一：redis简介：

redis的同类型产品（memcache、mongodb）

缓存数据库（非关系型数据库）:Remote Dictinoary server(内存告诉缓存数据库)

C语言编写，key-value

好处：支持丰富的数据结构：string、list、hash====、master/slave主从、数据的持久化

为什么使用：高速读取数据、减轻数据库负担、集合计算功能、多种数据结构支持

使用场景：

高并发场景、防攻击、热门话题

二、redis的安装、使用

2.1服务端的安装

windows：下载软件包

cd E:\wamp\redis =🡺 redis-server redis.conf

linux: 编译

2.2客服端在调用

2.2.1 直接通过redis-cli来访问redis服务端

cd E:\wamp\redis =🡺 redis-cli –h 127.0.0.1 –p 6379

set(设置key): SET key value [ex过期秒数] [px 过期毫秒数] nx xx

nx :只有键不存在时，才对键进行设置操作

xx：只有键已经存在时，才对键进行设置操作

get(读取数据)：GET key

2.2.2 phpredis

下载地址：http://pecl.php.net/package/redis

windows: 直接下载(php\_redis-3.1.2-7.0-ts-vc-14-x86)

$redis = **new** redis();  
$redis->connect( '127.0.0.1', 6379 );  
$redis->set( 'packphp', '666777' );  
//ctrl+左键 跟踪代码 ctrl+o 显示类的方法

$handsome = $redis->get( 'packphp' );  
var\_dump( '<pre>', $handsome );**exit**;

linux: phpize编译安装

三、redis的数据结构

3.1、存储方式

Keys：给存储在redis内存中的数据起的变量名字

命名规范：空格、\n 不能使用

取名：简短、有意义（sixstar:teacher:hansome）

values:

Strings Lists Sets Sorted sets Hash

3.2、string类型

存储任何数据：通过序列化来存储，serialize unserialize 序列化 反序列化

持久化的memcache

操作方法：

set( ‘key’, ‘value’) 设置值 get( ‘key’ )

mset( $array ) 一维数组（一次设置多个元素的值）

mget( $array ) 一次获取多个值

incr 对key做自增，返回新的值

decr 对key做自减，返回新的值

incrby（’key’, value） 加指定值 $redis-> incrby( ‘pack’, 100 );

decrby() 减指定值

append() 给指定key的字符串值追加value

substr( $key, $start, $end ) 字符串截取

3.3、List

list类型：双向链表。通过push,pop操作从链表的头部或者尾部添加删除元素

list既可以用作栈，也可以用做队列

lrange( key,start,end); 读取指定区间的元素，从0开始索引

lpush( key, val1, val2,…valn); 在key对应的头部添加字符串元素

rpush(key, val1, val2,…valn )； 在key对应的尾部添加

$redis->lpush( 'packpython1', 'shell', 'linux', 'oracle');  
$redis->rpush( 'packpython1', 'mui','mui1','mui2');

lpop( ‘key’) 从list的头部删除元素， 并返回被删除的元素

rpop(‘key’) 从list的尾部删除元素， 并返回被删除的元素

llen() 返回list长度

ltrim( key, start , end ); 截取list,保留指定区间元素

3.4 Set(Set) string类型的无序集合(2的32次方-1)

操作方法：添加、删除、并集、交集、差集

sAdd( key, value1,value2,….valuen)添加一个string元素到key对应的set集合中

sMembers( key ); 返回key对应set的所有元素，结果是无序的

sRem( key, val1 ) 移除指定元素（value）

scard( key ); 返回set元素的个数

sismember( key, val ) 判读key的集合中是否存在元素val

sinter( key1, key2 ) 返回所有给定的set(集合)的交集

sunion( key1, key2 ) 返回所有给定的set(集合)的并集

sdiff( key1, key2 ) 返回所有给定的set(集合)的差集

3.5 Sort Set string类型的元素的集合

每一个元素都会关联一个权，通过权值可以有序的获取集合中的元素

zAdd( key, score, value1, score2, value2 ) 添加元素到集合，如果元素在集合中存在更新对应score

zCard( key ); 返回集合中元素个数

zScore( key, value ); 返回给定元素的score

zRem( key, val ); 删除指定元素（元素值）

zRemrangebyrank( key, score1, score2 ); 删除集合中排名（权重）在给定区间的元素

zRange( key, start, end ) 从集合中返回指定区间（索引），返回按自增排序的数组

zRangebyscore(key, start, end ) 从集合中返回指定区间（权重值），返回按自增排序的数组

zRevrange() 同上，降序排序

zRevrangebyscore() 同上，降序排序

zIncrby( key, ++score, val ) 增加某个val的权重值

3.6、hash(h)

hSet( ‘key’, ‘hashkey1’, val1 ) 设置key的某个hashkey1为val1

hGet( ‘key’, ‘hashkey1’ ) 读取key下hashkey1的值

hMset( ‘key’, ‘arr’) 设置key下多个hashkey的值（通过arr的key(hashkey)对应到val(hashval)）

hMget( ‘key’, ‘arr’ ) 读取key下多个hashkey的值（arr的value值为hashkey）

hDel( ‘key’, ‘hashkey1’,’hashkey2=null’,’hashkey3=null’) 删除key下指定hashkey1的值

hIncrBy( ‘key’, ‘hashkey1’, int number) 给key下指定的hashkey1增加整型数值

hIncrByFloat( ‘key’, ‘hashkey1’,float number) 给key下指定的hashkey1增加金额

hLen( ‘key’); 获得key下的field个数

hGetall( ‘key’); 返回指定key的数据结果（返回所有的field和value）

hKeys( ‘key’); 返回指定key下的所有field

hVals(‘key’); 返回指定key下的所有value