_dump($re);
/*
输出
array(8) {
    [0]=>
    string(3) "ccc"
    [1]=>
    string(1) "1"
    [2]=>
    string(16) "ccc@myframe.test"
    [3]=>
    string(11) "15828338094"
    [4]=>

```

```
string(3) "abc"  
[5]=>  
string(1) "1"  
[6]=>  
string(16) "abc@myframe.test"  
[7]=>  
string(11) "15828837894"  
}  
*/
```

array_map();

- 语法：
 - array_map(callable \$callback, array \$array, array ...\$arrays): array
- 功能：
 - 为数组的每个元素应用回调函数。
- 返回值：
 - 返回数组，是为 array 每个元素应用 callback函数之后的数组。
- 实现：

```

function cube($n)
{
    return ($n + $n + $n);
}

function arrayMap($callback,array $arr){
    $a = [];
    foreach($arr as $v){
        $r = $callback($v);
        $a[] = $r;
    }
    return $a;
}
// 测试
$arr = [1, 2, 3, 4, 5];
$re = arrayMap('cube',$arr);
// 1*3,2*3,3,4*3,5*3
var_dump($re);
/*
输出:
(
    [0] => 3
    [1] => 6
    [2] => 9
    [3] => 12
    [4] => 15
)
*/

```

- 应用场景：
 - Model类的update方法中，让每个需要更新的字段连上'=?',即\$f1=?, \$f2=?...;

```

$fields = implode(',',array_map(function($v){
    return "`$v`=?" ;
},$fields));

```

array_merge();

- 语法：
 - array_merge(array1,array2,array3...)
- 功能：
 - 把一个或多个数组合并为一个数组。
- 注意：
 - 两数键相同。
 - 若键字符串，后值覆盖前。
 - 若键为数字，后值附加前。

- 应用场景：array_reduce()函数的回调函数内部使用了这个函数，目的是把多维数组变成一维数组。

```
$data = array_reduce($data, function ($carry, $item) {  
    return array_merge($carry, array_values($item));  
}, []);
```