**PHP数组处理函数**

**第30课：**

implode(); 把数组合成字符串

explode(); 把字符串分割成数组

in\_array(); 检测内容是否在数组中

each(); 把数组元素拆分成新的数组

list(); 把数组元素赋值给变量

**第31课：**

count(); 统计数组中元素的个数

reset(); 把数组内部指针移动到数组第一个元素，并返回元素值

end(); 把数组内部指针移动到数组最后一个元素，并返回元素值

prev(); 把数组指针指向上移动一位，并返回当前指针，如果到结尾返回 FALSE，如果当前元素的值为空，或者0，返回值

next(); 把数组指针指向下移动一位，并返回当前指针，如果到结尾返回 FALSE，如果当前元素的值为空，或者0，返回值

current(); 返回当前指针指向的数组元素，如果当前元素为空返回false

key(); 返回当前指针指向元素的键名

**第32课：**

in\_array(); 判断内容是否在数组中

array\_search(); 查找内容是为数组的键值，存在返回键名，不存在返回BOOL假

array\_change\_key\_case(); 更改数组键名大小写

array\_chunk(); 把数组进行拆分

array\_combine(); 生成新数组

array\_diff(); 返回在其他数组中不存在的键值

array\_diff\_key(); 返回在其他数组中不存在的键名

array\_diff\_assoc(); 返回在其他数组中不存在的键名+键值

**第33课：**

array\_diff\_ukey(); 通过回调函数的方式，返回一个数组在其他数组中不存键名的数 组元素

array\_diff\_uassoc(); 通过回调函数的方式，返回一个数组在其他数组中不存键名和键 值的数组元素

array\_intersect(); 在两个或多个数组当中，返回键值都存在的，数组元素（第一个 数组）

array\_intersect\_key(); 在两个或多个数组当中，返回键名相同的数组元素（第一个数组）

array\_intersect\_assoc(); 在两个或多个数组当中，返回键名和键值都相同的数组元素（第 一个数组）

array\_intersect\_ukey(); 通过回调函数的方式，返回第一个数组的键名在其它数组当中同 时存在的数组元素

array\_intersect\_uassoc(); 通过回调函数的方式，返回第一个数组的键名和键值在其它数组 当中同时存在的数组元素

**第34课：**

array\_fill(); 通过指定的索引顺序及个数生成数组

array\_filter(); 数组过滤函数，通过回调函数的方式返回新数组，如果回调函数 返回TRUE，数组元素返回到新数组当中

array\_flip(); 把数组中的键名和键值进行交换

array\_key\_exists(); 判断内容是否是数组的键名

array\_keys(); 返回数组中所有的键名

**第35课：**

array\_udiff(); 通过回调函数的方式，比较两个或多个数组的差值，只比较值， 不进行键名的比较

array\_udiff\_assoc(); 通过回调函数的方式，比较两个或多个数组的键名和键值的差值

array\_udiff\_uassoc(); 通过回调函数的方式，比较多个数组的键名和键值,要定义两个 回调函数来处理键名和键值的比较

array\_uintersect(); 通过回调函数的方式，得到多个数组中键值的交集，键名不会参 与比较

array\_uintersect\_assoc(); 通过回调函数的方式，得到多个数组中键名和键值的交集

array\_uintersect\_uassoc(); 通过回调函数的方式，得到多个数组中键名和键值的交集,要定 义两个回调函数来处理键名和键值的比较

**第36课：**

array\_unique(); 清除数组中的重复项

array\_unshift(); 向数组的顶部最加数据元素

array\_shift(); 删除数组第一个元素，如果删除失败返回null

array\_pop(); 删除数组最后一个元素，如果删除失败返回null

array\_keys(); 得到数组的键名，以数组的凡是返回

array\_values(); 得到数组的键值，以数组的凡是返回

**第37课：**

each(); 返回数组中的键名和键值生成为新数组，如果指针结束，返回布 尔假FALSE

list(); 将索引数组的值赋值给变量

array\_map(); 数组的每一个元素，都经过回调函数处理，返回值是处理过的元 素组成的新素组

array\_walk(); 数组中的吗，每一个元素，应用回调函数，成功返回真TRUE 失 败返回假

array\_walk\_recursive(); 递归的操作数组，使用数组中的元素应用回调函数

**第38课：**

compact(); 将多个变量转化为数组，变量名为数组键名 变量值为数组键值

extract(); 将数组转化为变量 键名作为变量名 键值为变量值

in\_array(); 判断元素是否存在于数组中，如果加参数TRUE，同时进行数据 类型的判断，返回BOOL类型值

**第39课：**

array\_merge(); 将多个数组合并，生产新数组

array\_merge\_recursive(); 将多个数组进行合并，如果xi键名相同，生产二维数组

array\_change\_key\_case();

**第40课：**

range(); 生成包含指定范围的数组，第三个参数是步长值

array\_count\_values(); 统计数组中值出现的次数，返回统计数组

array\_pad(); 对数组进行填充，第二个参数如果是负数，从左侧添加，整数为 右侧，第三个参数为填充内容

array\_product(); 对数组中的键值进行乘积的运算，返加数值类型

**第41课：**

array\_push(); 在数组的末端增加一个或多个个元素，入栈

array\_pop(); 删除数组末端元素，出栈

array\_rand(); 随机取得数组元素，如果是取得多个元素返回元素索引的数组， 单个只返回这个元素的索引

array\_reverse(); 反转函数，如果第二个参数指定为TRUE，保留元素的键名

**第42课：**

array\_reduce(); 用回调函数递归的的对数组元素进行处理，返回处理后的值

array\_slice(); 截取指定范围的数组元素，第二个参数截取开始位置，第三个参 数截取长度，第四个参数是保留数字索引值

array\_splice(); 删除指定范围的素组元素，如果指定第四个参数，则在删除元素 的位置放入第四个参数值

array\_sum(); 对数组中的值进行求和运算

array\_product(); 对数组中的值进行乘积运算

**第43课：**

sort(); 对数组元素进行递增的排序

natsort(); 通过自然顺序对数组进行排序

shuffle(); 随机对数组进行排序

natcasesort(); 不区分字符大小写进行排序

array\_multisort(); 对多个数组或多维数组 进行排序

**第44课：**

usort(); 通过自定义函数对数组进行排序，原数组索引不保留

uasort(); 通过自定义函数对数组进行排序，保留原数组组索引

uksort(); 通过调用自定义函数对数组的键名进行排序

**第45课：**

asort(); 对数组按值排序，保留键名

ksort(); 对数组按键名排序，保留键名

rsort(); 对数组进行反向排序，不保留键名

arsort(); 对数组进行反向排序，保留键名

krsort(); 对数组的键名进行反向排序，保留键名