此文是根据php官方文档与php程序设计一书完成的。

## php简介

php(HyperText preprocessor,超文本预处理器)，是一种被广泛应用的开源代码的多用途脚本语言，可以嵌入到html中，尤其适合web开发。

与c写的脚本的区别—与用大量的命令来编写程序以输出html不同的是，php页面就是html.只不过在其中嵌入了一些代码来做一些事情

与客户端的javascript不同的是，php代码运行在服务端。

## php能做什么

### 服务端脚本

创建动态的web内容

### 命令行脚本

Php通过使用命令行脚本来执行系统管理任务，例如备份和日志解析

### 客户端GUI应用

Php可以通过PHP-GTK来编写客户端应用程序，php-GTK是php的一个扩展

## 基本语法

### Php标记

<? 和?>,解析php文件，php会寻找起始标记和结束标记，这种解析方式使得php可以嵌入到各种不同的文档中去，而任何起始和结束标记之外的部分都会被php解析器忽略

### 指令分割符

Php每个语句后用分号结束指令

### 注释

//

/\* \*/

## 词法结构

### 大小写

和内置结构的关键字(echo,class,while)一样,用户定义的类名和函数名也是不区分大小写的，

但是变量区分大小写

### 空白符与换行符

Php中空白符无关紧要，一条语句可以跨多行，也可以紧密的连在一起

### 标识符

标识符只是一个名字，在php里标识符用来命名变量、函数、常量和类

关键字

是语言为其核心功能而预留的单词，在对变量、函数、类或常量命名时不能使用和关键字相同的名字。

## 数据类型

## 变量

## 常量

## 表达式

## 运算符

## 函数

## 字符串

## 数组

### 索引数组与关联数组

索引数组的键名是整数并从零开始

关联数组的键名时字符串

### 识别数组元素

$arr[‘Fred’]

$arr[3]

键可以是一个字符串或一个整数，等价于证数但不以0开头的字符串被当成整数看待，

因此，$arr[3]和$array[‘3’]是一样的，$arr[‘03’]不一样

负数是有效的键

不一定需要将单个字符串加上引号，$arr[‘Fred’]和$arr[Fred]是一样的，但一般不这么做，

因为没有引号的键不能和常量区分开

Define(‘index’,5);

Echo $arr[index];----输出的是$arr[5]

### 数组初始化

### 直接赋值

$arr[0]=1;

$arr[1]=2;

$arr[2]=3;

$test[‘color’]=’red’;

$test[‘size’]=13;

$tes[‘kind’]=’my’;

### Array初始化

$arr=array(1,2,3,4);

$test=array(‘color’=>’red’,’size’=>13,’kind’=>’my’)

可以用=》来指定一个初始索引，并在后面指定一些列键值

$days=array(1=>’mondy’,’tuesday’,’wednesday’)

数组的2号元素是Tuesday，3号元素是Wednesday

如果初始索引是一个非数字的字符串，那么后面的索引是从零开始的

$day=array(‘Friday’=>’black’,’brown’,’green’)

等价于

$day=array(‘Friday’=>’black’,0=>’brown’,1=>’green’)

### 在数组末尾添加值

使用[]语法,前提数该数组的索引是数字并且索引从零开始

$arr=array(‘color’,’size’,’kind’);

$arr[]=’name’;

所以$arr[3]=name

如果是关联数组会给新的元素指定数组

$arr[‘test’=>’color];

$arr[]=’size’;

结果是

$person[0]=size;

### 指定值的范围

range()函数创建一个连续的整数或字符数组，整数或字符的值在传递给函数的两个参数之间

$number=range(1,5); //等价于 $number=array(1,2,3,4,5);

$letter=range(‘a’,’f’); //保存字母到数字 $letter=range(‘a’,b’,c’,’d’,’e’,’f’);

$reversed=range(5,1);// $reversed=array(5,4,3,2,1);

字符串参数中只有第一个字母被用于限定范围

range(‘aaa’,’’zzz’);//等价于 range(‘a’,’z’);

### 获取数组大小

count()和sizeof()函数在用法和效果上是一样的

### 填充数组

array\_pad()函数用重复的内容来创建数组

array\_pad(第一个参数是函数，第二个参数是你想要拥有元素个数，第三个元素是要给每个元素的初始值)

$arr=array(1,2,3)

$pad=array\_pad($arr,5,0); //填充后的数组 array(1,2,3,0,0)

将新值填充到数组开头

$pad=array\_pad($arr,-5,0);//填充后的数组 array(0,0,1,2,3)

将array\_pad()函数的处理结果赋值给原来的数组，可达到改变原来位置的效果

$arr=array\_pad($arr,5,0);

如果你填充元素到一个关联数组，已经存在的键将被保留，新的元素将使用从0开始的数字键

### 多维数组

数组的值也可以是数组

$row\_0=array(1,2);

$row\_1=array(3,4);

$row\_2=array(5,6);

$test=array($row\_0,$row\_1,$row\_2);

要使用变量替换来访问多维数组的 中的元素，必须将整个元素用大括号括起来

Echo “this value is {$test[1][2]}”

不适用大括号结果如下

This value is array[2]

### 析取多个值

要把一个数组的所有值都赋值到变量中，可以使用list()结构

$person(‘lili’,23,’男’)；

list($name,$age,$gender)=$person;//$name=’lili’;$age=23 $gender=’男’

如果数组中的元素个数比list()中的变量个数多，则多余的数被忽略

list($name,$age,$gender)=$person;//$name=’lili’;$age=23

如果list()中的元素个数比数组中的多，则多余的值被设为NULL

$person(‘lili’,23’)；

list($name,$age,$gender)=$person;//$name=’lili’;$age=23 $gender=NULL

在list()中用两个或多个连续的逗号跳过数组中的值

$value=range(‘a’,’e’);

List($m,,$n,,$o)=$value; //$m=’a’,$n=’c’,$o=’e’;

### 切割数组

array\_slice()用来析取数组的一个子集

$subject=array\_slice(array,offset,length)

Offset指定要赋值元素的初始元素。Length指定要复制的值的个数

$people=array(1,2,3,4,5);

$middle=array\_slice($people,2,2) // 结果为 $middle=array(3,4)

只有在索引数组上使用array\_slice()才有效，

$person=array(‘name’=>’lili’,’age’=>35,’wife’=>’betty’);

$test=array\_slice($person,1,2);//结果为 $test=array(0=>35,1=>’betty’);

结合list()和array\_slice()使用

$order=array(1,2,3,4,5,6);

List($test1,$test2)=$array\_slice($order,1,2);

//结果为$test1=2;$test2=3;

### 数组分组

array\_chunk()函数把数组划分成小数组或固定大小的数组

$chunk=array\_chunk(array,size,preserve\_keys)

Size为划分数组的长度

Preserve\_keys是一个布尔值，它指定新数组的元素是否有和原数组相同的键，还是从0开始的新数字键,默认是分配新键值的

True 保留原始数组的键名

False 默认，每个新数组块使用从0开始

$num=range(1,7)

$rows=array\_chunk($num,3)

打印结果：

