**字符串、数组类型**

目录

[二、 字符串 4](#_Toc495332045)

[1. 常见的字符串定义语法 5](#_Toc495332046)

[1) 单引号 5](#_Toc495332047)

[2) 双引号 5](#_Toc495332048)

[3) heredoc字符串定义结构 5](#_Toc495332049)

[2. 字符串长度问题 6](#_Toc495332050)

[1) 计算长度的函数strlen 6](#_Toc495332051)

[2) 多字节字符串，如何计算字符串的字符数？ 7](#_Toc495332052)

[三、 字符串相关函数 7](#_Toc495332053)

[1. 字符串输出函数printf() 7](#_Toc495332054)

[1) printf 8](#_Toc495332055)

[2. 字符串替换与重复 8](#_Toc495332056)

[1) str\_replace() 8](#_Toc495332057)

[2) str\_repeat() 8](#_Toc495332058)

[3. 大小写转换函数 9](#_Toc495332059)

[1) strtolower() 9](#_Toc495332060)

[2) strtoupper() 9](#_Toc495332061)

[3) ucfirst() 9](#_Toc495332062)

[4. 去除空白函数 9](#_Toc495332063)

[1) trim(),ltrim(),rtrim() 9](#_Toc495332064)

[5. 数组与字符串转换函数 10](#_Toc495332065)

[1) explode() 10](#_Toc495332066)

[2) implode() 10](#_Toc495332067)

[3) str\_split() 11](#_Toc495332068)

[6. 位置查找函数 11](#_Toc495332069)

[1) strpos() 11](#_Toc495332070)

[2) strrpos() 12](#_Toc495332071)

[7. 截取子字符串函数 12](#_Toc495332072)

[1) substr() 12](#_Toc495332073)

[2) strchr()、strrchr() 13](#_Toc495332074)

[8. 字符串转义问题 13](#_Toc495332075)

[1) 常见的转义字符 13](#_Toc495332076)

[四、 数组入门 14](#_Toc495332077)

[1. 数组概念 14](#_Toc495332078)

[2. 创建数组的两种方式 14](#_Toc495332079)

[1) 显式创建 14](#_Toc495332080)

[2) 隐式创建 15](#_Toc495332081)

[3. 各种数组特点 15](#_Toc495332082)

[1) 索引数组： 15](#_Toc495332083)

[2) 关联数组 16](#_Toc495332084)

[3) 特例：混合数组 17](#_Toc495332085)

[4. 多维数组 17](#_Toc495332086)

[5. 数组元素的访问 17](#_Toc495332087)

[1) 一维数组元素访问 17](#_Toc495332088)

[2) 二维数组元素访问 18](#_Toc495332089)

[五、 数组遍历foreach 18](#_Toc495332090)

[1. 语法 18](#_Toc495332091)

[1) 第一种写法 19](#_Toc495332092)

[2) 第二种写法 19](#_Toc495332093)

[2. 遍历原理 19](#_Toc495332094)

[3. 二维数组遍历 20](#_Toc495332095)

[六、 数组常用函数 20](#_Toc495332096)

[1. 指针操作相关函数 20](#_Toc495332097)

[1) key($arr); 20](#_Toc495332098)

[2) current($arr); 21](#_Toc495332099)

[3) next($arr); 21](#_Toc495332100)

[4) reset($arr); 21](#_Toc495332101)

[5) 指针函数模拟数组遍历 21](#_Toc495332102)

[2. max(),min(),count()，in\_array() 22](#_Toc495332103)

[1) max,min 22](#_Toc495332104)

[2) count（$arr） 22](#_Toc495332105)

[3) in\_array 22](#_Toc495332106)

[3. range(),array\_merge(),array\_rand(),shuffle 22](#_Toc495332107)

[1) range 23](#_Toc495332108)

[2) array\_merge 23](#_Toc495332109)

[3) array\_rand 24](#_Toc495332110)

[4) shuffle 24](#_Toc495332111)

[5) 使用数组函数生成验证码字符串 24](#_Toc495332112)

[4. 键值相关函数 25](#_Toc495332113)

[1) array\_keys() 25](#_Toc495332114)

[2) array\_values() 25](#_Toc495332115)

[5. 数组排序函数sort,rsort,asort,arsort,ksort 26](#_Toc495332116)

[1) sort() 26](#_Toc495332117)

[2) asort() 26](#_Toc495332118)

[3) rsort() 26](#_Toc495332119)

[4) arsort 27](#_Toc495332120)

[5) ksort 27](#_Toc495332121)

[6) krsort 28](#_Toc495332122)

[7) 总结 28](#_Toc495332123)

[七、 数据库简介 28](#_Toc495332124)

[1. 数据库概念 28](#_Toc495332125)

[2. 关系型数据库 29](#_Toc495332126)

[1) 数据库的组织架构： 29](#_Toc495332127)

[2) 关系型数据库产品—MySQL介绍 30](#_Toc495332128)

[3. SQL（了解） 30](#_Toc495332129)

[1) 概念 30](#_Toc495332130)

[2) 分类 31](#_Toc495332131)

[3) SQL语句书写规范 32](#_Toc495332132)

[八、 MySQL数据库服务器访问 32](#_Toc495332133)

[1. 访问步骤 32](#_Toc495332134)

[2. 开启数据库服务器 33](#_Toc495332135)

[3. 客户端连接数据库服务器 33](#_Toc495332136)

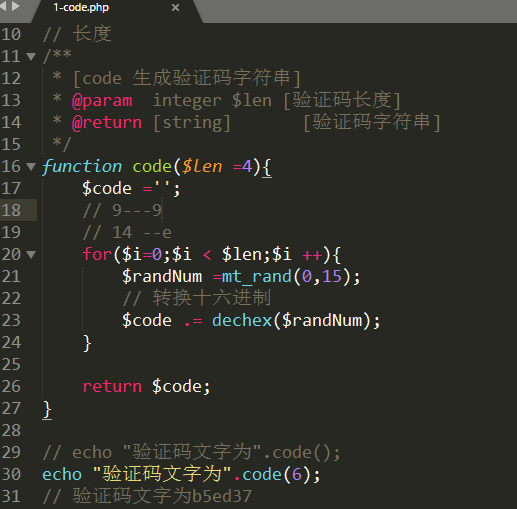
[1) 命令行连接命令 33](#_Toc495332137)

[2) 设置字符集 35](#_Toc495332138)

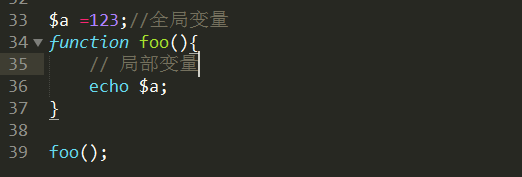
[3) 退出 36](#_Toc495332139)

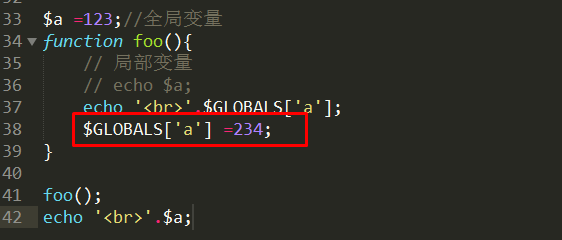
**回顾：**

1. 函数封装



1. 变量作用域



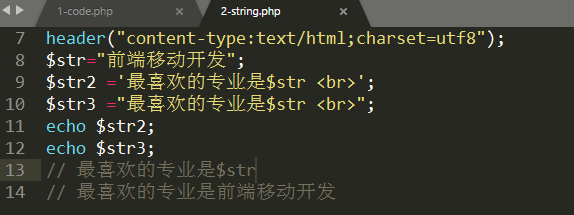


# 字符串

## 常见的字符串定义语法

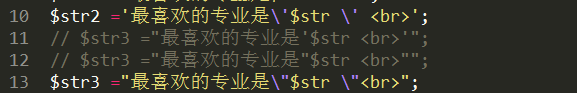
### 单引号

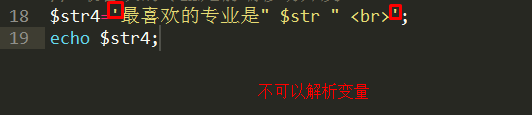
* 不解析变量。
* 单引号中包含单引号需要转义
* 单引号中中可以包含双引号



### 双引号

* 可以解析变量
* 双引号其中包含双引号，需要转义
* 变量最好使用变量{}边界符号
* 单引号与双引号嵌套的问题：只有最外层是双引号时，才能解析变量





### heredoc字符串定义结构

单引号及双引号，在多行字符串、复杂字符串如JS代码的定义时显得不便。

* heredoc语法结构

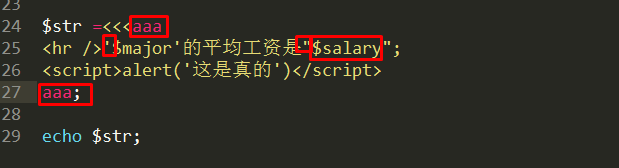
语法：

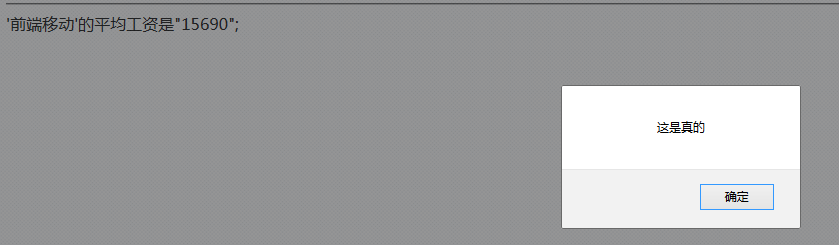
$str =<<<heredoc标记名

字符串内容

heredoc标记名;

* 注意事项
* 以<<<开头，其后跟着自定义的标识符。命名遵循PHP标识符命名规范。
* 在换行的位置，开始字符串的书写。
* 字符串输入完成后，换行输入原先自定义的标识符。标识符必须为本行第一列。
* 两处的标识符必须一致。
* 可以解析变量。单引号及双引号不需要转义或者处理。一般用于保存较长字符串或者JS代码等。

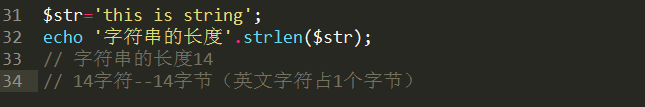




## 字符串长度问题

### 计算长度的函数strlen

返回的是单字节字符串的字节数（等于字符数）。多字节字符返回对应单字节的长度，即字节数。



### 多字节字符串，如何计算字符串的字符数？

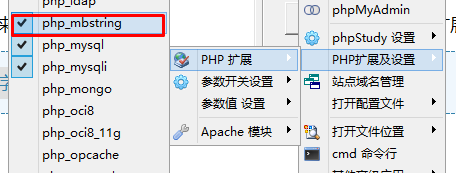
UTF-8字符集中，一个汉字占3个字节，一个英文字符占1个字节。



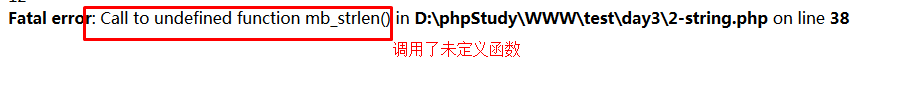
PHP提供多字节字符串扩展模块php\_mbstring.dll(动态链路库文件)。开启该扩展后，可以使用mb\_strlen函数来统计多字节字符串的字符长度。

mb\_strlen(字符串，使用的字符集);

在PHPstudy中，该扩展默认开启的：



函数库（扩展）未加载的情况：





# 字符串相关函数

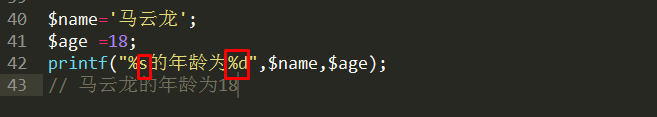
## 字符串输出函数printf()

### printf

printf ( string $format [, mixed $args [, mixed $... ]] )

依据 format 格式参数产生输出。

* 占位符：%d: 十进制有符号整数 %s:字符串。
* 注意事项参数列表与占位符一一对应。



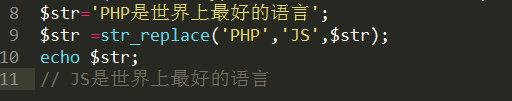
## 字符串替换与重复

### str\_replace()

str\_replace ( mixed $search , mixed $replace , mixed $subject )

该函数返回一个字符串或者数组。该字符串或数组是将 subject 中全部的 search 都被 replace 替换之后的结果。

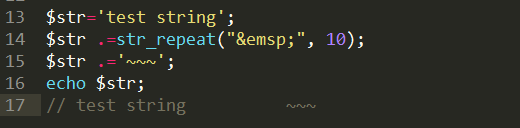
返回替换后的字符串。必须接收替换后的结果。



### str\_repeat()

string str\_repeat ( string $input , int $multiplier )

返回 input 重复 multiplier 次后的结果。



## 大小写转换函数

### strtolower()

lower case :小写

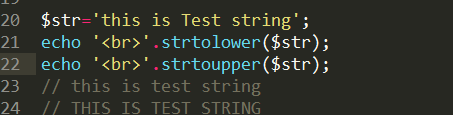
string strtolower ( string $string )

将 string 中所有的字母字符转换为小写并返回。

### strtoupper()

string strtoupper ( string $string )

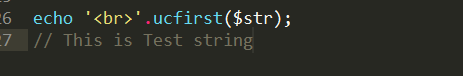
upper case：大写。将 string 中所有的字母字符转换为大写并返回。



### ucfirst()

string ucfirst ( string $str )

uc: upper case，大写。将 str 的首字符（如果首字符是字母）转换为大写字母，并返回这个字符串。

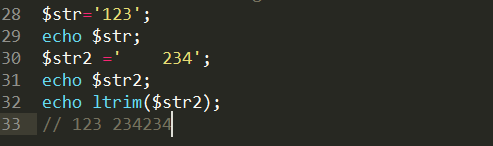


## 去除空白函数

### trim(),ltrim(),rtrim()

string trim ( string $str [, string $charlist = " \t\n\r\0\x0B" ] )

此函数返回字符串 str 去除首尾空白字符后的结果。可以用在表单提交数据的处理。l,left,r,right

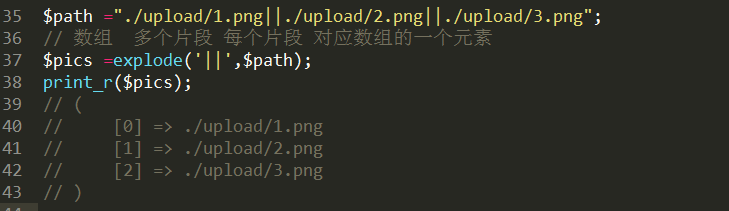


## 数组与字符串转换函数

### explode()

explode（分隔符，目标字符串）；

将目标字符串使用分割符分割，返回子字符串组成的数组。





### implode()

implode（连接的字符，数组）；

将数组的元素使用连接字符串拼接成新的字符串

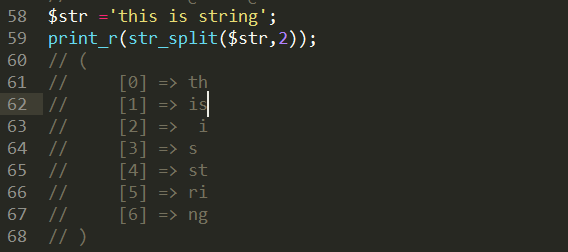


### str\_split()

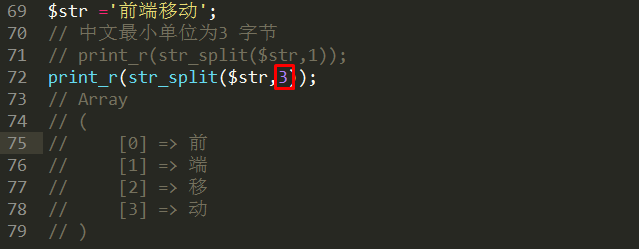
str\_split($str[,子字符串的长度=1])；

将字符串分成指定长度（默认为1字节）的子字字符串。返回数组。

* 英文字符分割



* 中文字符的分割

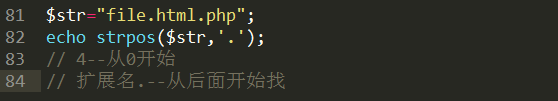


## 位置查找函数

### strpos()

strpos(目标字符串，要查找的子字符串)；

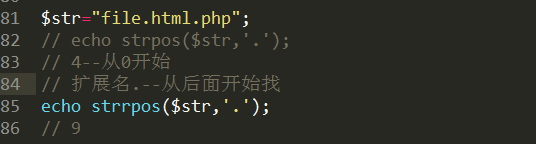
查找子字符串在目标字符串中的**首次**出现的位置。从左侧开始查找。下标从0开始。position,位置



### strrpos()

strrpos(目标字符串，要查找的子字符串)；

r: revese,反转. ,从右侧开始查找,查找子字符串在目标字符串中首次出现的位置。下标从0开始。



## 截取子字符串函数

### substr()

substr(目标字符串，起始下标[,截取的长度]);

截取指定位置开始的子字符串。省略第三个参数，直接截取到末尾。

案例：获取文件的扩展名



### strchr()、strrchr()

strchr(目标字符串，截取的起始子字符串)；

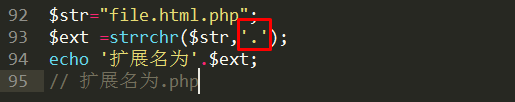
//strstr

从左侧，自字符串出现的位置截取到末尾。

strrchr(目标字符串，截取的起始子字符串)；

从右侧，自字符串出现的位置截取到末尾。

案例：获取文件的扩展名



## 字符串转义问题

单引号与双引号在特殊场合需要通过添加\实现转义。转义字符必须包含在双引号之中，才能被正确解析。

### 常见的转义字符

\n 换行

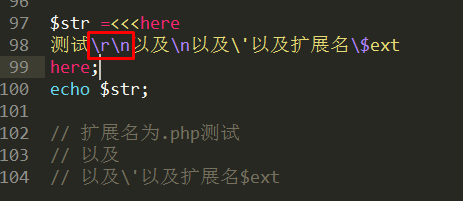
\r 回车

\t 水平制表符

\' 单引号

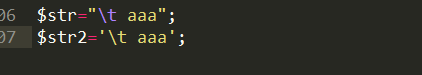
\" 双引号

\$ 美元标记



在windows,linux,mac os不同平台，中使用的换行符不一样，尽量使用\r\n表示。

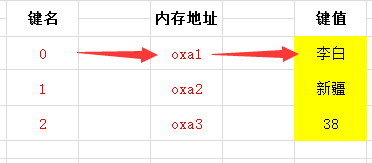
此外，转义字符只有在双引号中以及heredoc结构中才能正确的解析。



# 数组入门

学生信息、文章相关信息、订单相关信息都有很多方面，在程序中应该保存在什么类型的变量中？--数组

## 数组概念

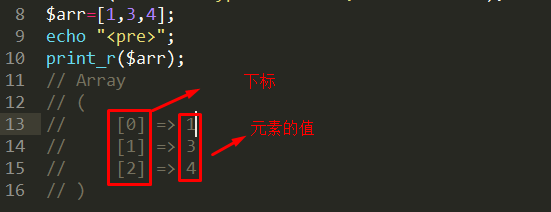


* 数组是内存中一块连续的空间（堆区），用来保存不同类型的数据。数据没有长度限制。
* 数组由多个元素（element）组成。每个元素有下标，称为键(key)；元素有值(value)。数组由多个键值对（key-value）组成。

## 创建数组的两种方式

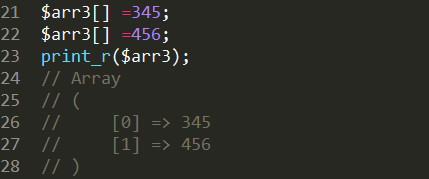
### 显式创建

* 下标可以为数字，也可以为字符串。
* PHP5.4支持数组的短标记方式[]



### 隐式创建

不使用array关键字，直接向数组中添加元素。





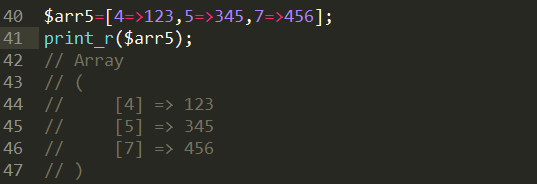
## 各种数组特点

### 索引数组：

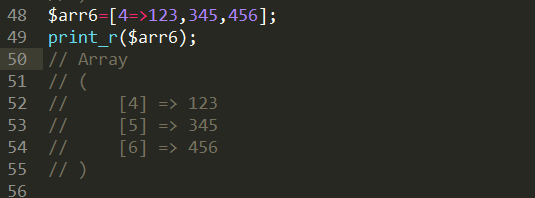
又称枚举数组

* 下标仅为一个索引，指示元素在数组中的位置。不表示对应值的含义。

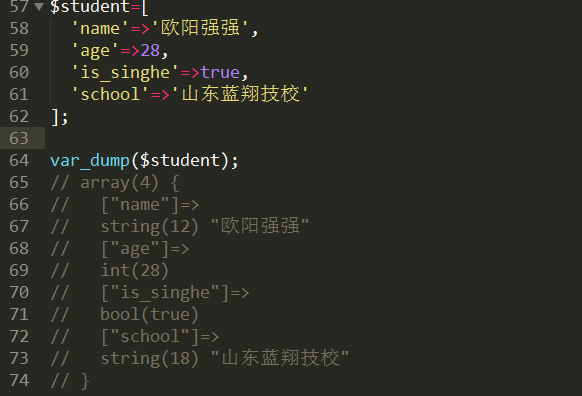




* 下标为整型
* 不指定元素下标时，自动从0开始分配。
* 部分元素省略下标时，新增的元素的下标，在原来下标的最大值基础上增长。



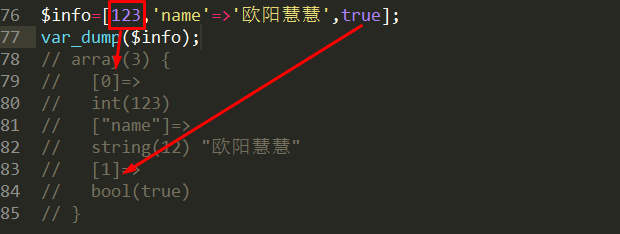
### 关联数组



* 下标为字符，除了表示元素在数组中的位置，还可以描述元素的意义。
* 定义数组元素时，下标使用引号包括（只要是字符串，则必须使用引号，否则会被PHP解析为常量）。
* 显示时，以元素的添加顺序有关，与下标的排序无关。

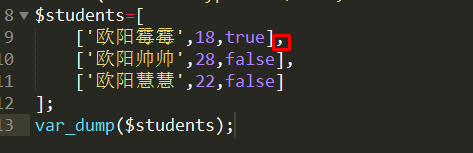
### 特例：混合数组

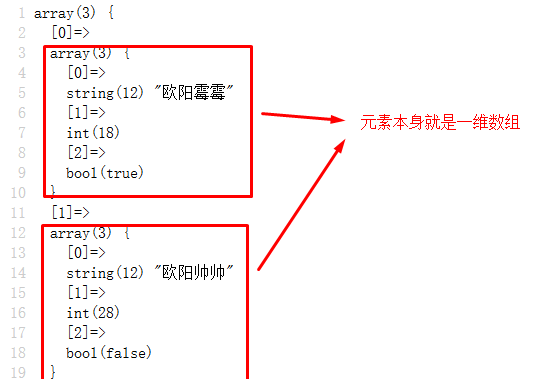
既有数字下标又有字符串下标的数组。



## 多维数组

数组的元素可能是一个数组。用来表示如多个学生信息，订单信息，文章信息等。

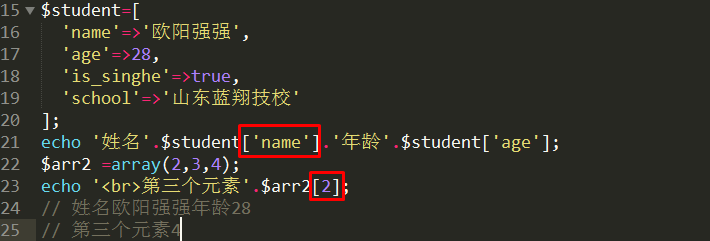




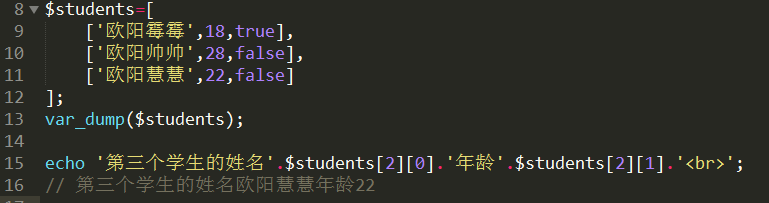
## 数组元素的访问

使用元素的下标定位元素。

### 一维数组元素访问

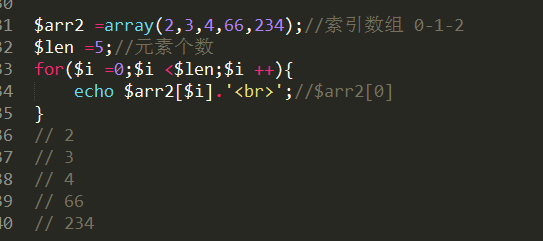


### 二维数组元素访问



# 数组遍历foreach

索引数组的循环读取：



循环读取索引数组元素很好实现，如何实现关联数组元素的循环读取？

## 语法

foreach($arr as [$key =>] $value){

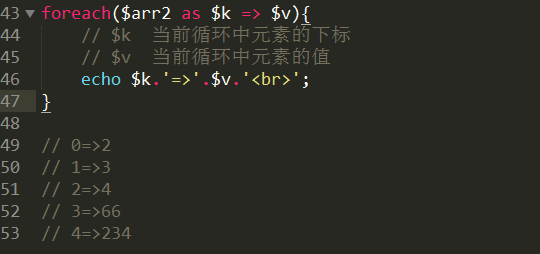
//循环体

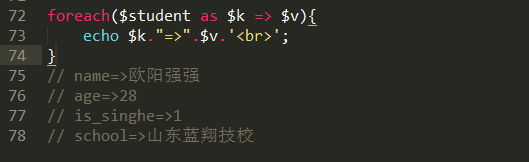
}

$key可以自行命名，该位置的变量表示当前循环对应元素的下标

$value可以自行命名，该位置的变量表示当前循环对应元素的值

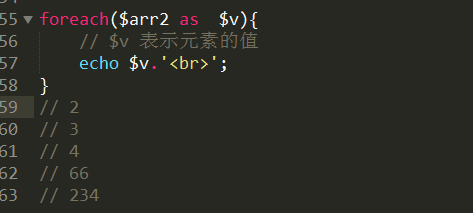
### 第一种写法





### 第二种写法

循环时不使用元素的下标，可以简写。

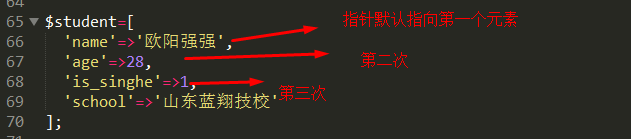


## 遍历原理

数组中存在指针。数组的遍历是靠操作指针实现的。

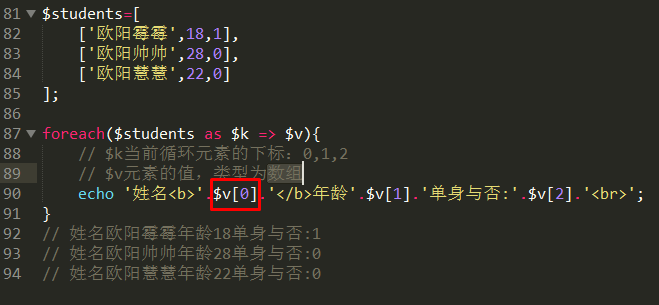
遍历时，获取当前指针指向的元素的键及值，分别赋值给变量$key 及$value。然后移动指针一次。

当获取最后一个元素后，指针向下移动，非法，循环自动终止。



## 二维数组遍历

特点：在每次循环中，读取出来的元素本身还是数组。



# 数组常用函数

## 指针操作相关函数

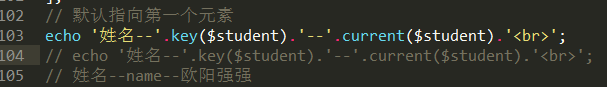
数组中存在指针，创建一个数组时，指针默认指向数组的第一个元素。foreach遍历时，自动移动指针，指向下一个元素，即可获取不同元素的值及键名。此外，每次循环中都会判断指针是否非法，指针非法时循环自动终止。

### key($arr);

获取当前指针指向的元素的键名。

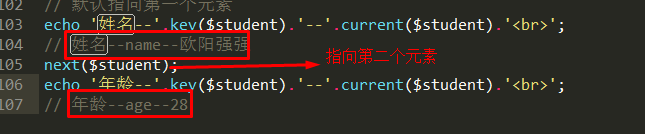
### current($arr);

获取当前指针指向元素的键值。



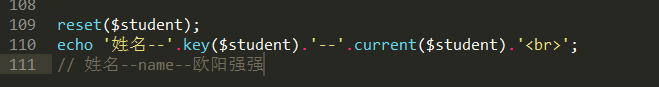
### next($arr);

指针移动一次，指向下一个元素。

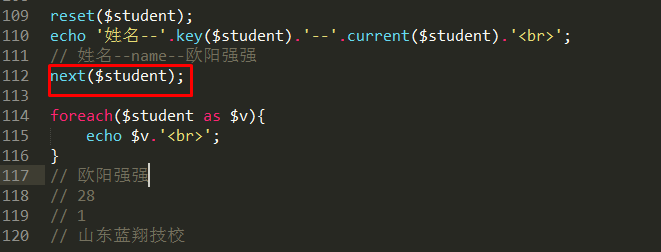


### reset($arr);

指针复位。指针指向数组的第一个元素。



foreach中每次开始需要执行reset函数，将指针复位。

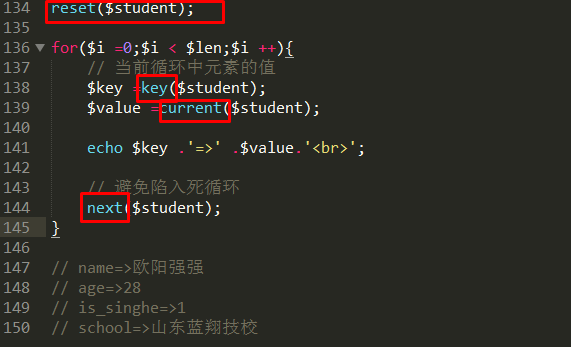


### 指针函数模拟数组遍历

第一次循环开始之前：复位reset

循环中：获取键值对key,current

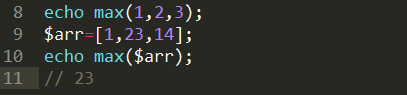
移动指针：next



## max(),min(),count()，in\_array()

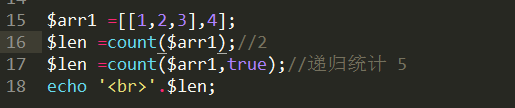
### max,min

可以返回数组元素中最大值及最小值。



### count（$arr）

统计数组元素的个数。



### in\_array

in\_array(目标元素，目标数组);

查找目标数组中是否存在该元素。字符串元素的比较时，区分大小写。

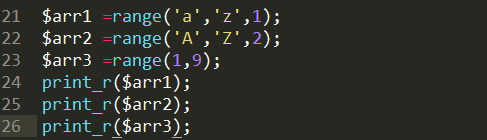


## range(),array\_merge(),array\_rand(),shuffle

### range

range(起始元素，终止元素[，步长=1])

返回位于起始元素及终止元素之间值，组成的新数组。可以设置元素值的步长。



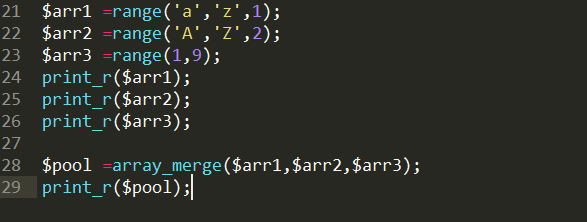


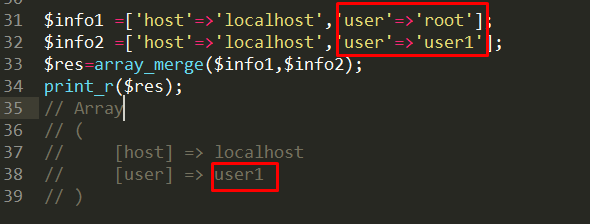
### array\_merge

array\_merge($arr1,$arr2);

合并多个数组

关联数组合并时，若元素下标相同，后面的数组的元素覆盖前面的。索引数组不覆盖

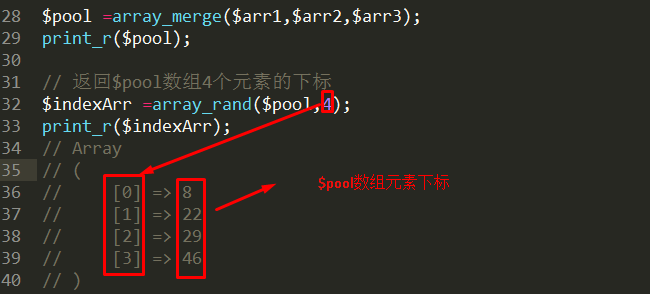




### array\_rand

array\_rand(目标数组，键的个数);

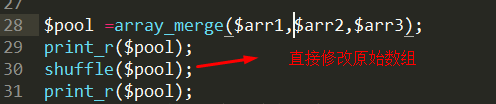
随机返回目标数组中指定个数的键名，以数组的形式返回。



### shuffle

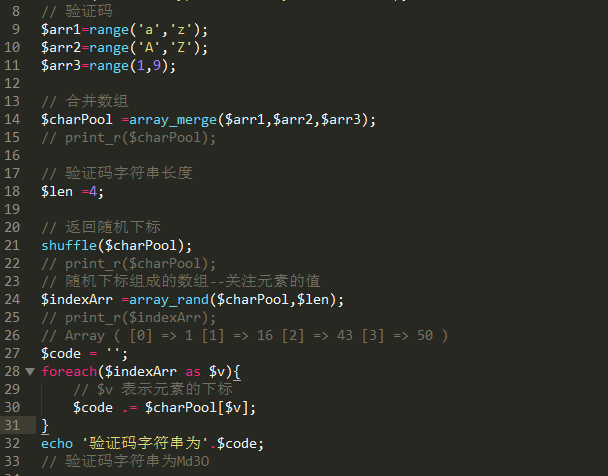
shuffle($arr);

打乱数组，增加数组元素的随机性。



### 使用数组函数生成验证码字符串

验证码字符串是由英文大小写，数字组成的指定长度随机字符串。



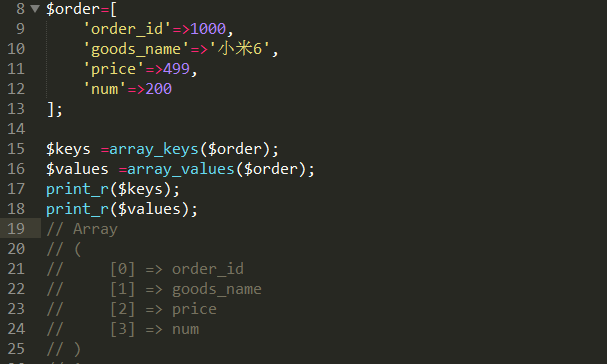
## 键值相关函数

### array\_keys()

返回数组中元素的键名组成的数组。

### array\_values()

返回数组中所有元素的值组成的索引数组。

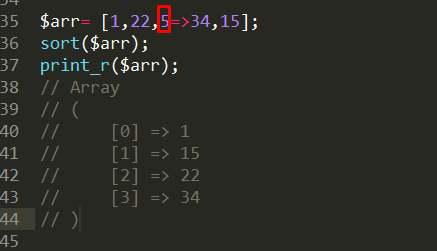


## 数组排序函数sort,rsort,asort,arsort,ksort

### sort()

sort($arr);

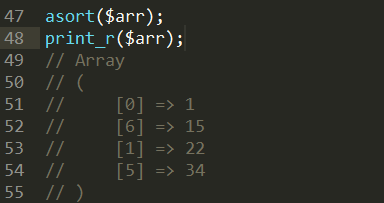
对数组元素进行升序排序，重建数字索引。



### asort()

asort($arr);

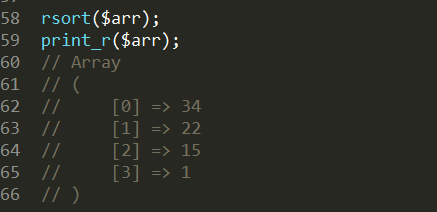
对数组元素进行升序排序，保持索引（a, Associative Array，关联数组）。



### rsort()

rsort($arr);

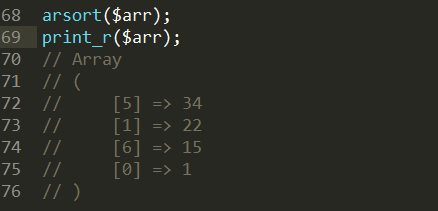
对数组元素进行降序排列(r,reverse,逆向)，重建索引。



### arsort

arsort($arr);

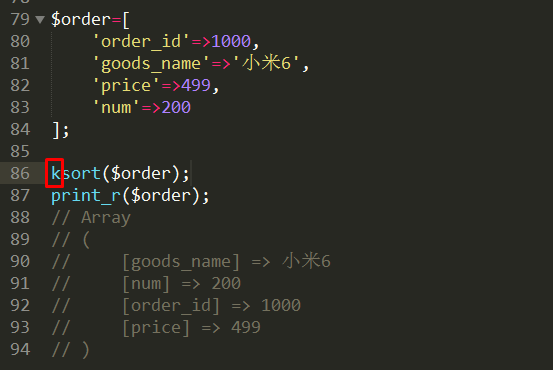
对数组元素进行降序排列，保持索引。



### ksort

ksort($arr)

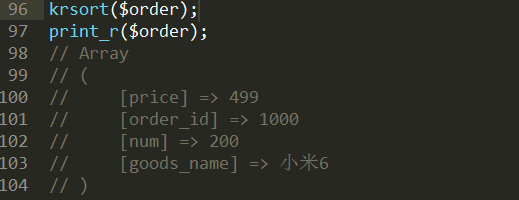
按照键名（key）升序排序，主要用于关联数组。保存数组的索引。



### krsort

krsort($arr)；

按照键名降序排序，主要用于关联数组。保存数组的索引。



### 总结

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 函数 | sort | asort | rsort | arsort | ksort | krsort |
| 按照下标或者值 | 值 | 值 | 值 | 值 | 键名 | 键名 |
| 升序/降序 | 升 | 升 | 降 | 降 | 升 | 降 |
| 索引保持 | N | Y | N | Y | Y | Y |

# 数据库简介

WAMP =Windows + Apache web服务器 + MySQL数据库服务器 + PHP

**一个问题?网站页面展示的大量动态数据存储在哪里？--数据库服务器中**

## 数据库概念

专门用来存储、管理数据的仓库（空间），按照数据结构来组织、存储和管理，可以实现高效存储及读取的数据解决方案。英文：Database,DB。

开发中，一般将项目相关的数据全部存储在数据库中，便于数据的安全、高效管理。

## 关系型数据库

当前使用范围最广的数据库。按照关系模型（运用真实世界中事物与关系的观念，来解释数据库中的抽象的数据架构）组织的数据库。数据存储在硬盘文件中。

如 订单：信息包括订单编号，商品名称，价格，数量，配送地址

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **订单编号order\_id** | **商品名称**  **goods\_name** | **价格price** | **数量**  **num** | **配送地址**  **address** |
| 100123 | 华为mate10 | 4999 | 2 | 北京市京顺路99号 |
| 100124 | YSL花裙子 | 499 | 1 | 北京市清华大学1号楼 |

关系模型是采用二维表格（行、列）来组织数据。相对于其他模型，更贴合现实世界，更便于理解。

### 数据库的组织架构：

* 数据表（table）

表是以行和列的形式组织起来的数据的集合。项目一般需要不同的数据表，将数据分布存储在不同的数据表中。如在电商项目中一般至少需要：，用户表，订单表，商品表，配送地址表，商户表等，

数据库服务器

天猫DB1

数据表1

* 记录（row，record）

一行有意义的信息的集合

* 字段(column，field)

某一列的值的名称。

* 通过行列交叉确定具体数据

### 关系型数据库产品—MySQL介绍

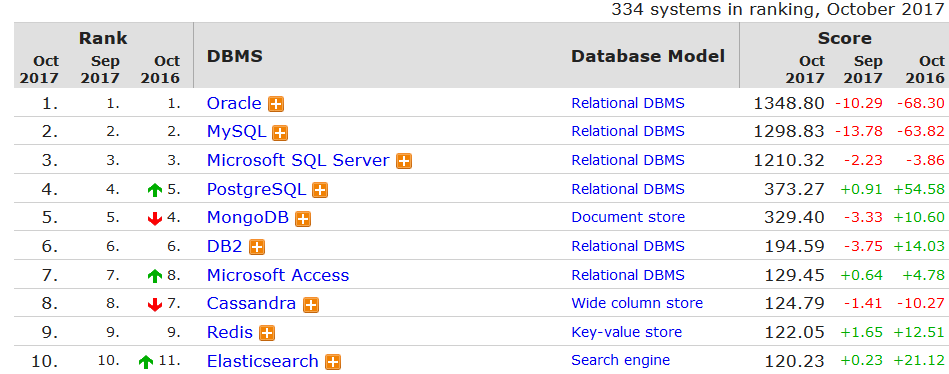
瑞典AB公司开发，现为Oracle公司所有。2010年04月22 发布了MySQL 5.5。

数据库官网地址：<https://dev.mysql.com>

英文手册地址：<https://dev.mysql.com/doc/refman/5.5/en/>

中文参考论坛：<http://www.mysqlpub.com/>

MySQL由于性能高、成本低、可靠性好，已经成为最流行的开源数据库，用于大规模网站和应用，比如[维基百科](https://zh.wikipedia.org/wiki/%E7%BB%B4%E5%9F%BA%E7%99%BE%E7%A7%91)、[Google](https://zh.wikipedia.org/wiki/Google)和[Facebook](https://zh.wikipedia.org/wiki/Facebook" \o "Facebook)、国内的淘宝等网站。非常流行的开源软件组合WAMP,[LAMP](https://zh.wikipedia.org/wiki/LAMP)中的“M”指的就是MySQL。



## SQL（了解）

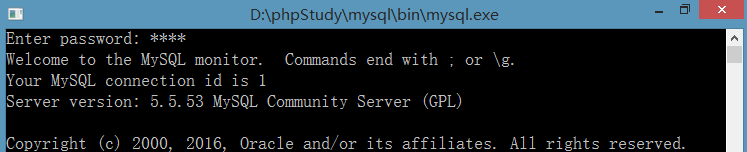
### 概念

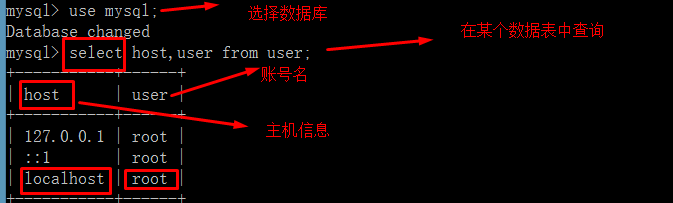
SQL：structured Query Language结构化查询语言。关系型数据库所使用通用语言。为一个ISO标准。

### 分类

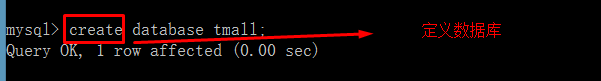
根据操作对象（数据库、数据表、字段）或者层次的不同，分为：

* **数据查询语句**(DQL,Data Query Language)：SELECT（查询数据）



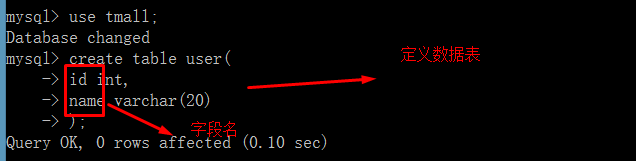


* **数据定义语句**(DDL, Data Definition Language)，定义存储数据的结构（数据库，数据表）：**CREATE**（创建）、**DROP**（删除）、**ALTER**（修改）

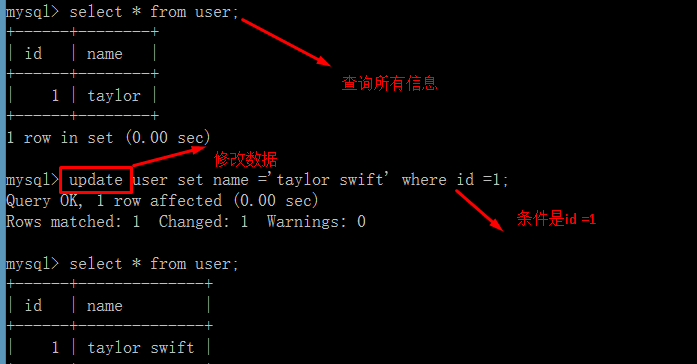


* **数据操纵语句**(DML , Data Manipulation Language)：**INSERT**（插入数据）、**UPDATE**（更新数据）、**DELETE**（删除数据）

注意是对数据进行操作。







* **数据控制语句**(DCL, Data Control Language)：GRANT（授权，给用户分配权限）、REVOKE（取消授权）

保证数据的安全，数据操作之前需要账号及权限的验证。

当前数据库服务器，默认只有一个超级管理员用户：root,PHPstudy安装后的默认密码为root.可以进行修改。

### SQL语句书写规范

eg. SELECT \* FROM `tb123` WHERE `name` = ”大头”;

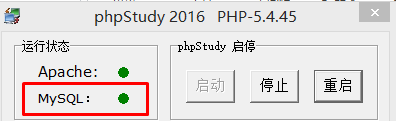
* 关键字尽量使用全大写,实际中不区分大小写;
* 数据库名、表名、字段名称使用反引号``包括
* 字符串信息使用单引号或者双引号包括

# MySQL数据库服务器访问

## 访问步骤

* 开启服务器

成功访问数据服务器的前提是：数据库服务器在运行状态。

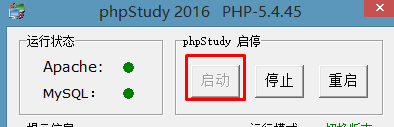


* 通过客户端发送命令(第一步为登录，建立起连接)

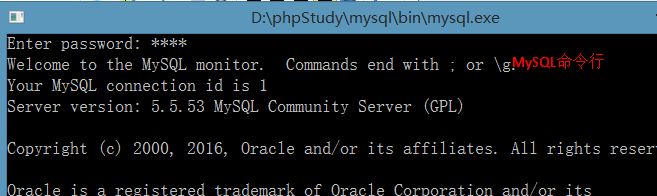
MySQL有权限验证（第二步），需要验证账号信息（用户名及密码）及用户的数据操作权限（是否具备查、删除等权限）。

## 开启数据库服务器

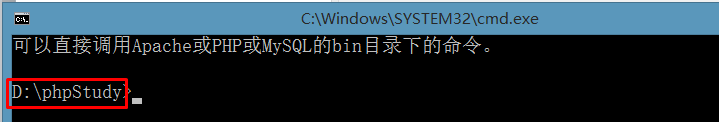
PHPstudy中可以通过启动按钮完成。



## 客户端连接数据库服务器



也可以通过**CMD命令行**：



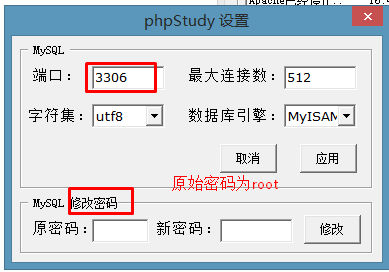
可以通过命令行，完成对数据库服务器的连接以及其他数据操作。也可以通过PHP，发送SQL命令操作数据库服务器中的数据。

### 命令行连接命令

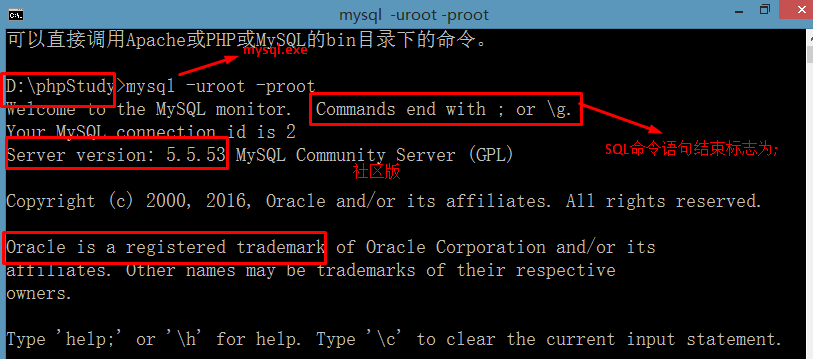
登录完成，建立起客户端与数据库服务器之间的连接。然后可以发送SQL命令。

mysql[.exe] [-h主机名] [–p端口号] –u用户名 –p密码

* mysql.exe数据库服务器登录验证主程序，命令行访问通常需要设置环境变量。
* 服务器主程序是：mysqld.exe
* h，host,主机名。主机地址 (域名或者IP地址)。数据库默认主机名为 localhost，可以省略。当前只有localhost本地主机。----127.0.0.1
* p（第一个p）,port，端口号。服务器监视该端口，提供数据库服务。数据库默认端口号 3306。可以省略。

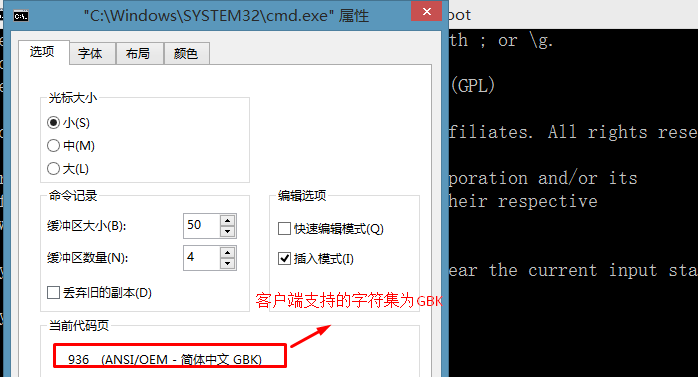


* u，user，用户名。当前数据库服务器只有root超级管理员账号，即用户名为root
* p（第二个p），password，密码。可以直接输入，也可以在回车之后输入，保护账号信息安全。
* 尚未登录，不是SQL语句，命令末尾没有分号
* 如果提示：cann’t connect to mysql server.原因是：服务器未启动



### 设置字符集

只针对命令行客户端来说。避免后续获取中文字符信息或者发送包含中文的命令时出现乱码。



你：中文 + 日文

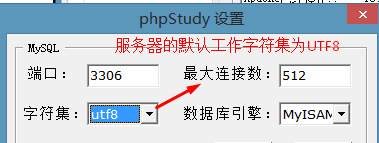
家里人：中文

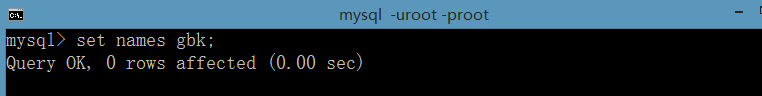
set names gbk;//GBK中文字符集

//通知数据库服务器在交互过程中使用的字符集为GBK，而不是UTF8

//数据库服务器中数据一般存储为UTF8，但是返回时数据库服务器可以转换为GBK编码

登录成功之后第一件事执行：set names gbk; 以避免后续操作出现数据的乱码。





### 退出

exit

