## 变量

### 1.1变量是用于临时存储数据的容器，php中用$来声明一个变量。

Php是弱类型语言，不需要在使用变量前声明变量类型。Php根据变量的值自动把变量转换成正确的数据类型。

### isset()与empty(),unset()

unset()用于销毁变量---unset($var)

isset()---函数参数存在则返回true否则返回false

empty()---参数为非零非空则返回flase,参数为空则返true,

var\_dump(isset($a)); // TRUE var\_dump(isset($a, $b)); // TRUE

Isset判断变量是否存在，可以传入多个变量，若其中一个变量不存在则返回假，empty判断变量是否为空，只可传一个变量，如果为空为假则返回真。

### 1.3可变变量

$hi=”hello”;

$$hi=”world”;

Echo “$hi $hello”🡺输出 hello world ----$$hi=$hello

### 1.4变量的引用赋值

$a=”test”;

$b=&$a;

改变其中任何一个另一个都会受到影响

PHP使用&进行引用赋值。

$a=”test”;

$b=&$a;

$b=”test is test”;<==>$a=”test is test ”  
echo $a;---输出test is test

Echo $b;--输出test is test

不能将表达式和函数引用赋值

$c=&(23+1);

$c=&tets();

### 1.5强制类型转换

三种方法：

A:(int)$a,(string)$b---括号加类型强制转化

B:intval($a);floatval($a);strval($a);---使用转换函数

C:settype($a,”string”);

A,B并没有改变变量本身类型，而是通过转换将得到的新数据类型赋给新的变量，而C会改变将本身的变量改变成其他的类型。

### 1.6变量类型测试函数

is\_booi();is\_int();is\_object()…..is\_string();is\_array()

### 1.7输出

A:echo可以输出字符串，变量、表达式等单项的值，不能输出数组、对象等复杂数据类型和数据类型，没有返回值

B：和echo一样可以输出单一的数据，但是又返回值true/false

C:print\_r(),既能输出字符串等单一数据又能输出数组等多种数据 但是不能输出数据类型. 与var\_dump的区别就是只打印数组里面元素的值，不打印其他的额外的信息！

D: 不仅仅可以输出一个或多个数据项的值，还可以打印出数据项的类型、长度等额外的信息！而且可以输出复杂数据类型！

### 1.8预定义变量

预定义变量是PHP预设的一组数组，其数据包括运行环境、用户表单输入的数据等。

如：服务器变量$\_SERVER；环境变量$\_ENV；超全局变量$GLOBALS。

$\_SERVER[‘SERVER\_ADDR’]------服务器所在IP地址

$\_SERVER[‘HTTP\_USER\_AGENT’]------当前用户使用的浏览器信息

$\_SERVER[‘SERVER\_PORT’]-----服务器所使用的端口

$\_SERVER[‘PHP\_SELF’]------当前正在执行的脚本文件名

$\_SERVER[‘SERVER\_SIGNTURE’]-----当前服务器版本和虚拟主机名的字符串

$\_POST 获取post方法提交的数据

$\_GET 获取get方法提交的数据

### 1.9销毁变量

通过unset()来销毁变量。unset()销毁的是变量名，值是由PHP垃圾回收机制销毁。

<?php

$a=10;

unset ($a); //销毁变量名

echo $a; //无输出，已经销毁

?>

<?php

$a=10;

$b=&$a; // $b引用$a的地址

unset( $a); // 销毁变量

echo $b; //结果还是10（值还在，仅销毁了变量名）

?>

### 1.1.0常量

在PHP中，常量是一个简单值的标志符，在脚本执行间值是不变的。

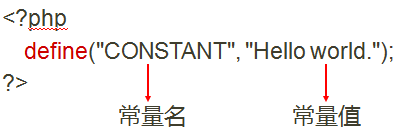
常量区分大小写，常量标志符一般都是**大写**。

常量名和其它任何PHP标记都遵循相同的命名规则。

自定义常量使用**define()**函数来定义（常量不能重复定义）

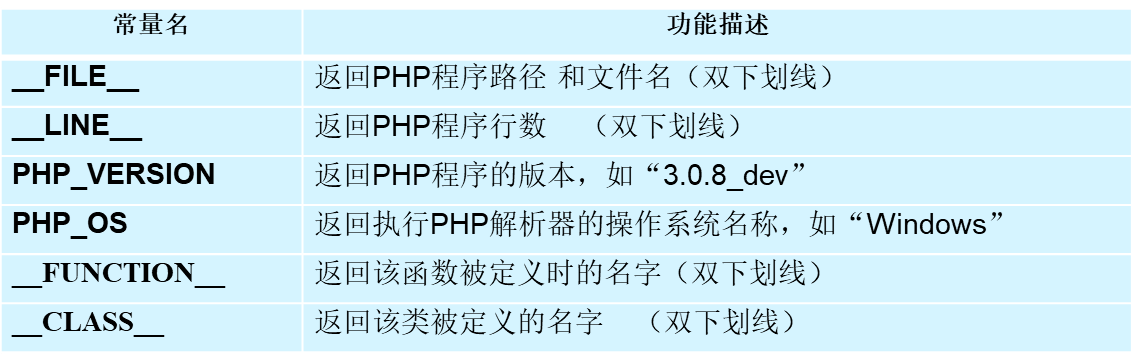
define(“常量名” ，“常量值” ，是否区分大小写）；

true 不区分； false 区分（默认是区分的）。



### 1.1.1预定义常量

PHP中还提供了很多预定义常量。这些常量专门用于获取PHP中的信息，并且是不允许开发人员随意修改的



### 1.1.2单引号双引号区别

双引号解析变量和特殊字符(\” \’ \\ \$ \n \r)，单引号不能

单引号能准确打印 输入的内容，但除了转义字符的单引号（\’)，

反斜杠（\\）外。

由于双引号的内容会被解释，所以双引号的效率会比单引号的效率低

单引号内部的变量不会执行， 双引号会执行

单引号解析速度比双引号快。

单引号只能解析部分特殊字符，双引号可以解析所有特殊字符。

### 1.1.3 heredoc结构定义字符串

在需要大量输出HTML文本时可采用heredoc的定界符“<<<” 格式：

其中STR为标志符，最后用同样的标志符结束字符串(前面不能有空格）。

echo <<< STR

要输出的字符串

STR;

### 1.1.4数据类型的自动转换

PHP中自动 转换通常发生在不同数据类型的变量进行混合运算时。若参与运算的类型不同，则先转换成同一类型，然后才运算。通常4种标量类型才使用自动类型转换。（integer 、float 、string、 boolean）

有布尔型时，true转化为整型1，false转化为整型0后再参与运算。

有null值参与运算时，null值转化为整型0再进行运算。

有integer型和float型参与运算时，先把整型转为浮点型再进行运算。

有字符串和数字型参与运算时，字符串先转换为数字再参与运算。

如：字符串“123abc”转为整数123；

字符串“abc”转为整数0；

字符串“12.45abc”转为浮点数12.45。

## 字符串处理函数

获取字符串长度 函数strlen()

字符串比较函数 strcmp() 和 strcasecmp()

查找和替换函数 substr()

strrpos() 函数查找在字符串中最后一次出现的位置

字符串裁剪函数 **trim()**、**rtrim()**和**ltrim()**

### 2.1并置运算符

并置运算符 (.) 用于把两个字符串值连接起来。

$txt1="Hello world!";   
$txt2="What a nice day!";   
echo $txt1 . " " . $txt2;

### 2.2strlen() 函数

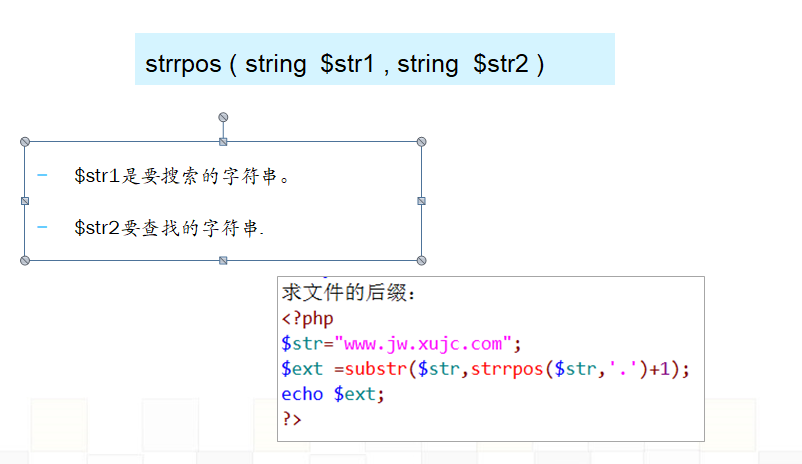
返回字符串的长度

strlen(“hello world!”);返回12

### 2.3Strops()函数

文件后缀名一般1~4个，所以如果要是想获取文件扩展名，还可用**strrpos() 函数**

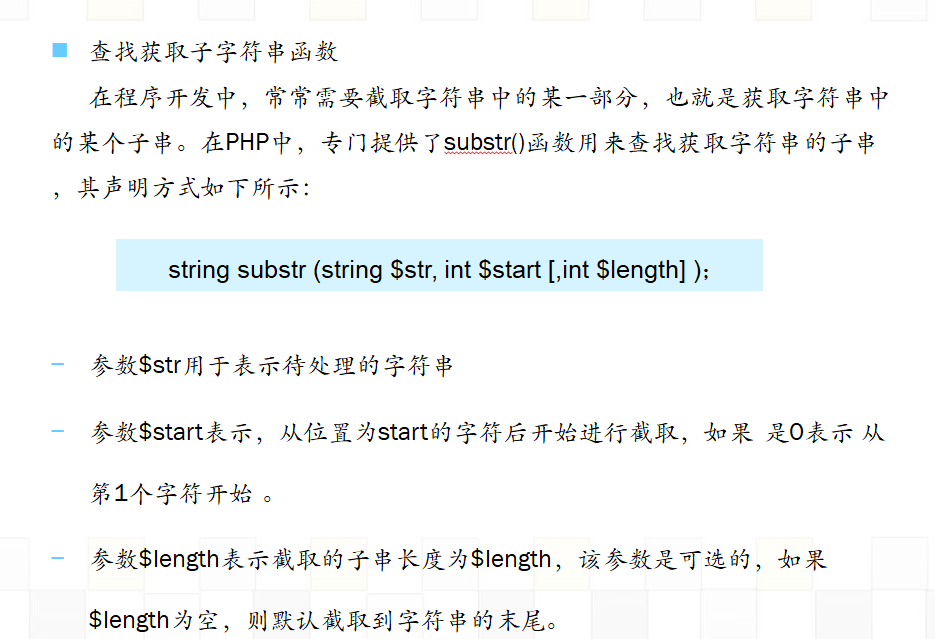
查找字符串在另一字符串中最后一次出现的位置



查找一个字符或一个文本，返回第一个匹配的字符位置

strops("Hello world!","world");返回6

### 2.4substr()函数





substr(str,start,length)

start>0,length没有，从左往右，从指定的start位置到字符串结尾

echo substr("Hello world",10)."<br>";===》d

start <0,length没有,从左往右，从指定的start位置到字符串结尾

echo substr("Hello world",-1)."<br>";===》d

指定length

Start>0 ; 返回start-length 不包括length

echo substr("Hello world",0,10)."<br>"; Hello worl

echo substr("Hello world",0,-1)."<br>" Hello worl

Start<0;返回start-length 不包括length

echo substr("Hello world",-10,-2)."<br>"; ello wor

echo substr("Hello world",-10,4); ell

### 2.5array\_count\_values() 函数

用于统计数组中所有值出现的次数

### 2.6 strcmp 进行字符串二进制比较

0 - 如果两个字符串相等

<0 - 如果 string1 小于 string2

>0 - 如果 string1 大于 string2

### 2.7define()定义一个常量

在设定以后，常量的值无法更改

常量名不需要开头的美元符号 ($)

作用域不影响对常量的访问

常量值只能是字符串或数字

### 2.8 {$a}和$a区别

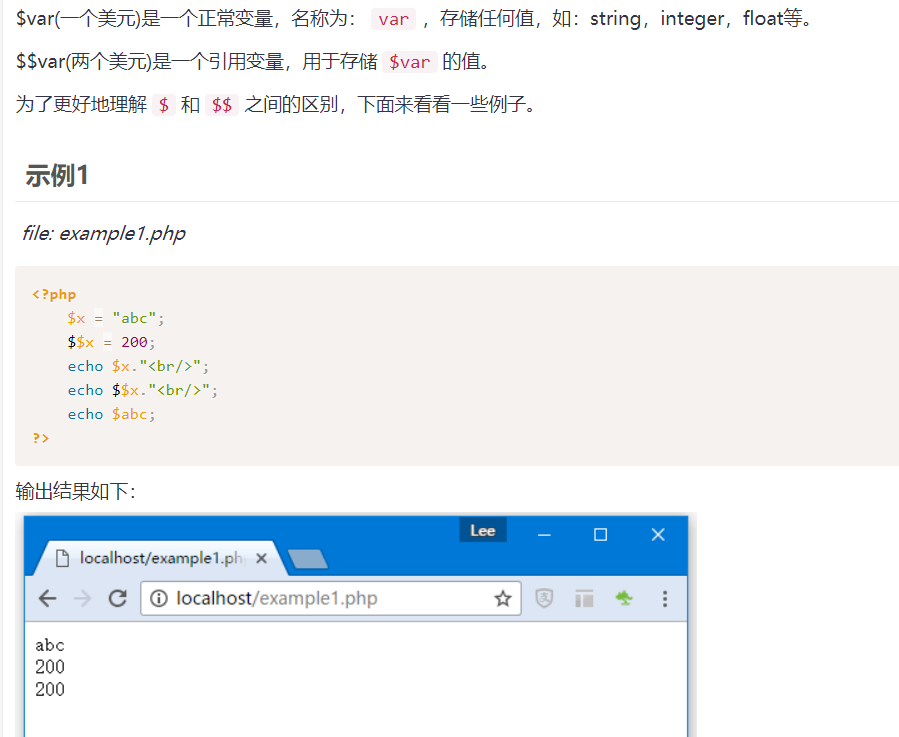
$Try[2]['key']="Jim";

echo "hello,$Try[2]['key']"; //hello,Array['key']

echo "</br>";

echo "hello,{$Try[2]['key']}";//输出的是hello，jim

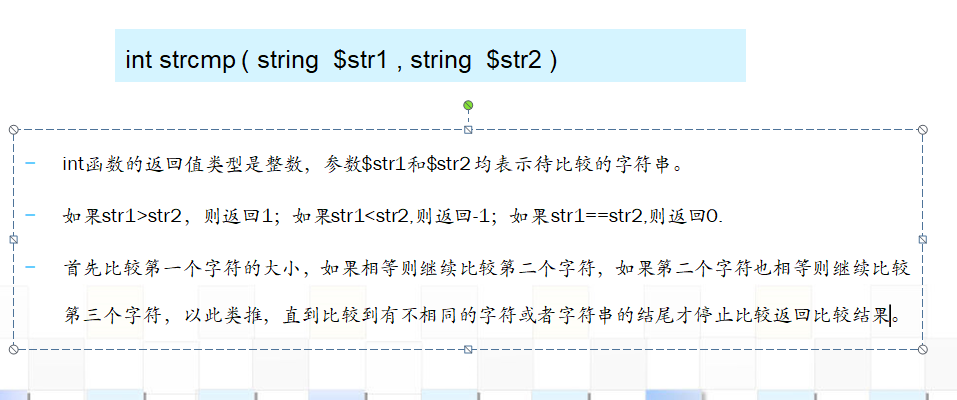
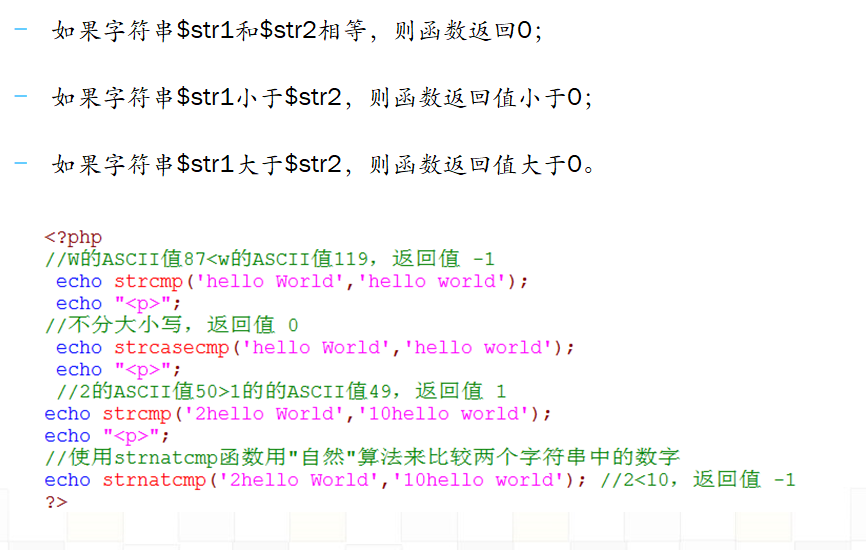
### 2.9 $和$$之间的区别



### 2.1.1字符串比较函数 strcmp() 和 strcasecmp()

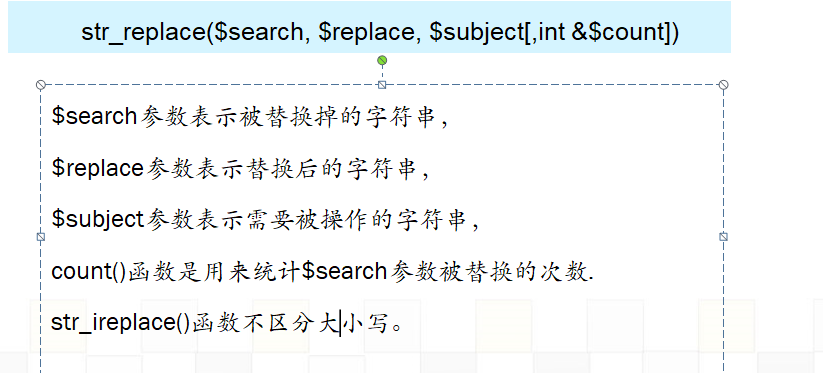
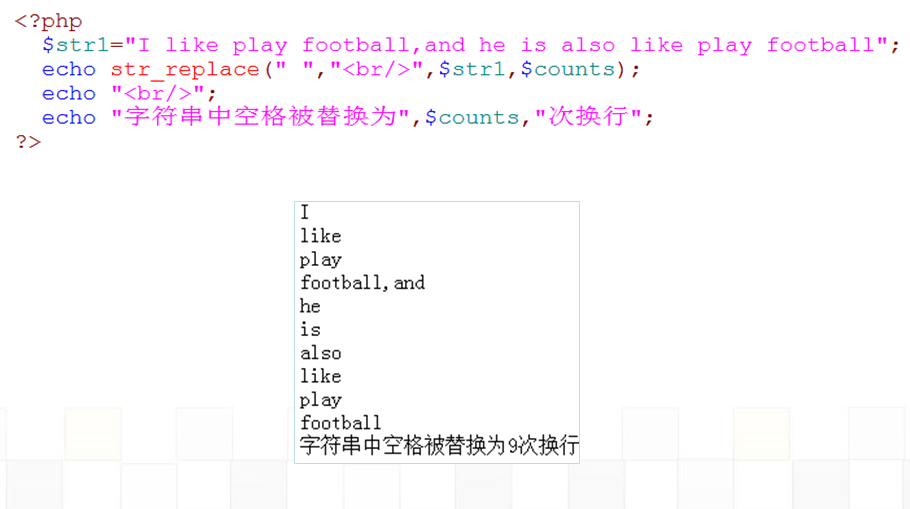
两者的区别在于strcasecmp()不区分比较字符串的大小写，

按照字节的ASCII值进行比较

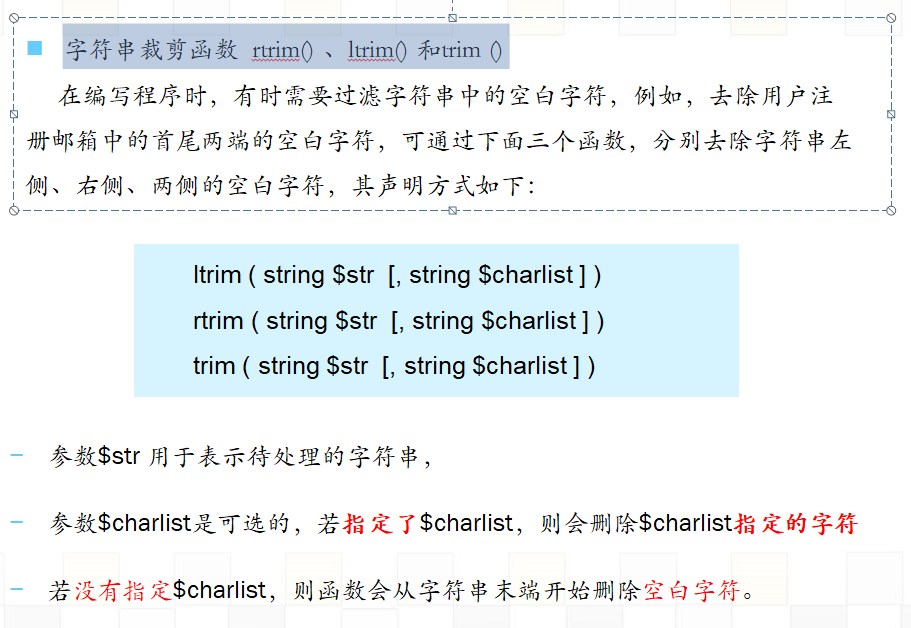
 

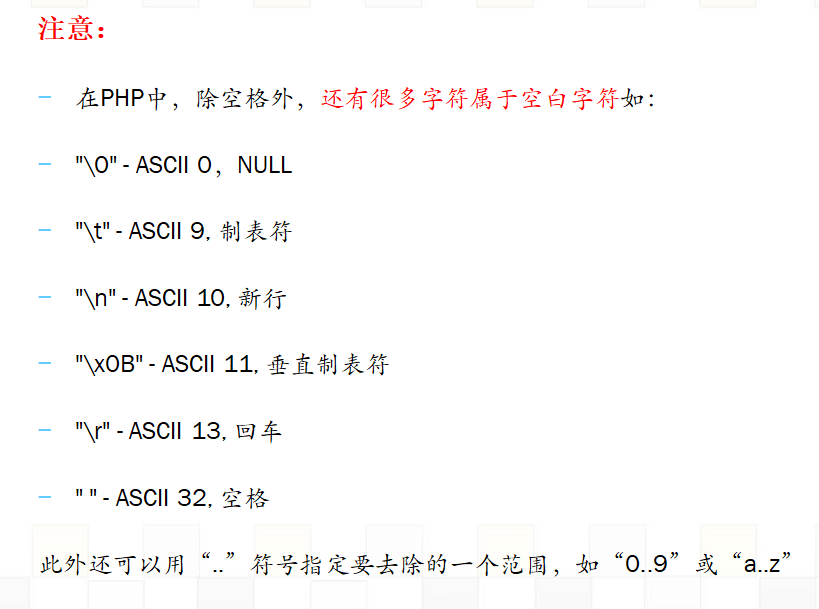
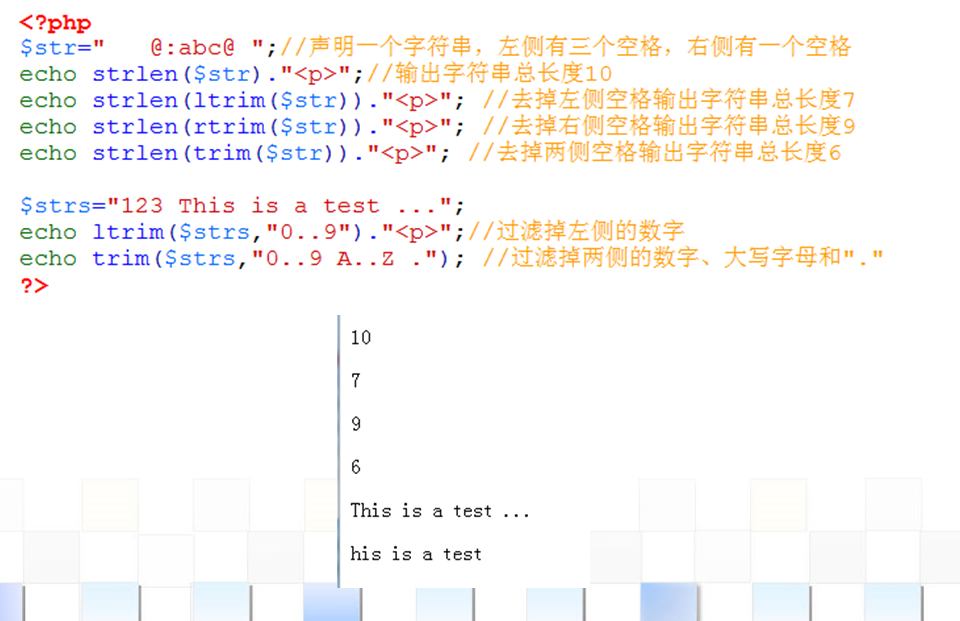
### 2.1.2字符串替换函数str\_replace()和str\_ireplace()

在程序开发中，经常会对字符串中的某些字符进行替换操作，这时可以通过str\_replace()函数和str\_ireplace()函数来完成。

### 2.1.3字符串裁剪函数 rtrim() 、ltrim() 和trim ()



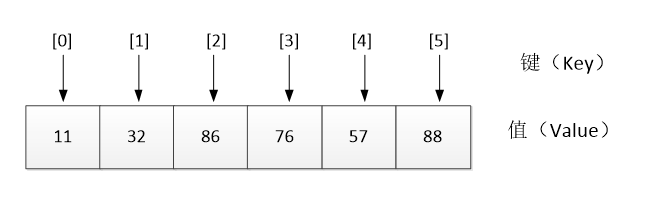
### basename() 函数返回路径中的文件名部分。

## 3．数组

### 3.1数组类型

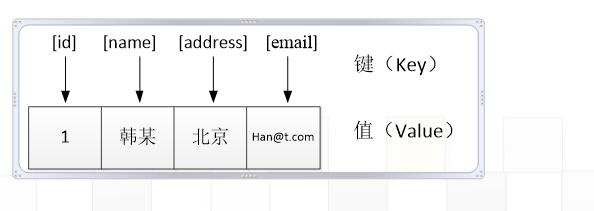
索引数组

指下标为整数的数组。索引数组的下标是从0开始，并依次递增



关联数组

下标是一些字符串



### 3.2数组定义

**赋值方式定义数组**



**array()函数定义数组**



### 3.3数组的打印

打印单个数组元素：$array\_name[键名]，如在上例中：

echo $arr[“name”] //返回 李玲

因为数组包含多个值，因而不能 echo $arr;

当数组使用字符串作为它的键时，把数组名和键装在花括号中。

echo “用户名是{$arr[‘name’]}”；

打印整个数组结构 print\_r()函数

print\_r($arr) //返回键 、值

var\_dump输出键、值、数据类型

### 3.4数组与字符串的转换

Implode(),explode()是常用的将数组和字符串之间转换的两个函数

implode()函数

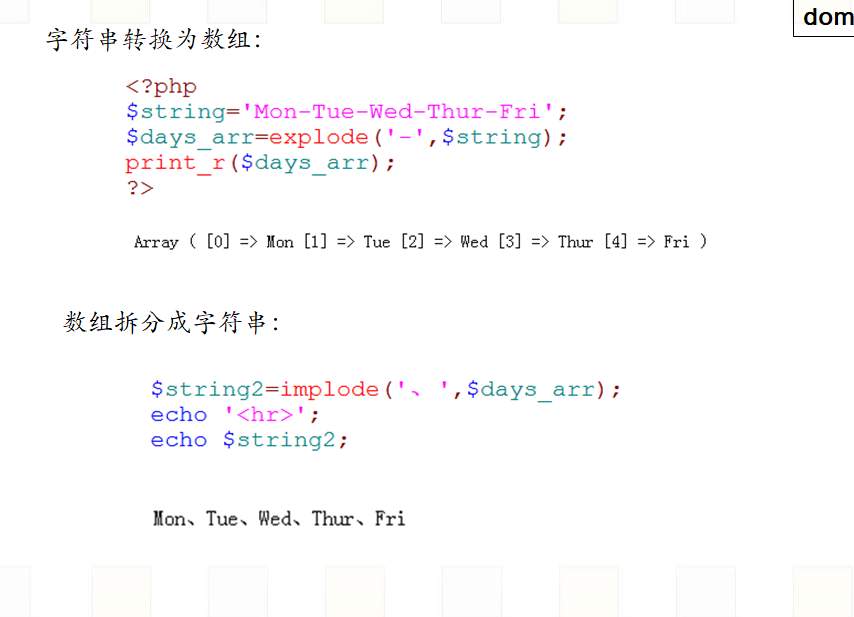
$string=implode($glue,$array)

将数组转换为字符串，需要定义用什么作分隔符$glue

explode()函数

$array=explode($separator,$string)

将字符串转变成数组时，要指定分隔符$separator通常可能为一个逗号、制表符或空格。



## 4函数

### 4.1 include()与require()引入文件函数

include()与require()的区别

include()在包含文件发生错误时，会发出警告，但脚本会继续运行。

而require()在包含文件发生错误时，会发出警告并停止脚本的运行

### 4.2php内置函数

**rand()生成随机数**

rand(min,max); main和max为两个可选参数，用于规定随机数产生的范围。

如果没有提供可选参数，rand()返回0到RAND\_MAX间的伪随机整数（32768）

在网页验证身份时常见的验证码功能即是由随机字符串、数字构成。

**MD5加密**

md5(string,charlist);

参数：string规定要计算的字符。

参数：charlist可选，默认为FALSE（32个字符十六进制）

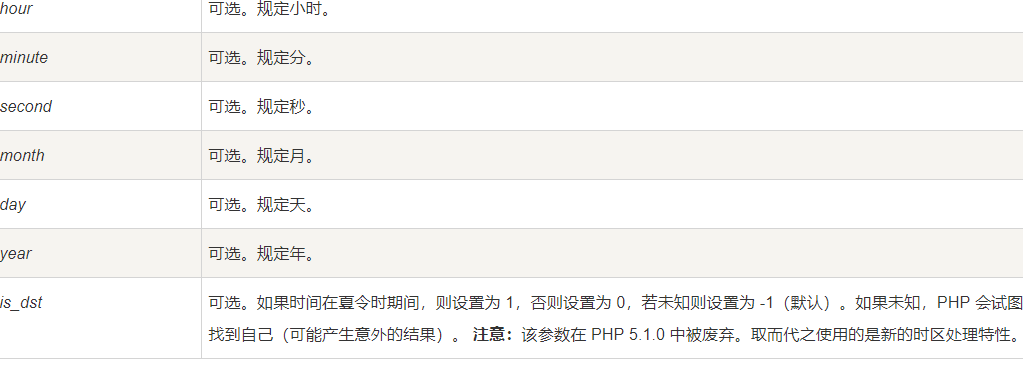
TRUE（16个字符二进制格式）

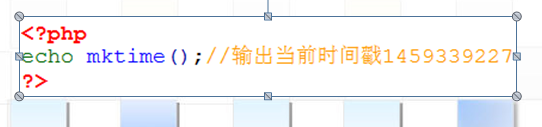
### 4.3时间戳函数

PHP提供了**mktime()函数**用于返回一个指定时间的时间戳。

mktime(hour,minute,second,month,day,year,is\_dst);

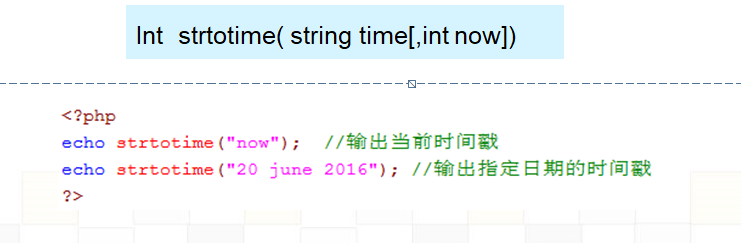
在上述声明中，该函数的返回值是一个Unix时间戳，参数可以从右向左省略，任何省略的参数会被设置成本地日期和时间的当前值



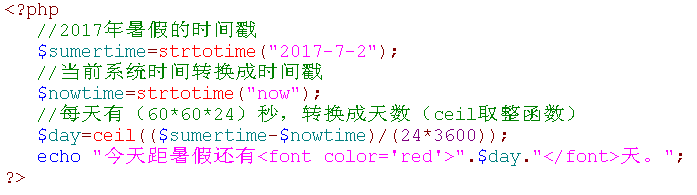


#### time()函数获取当前的UNIX时间戳

#### Strtotime()函数将字符串转为时间戳



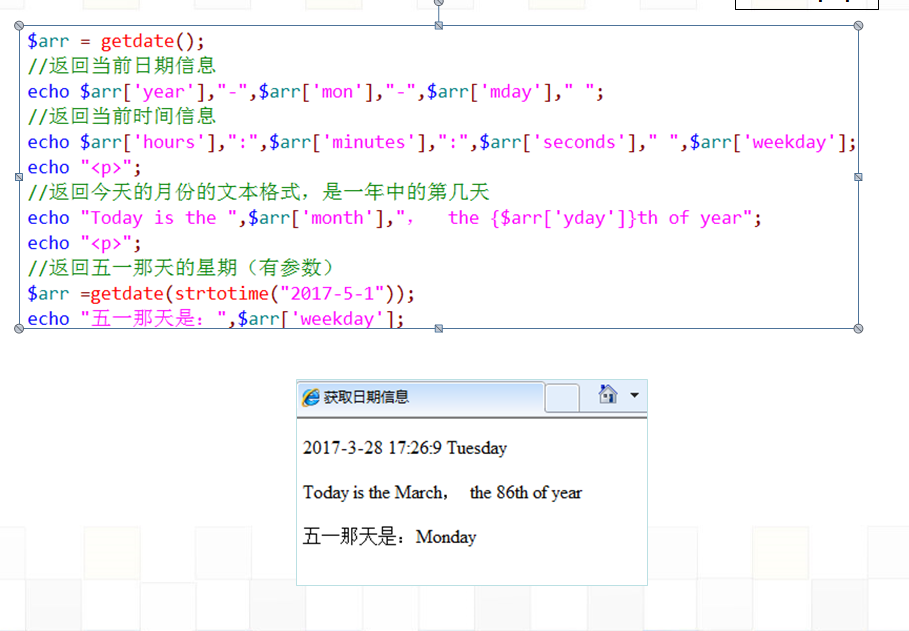
#### 日期的计算



#### getdate()函数取得时间信息

用getdate()函数取得时间信息, 返回**数组形式**的日期和时间信息。如果 没有参数 ，则返回当前时间戳。





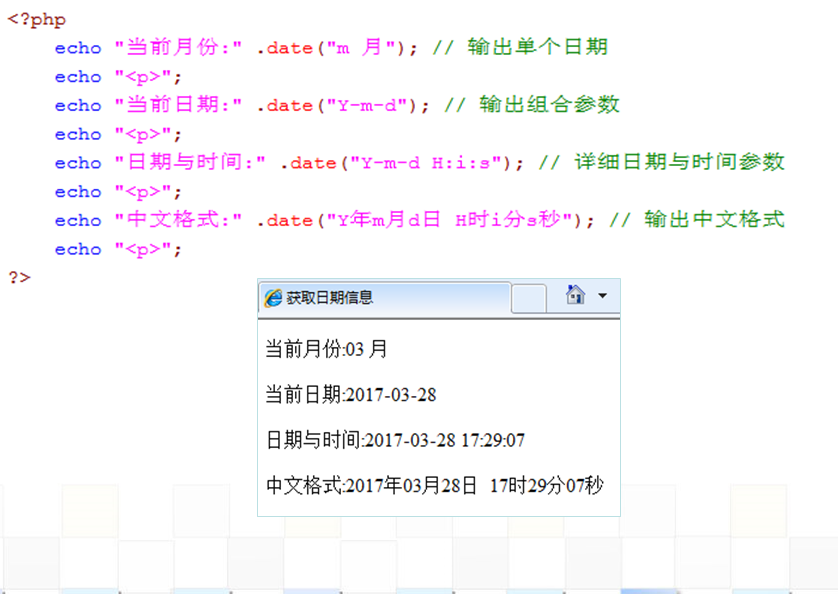
带参数返回某日的信息：

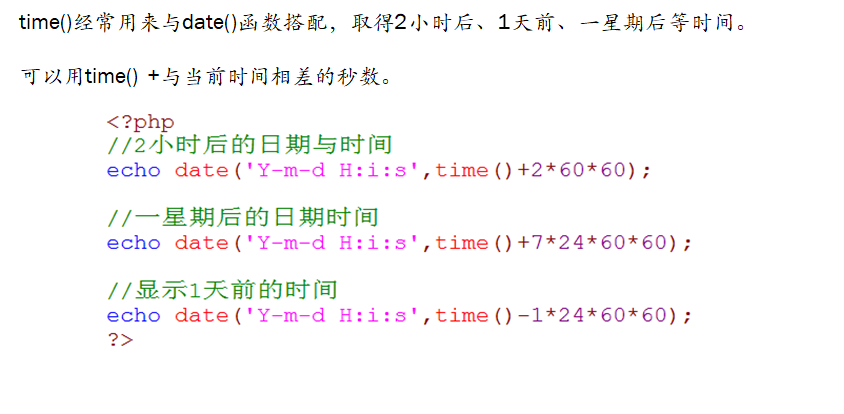
$arr =getdate(strtotime("2017-5-1"));

#### 日期和时间格式化输出

使用Unix时间戳保存和计算时间非常方便，但是时间戳的可读性很差，乍一看就是一个毫无意义的整型数值，所以调用date()函数格式化本 地时间和日期。







#### 修改PHP的默认时区

打开php.ini文件，修改date.timezone配置

date.timezone=PRC

### 3.1数组排序

#### 3.11 sort()对数组进行升序排列.rsort()对数组进行降序排列

$a=array(‘cww’,’add’,’sdd’);

Sort($a);---将数组中字按字母升序排列

输出—add,cww,sdd ,

#### 3.12 asort()与ksort()

asort() - 根据数组的值，对数组进行升序排列

ksort() - 根据数组的键，对数组进行升序排列

arsort() - 根据数组的值，对数组进行降序排列

krsort() - 根据数组的键，对数组进行降序排列

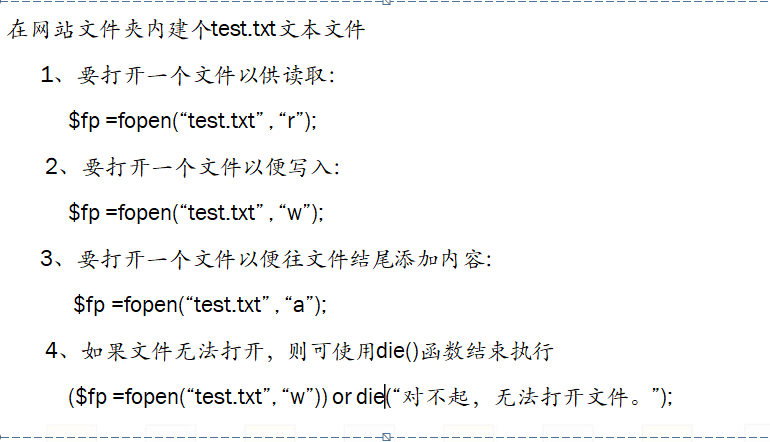
## 5.目录与文件操作

### 5.1文件的打开与关闭

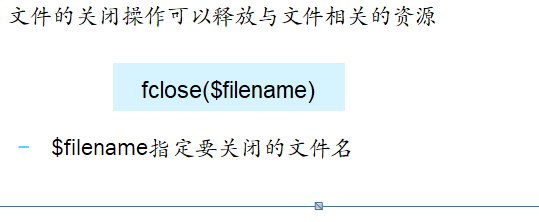
**打开文件fopen()函数**

fopen($filename,$mode)



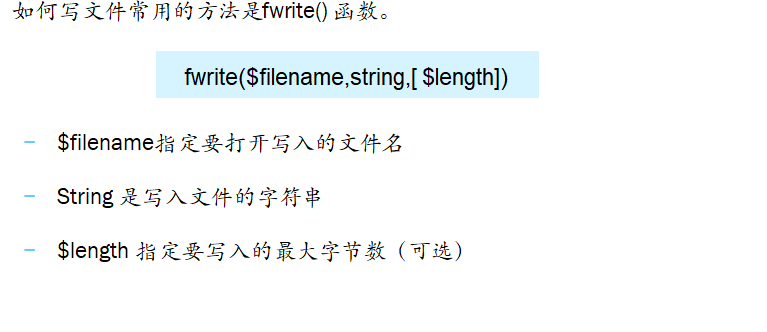


**关闭文件fclose()函数**



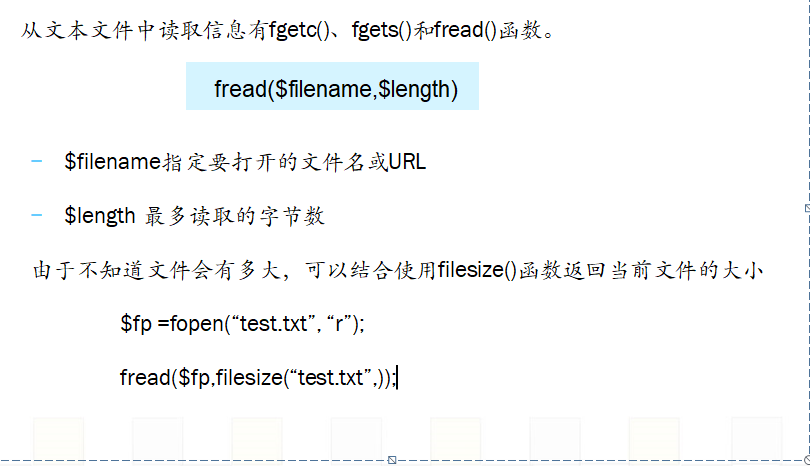
### 5.2文件的写入

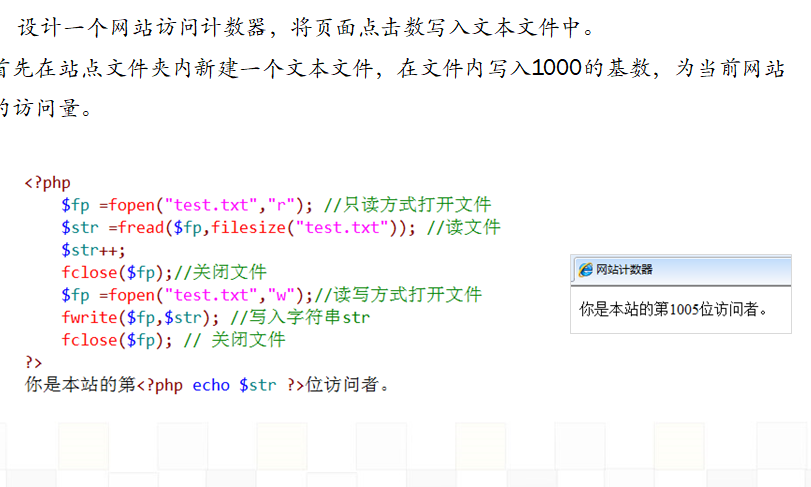
写文件fwrite()函数



### 5.3文件的读取

读文件fread()函数





### 5.4文件的上传

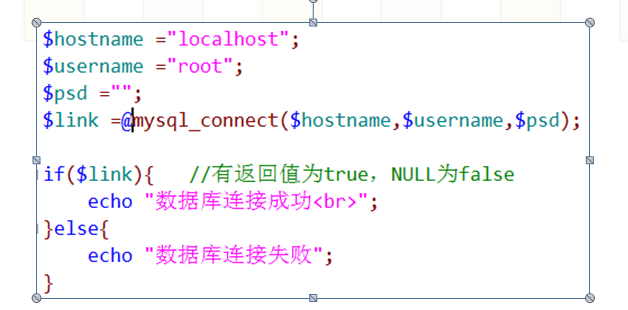




## 6.php操作数据库

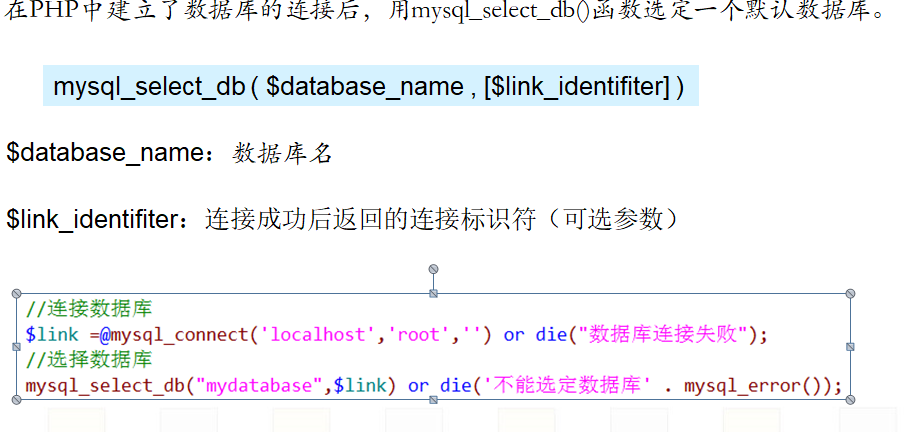
### 连接数据库

@mysql\_connect



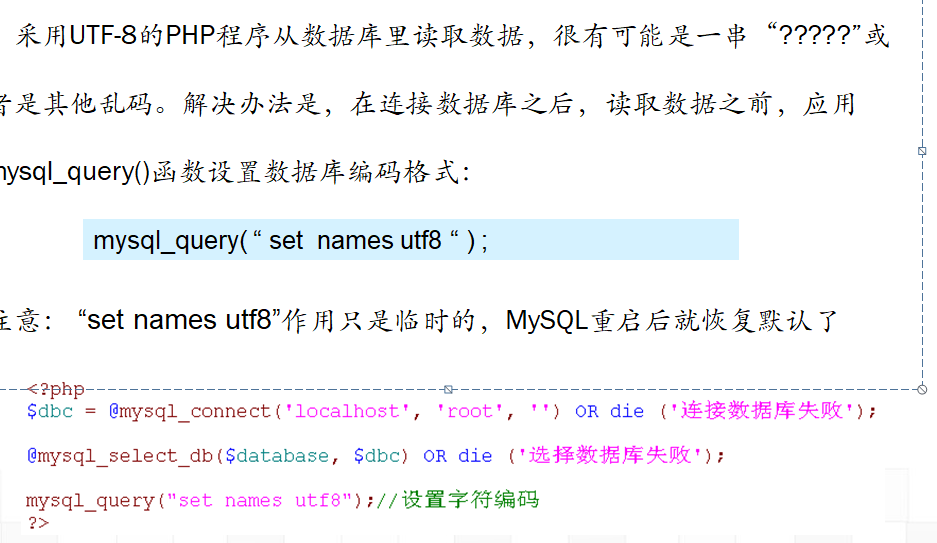
### 选择数据库

mysql\_select\_db

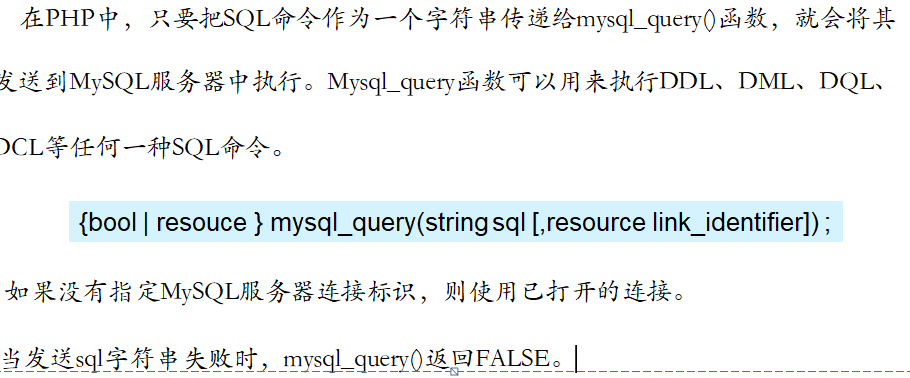


### 设置字符编码

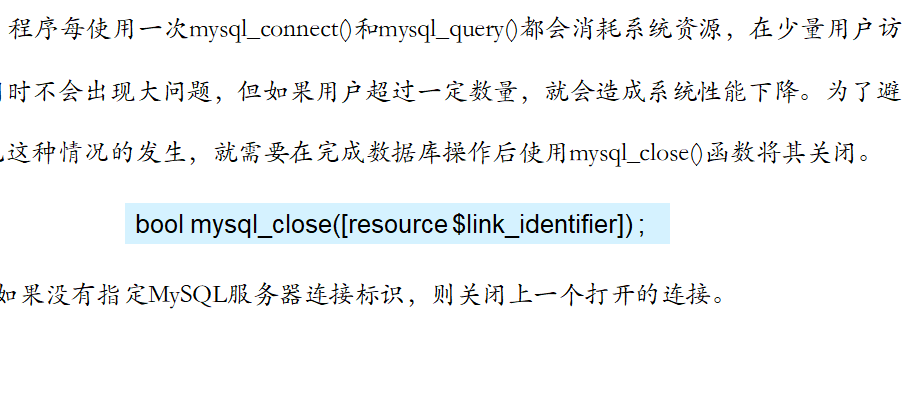
mysql\_query(“set name utf8”)



### 对数据库执行SQL命令



### 关闭连接



### 示例

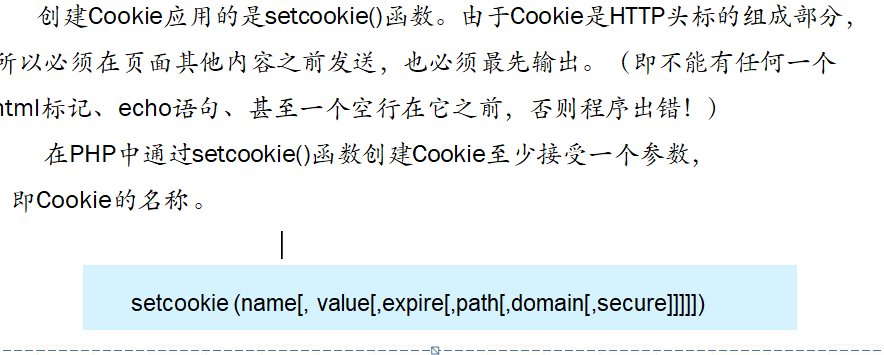


## 7.cookie

cookie是一种在远程客户端存储数据并以此来跟踪和识别用户的机制

### 创建cookie

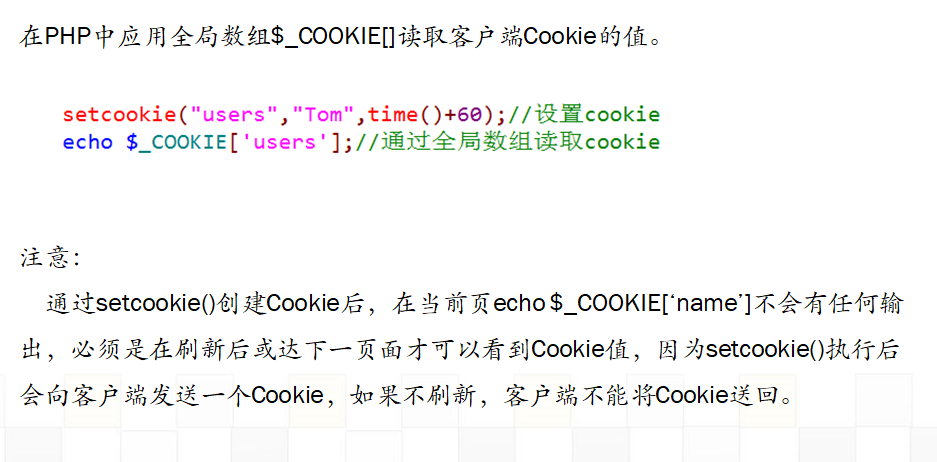
Setcookie(‘name’,12313,time()+50)





### 读取cookie

$\_COOKIE[‘name’]



### 使用Cookie存储数组





### 删除cookie



### Session与cookie区别

Session对象把信息记录在服务器端，Cookie则是记录在客户端。

Cookie可以保留相当长时间，直到记录被删或到失效日期。

Session对象在用户结束会话后就自动清除

Cookie传输的是明码信息；Session传输的只有SessionID，并且

每次登录的时候会生成不同的SessionID。

## 8.session使用

### 初始化会话

首先检查SessionID是否存在，如果不存在则使用session\_start()创建。

### 注册会话变量

会话变量保存在预定义数组$\_SESSION中，所以可以直接定义数组单元

的方式来定义一个会话变量：$\_SESSION[“name”] =value;

### 访问会话变量

利用$\_SESSION数组访问会话变量，可以用isset()或empty()函数来判断

会话变量是否存在。

### 销毁会话变量

Session\_destory（）

使用unset()函数删除会话变量

### 示例





## 9.trait

参考：<https://www.jianshu.com/p/fc053b2d7fd1>

php从以前到现在一直都是单继承的语言，无法同时从两个基类中继承属性和方法，为了解决这个问题，php出了Trait这个特性

用法：通过在类中使用use 关键字，声明要组合的Trait名称，具体的Trait的声明使用Trait关键词，Trait不能实例化

所以：Trait中的方法会覆盖 基类中的同名方法，而本类会覆盖Trait中同名方法

## 继承

子类可以直接是用父类的属性和方法不用再次声明

## 数组相关的处理函数

### 5.1 array\_values()

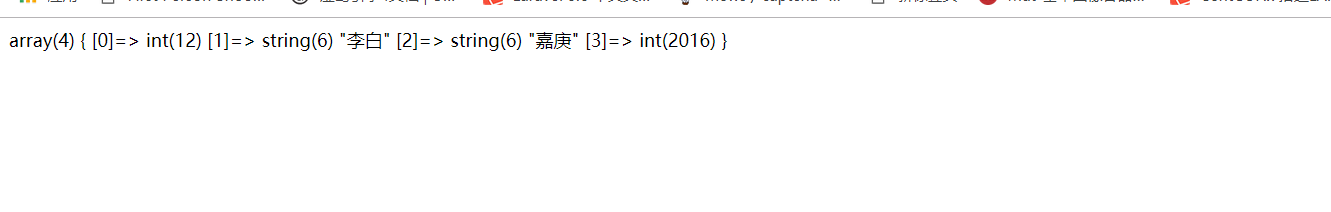
参数：arr---数组类型

作用：返回包含arr数组所有值的数组，返回的数组重新建立索引0,1,2,3…

适用于：数组元素下标混乱，或将关联数组转换成索引数组

例子：

**class** arr{  
 **public function** arrvalues($arr){  
 **return** *array\_values*($arr);  
 }  
  
}  
$arrdata=**array**(  
 **'ID'**=>12,  
 **'姓名'**=>**'李白'**,  
 **'学校'**=>**'嘉庚'**,  
 **'年级'**=>2016  
);  
$arr=**new** arr();  
$res=$arr->arrvalues($arrdata);  
*var\_dump*($res);



### 5.2array\_keys()

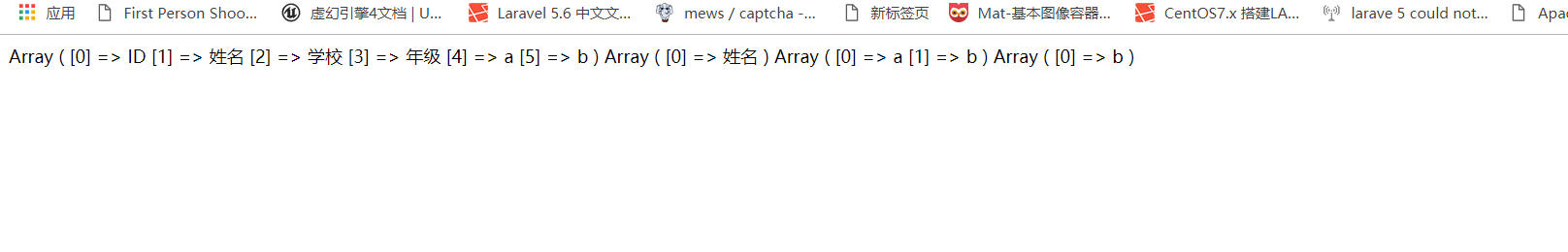
参数：arr---数组类型，参数2(可选)，指定返回键名的键值，参数3(可选)—true ,false,

True为全等，false 为不全等

作用：返回键名的键值

例子：

$arrdata=**array**(  
 **'ID'**=>12,  
 **'姓名'**=>**'李白'**,  
 **'学校'**=>**'嘉庚'**,  
 **'年级'**=>2016,  
 **'a'**=>10,  
 **'b'**=>**'10'**);  
*print\_r*(*array\_keys*($arrdata));  
*print\_r*(*array\_keys*($arrdata,**'李白'**));  
*print\_r*(*array\_keys*($arrdata,**'10'**,**false**));  
*print\_r*(*array\_keys*($arrdata,**'10'**,**true**));



Return $this 返回对象本身

参考：

<https://www.cnblogs.com/maodou/p/4230763.html>