（string，int，float，bool）标量

常量只能是标量

常量只能在顶层代码中定义（函数，大括号外）；常量不能改变 不能销毁

常量的定义方式：

1：define("常量名"，常量值)

2：const 常量名=常量值

判断常量是否存在：defined()

与变量的区别

定义形式不同：

使用形式不同：常量无需$符号

可变程度不同：常量的值不可以改变，常量也不可以销毁

作用范围不同：常量具有超全局作用域 （函数内外都可以直接使用）

可用类型不同：常量只能存储标量类型：int，float，string，bool

php中的变量之间的值传递 默认是指传递

例如：$v1=1;$v2=$v1;

这里是新建一个内存地址存值再将值与变量之间建立关系

如果要进行关系拷贝

$v1=&$v2;

也就是js里面的深拷贝和浅拷贝

判断变量是否存在：isset()

1：**预定义变量又叫超全局变量**，包括：

$\_GET,---接受get请求的参数 包括链接跳转地址后面的查询字符串

$\_POST,--接受post请求的参数

$\_SERVER，--服务器和客户端的一些请求信息或设置信息（ip地址 主机名 querry）

$\_REQUEST,--接受的get和post请求的所有参数

$GLOBALS,--自定义的全部全局变量

$\_COOKIE,

$\_SESSION,

..........（大约10个）

它们全部都是数组

数据类型

标量类型： int， float， string， bool

复合类型： array， object

特殊类型： null， resouce

**整型**

decbin()--十进制转二进制 bindec()--二进制转十进制

decoct()--十进制转八进制 octdec()八进制转十进制

dechex()--十进制转16进制 hexdec()十六进制转十进制

**浮点型**

0.123

1.23E3

这两个都是浮点型--只要他表示的方法中出现了小数

建议不要直接用浮点型直接比较大小 因为在计算机内部 用二进制并不能精确的表示浮点型

**字符串型**

1：单引号：

表示纯粹的字符 在其中不存在空格 换行 制表以及$a(变量)

2：双引号

表示字符外 还可以表示一些换行 空格 制表 以及变量

**bool:**

0

0.0

null

“0”

array(); //空数组

未定义的变量 //当然要尽量避免

这些都判断为false

系统内置与变量相关函数

var\_dump()：能够输出一个变量的完整信息。

isset()判断一个变量是否存在数据：

empty()判断一个数据是否为空：接近我们的日常观念（没有就是空）

unset()：销毁（删除）一个变量

is\_int(),

is\_float(),

is\_numeric()，

is\_bool(),

is\_array(),

is\_scalar():判断是否为标量类型

**类型转换：**--一般发生在标量之间

自动转换：与js大致相同

强制转换：$a=(int)$b;

$a=(string)$b;

运算符：

与js不相同：%

7.5%3.5=>1 因为取余之前7.5和3.5都先要取整 7%3=>1

**数组**：

数组的定义

$arr=array(0=>"a","name"=>"bang");

1， $v1 = current( $arr1); //取得$arr1中当前指针所指向的元素的值，如果没有指向元素，则为false

2， $v1 = key（$arr1）； //取得$arr1中当前指针所指向的元素的下标，。。。。。false

3， $v1 = next($arr1 )； //将指针移向“下一个元素”，然后取得该下一个元素的值；

4， $v1 = prev($arr1)； //将指针移向“上一个元素”，然后取得该上一个元素的值

5， $v1 = reset($arr1）； //将指针移向“第一个元素”，然后取得该元素的值——数组指针初始化

6， $v1 = end($arr1）； //将指针移向“最后一个元素”，然后取得该元素的值

7， $v1 = each($arr1)； //取得当前元素的下标和值，然后移动指针到下一个位置。

数组的遍历：

foreach( $arr as $key => $value ）{

}

这当中$value是值传递 即改变$value的值 不会改变 数组中的原值

也可以把它改成引用传递

这时改变$value的值 也会改变 数组中的原值

**mysql:**

1:开启mysql服务 net start mysql

2.关闭mysql服务 net stop mysql

3 登陆mysql mysql (-h服务器地址) -uusername (-p端口号) -p

输入密码

4.退出登录 exit?quit

5.根据当前运行环境设置编码格式，set names gbk cmd环境默认为gbk php等语言中为当前语言环境编码（主流是utf-8）

显示所有数据库：show databases

创建数据库：creat database (if nir exists) db\_name (charset ..)

删除数据库：drop database （if exists）db\_name

查看数据库的字符集：show create batabase db\_name

修改数据库的字符集：alter database db\_name default character set ..

选择进入数据库：use db\_name

显示所有的数据表：show tables

创建表的结构（字段）：creat table table\_name(字段名1 字段类型 字段属性1 字段属性2，字段名2 字段类型 字段属性1 字段属性2，)

字段的类型主要有：tinyint、int、bigint、char、varchar、text、date、time

字段的主要属性有: NOT NULL | NULL：该列是否可以为空。

default\_value：默认值

AUTO\_INCREMENT:自动增长

PRIMARY KEY：主键

一般关键字id设置自动增长和主键

删除数据表：drop table (if exists) table\_name;

显示表的结构定义：describe table\_name

数据类型：



插入数据：INSERT INTO table\_name(字段1,字段2,字段3,…) VALUES(值1,值2,值3,…);

删除数据：delete from news where author = ‘admin’ and id<100;

//如果不写where则会删除整个表；

//初始化整个表：truncate [table] 表名

修改数据：UPDATE table\_name SET 字段1 = 新值1，字段2=新值2,… [WHERE条件]

查询数据：SELECT 字段列表|\* FROM table\_name [WHERE条件] [ORDER BY name DESC排序][LIMIT 20 10限制输出](从第20行fanhui 10条数据，并降序排列)

从表中查询一些数据 组成一个数据集返回

Select 字段1，字段2 from table;

返回字段1和字段2组成的数据集

Select 字段1 as 字段名，字段2 as 字段名 from table;

返回字段1和字段2的数据集并从新给他们字段别名

Select id\*10,concat(字段1，字段2，字段3) as xxx from table

可以给字段名（字段别名）和值做运算

Select table1.name,table2.age from table1,table2;

查询两个表，将表1和表2的两个字段合并成一个数据集

group By:分组

分组后的数据没有完整保存每行数据，只能获取字段的最大/小值/总和/平均值、分组后各组的长度

count(\*): 统计一组中的数量，通常用“\*”做参数

max(字段名)：获取该字段中在该组中的最大值。

min(字段名)：获取该字段中在该组中的最小值。

sum(字段名)：获取该字段中在该组中的总和。

avg(字段名)：获取该字段中在该组中的平均值。

Exap:

select name,count(\*) as length, max(Id) as id from table1 group by kind;

select name, kind ,sex,count(\*) as length, max(Id) as id from table1 group by kind,sex;//二级分组 根据两个字段分组数据

having:分组后条件筛选

exap:

select name,count(\*) as length, max(Id) as id from table1 group by kind having length>10;

连接查询：join

将两张表的字段横向的合并起来，

Selsect 字段1，字段2，t2.字段a,t2.字段b from table1 as t1 join table2 as t2 on t1.字段x=t2.字段y where 字段1>10 and 字段a<100;

从表一里面选择字段1和2，从表二里面选择字段a和b 如果表一的自选x和表二的自选y相等 那么就把它们横向的合并成四个字段的数据集

联合查询：union

将两张表纵向的合并起来；

select 语句1

union [ALL | DISTINCT]

select 语句2；

两个select语句的数据集 字段数目一样 字段类型一样 合并后的字段名以第一个为准

如果要对整个联合结果进行排序或limit，则应该对各自的select语句加括号：

子查询：（select ..）

一个select 语句中 又出现了一个select

子查询：

多行多列--表子查询//可以当作普通数据集使用

一行多列—行子查询

一列多行—列子查询//可以当作多个值使用类似这种：(5, 17, 8, 22)

比如可以把结果和where in 结合使用

where 操作数 比较运算符 any ( 列子查询 );//筛选操作数满足任意一个列子查询子元素

where 操作数 比较运算符 all( 列子查询 );// 筛选操作数满足所有列子查询子元素

一列一行—标量子查询//可以当作单个值使用

比如可以把值当查询关键字使用

可以把值当做where条件值使用



逻辑运算符： and（与） or（或） not（非）

Select \* from table1 where id>10 and name is not null;

模糊查找：

Like:%任意长度的任意字符

\_一个长度的任意字符

Exap:… where 字段1 like %xxxx%;

….where 字段2 like \_xxxx;

备份数据库

mysqldump -u.. -p 数据库名 > 数据库备份后文件地址 G:\php\copydatabases\1.sql

恢复备份 \*通常数据库要先建立

mysql -u.. -p 数据库名<要恢复的数据库的文件地址

php中操作数据库的流程

1.链接数据库

$mylink=mysql\_connect("localhost","root","root")

2.设定当前编码

mysql\_set\_charset("utf8");（执行函数）//也可以使用mysql\_query("set names utf8") (执行命令)

3.选择数据库

mysql\_select\_db("数据库名")；//mysql\_query("use 数据库名")

4.执行sql命令

**类：**

class className{

一般属性：每个对象可能不一样，调用和修改：对象->$v1

public $v1=1;(var $v1=1);

静态属性：每个对象的共有属性，调用和修改：className::$v1//或则$this::$v1

static $v1=2;

一般方法：

Function aaa(){

Return $this->$v1;

}

静态方法：静态方法中不能；使用$this;不能调用非静态方法

Static function(){

Echo ‘我是静态方法’;

}

构造方法：

Function \_construct($a,$b){

This->$v1=$a+$b;

}

}

访问修饰符：

Public：公有的 内外都可以访问

Protected：受保护的 类和子类可以访问

Private: 私有的 只有自己可以访问

类的继承:

Class A{

Public $a=”a”;

}

Class B extent A{

Public $b=”b”;

}

子类访问父类的静态属性：parent::$name;

ｇｅｔ＿ｃｌａｓｓ（对象） 获取一个对象的类名