# 2019 年高级 PHP 面试题

## Beta 2.0

离线的面试题总会有过期的那一天，如想获取最新的面试题和答案，请登录跬步客学习平台, 查看最新面试题的网址: https://www.kuibuke.com/exam



 

## 1、给你四个坐标点，判断它们能不能组成一个矩形，如判断([0,0],[0,1],[1,1],[1,0])能组成一个矩形。

勾股定理，矩形是对角线相等的四边形。只要任意三点不在一条直线上，任选一点，求这一点到另外三点的长度的平方,两个短的之和如果等于最长的，那么这就是矩形。

## 2、写一段代码判断单向链表中有没有形成环，如果形成环，请找出环的入口处，即 P 点

```
/*
 *单链表的结点类
 */
class LNode{
    //为了简化访问单链表,结点中的数据项的访问权限都设为 public
    public int data;
    public LNode next;
}


class LinkListUtli {
    //当单链表中没有环时返回 null，有环时返回环的入口结点
    public static LNode searchEntranceNode(LNode L)
    {
        LNode slow=L;//p 表示从头结点开始每次往后走一步的指针
        LNode fast=L;//q 表示从头结点开始每次往后走两步的指针
        while(fast !=null && fast.next !=null)
        {
            if(slow==fast) break;//p 与 q 相等，单链表有环
            slow=slow.next;
            fast=fast.next.next;
        }
        if(fast==null || fast.next==null) return null;

        // 重新遍历，寻找环的入口点
        slow=L;
        while(slow!=fast)
        {
            slow=slow.next;
            fast=fast.next;
        }

        return slow;
    }
}
```

## 3、写一个函数，获取一篇文章内容中的全部图片，并下载

```
function download_images($article_url = '', $image_path = 'tmp'){
```



 

```
    // 获取文章类容
    $content = file_get_contents($article_url);

    // 利用正则表达式得到图片链接
    $reg_tag = '/<img.*?\"([^\"]*(jpg|bmp|jpeg|gif|png)).*?>/';
    $ret = preg_match_all($reg_tag, $content, $match_result);
    $pic_url_array = array_unique($match_result1[1]);

    // 创建路径
    $dir = getcwd() . DIRECTORY_SEPARATOR .$image_path;
    mkdir(iconv("UTF-8", "GBK", $dir), 0777, true);


    foreach($pic_url_array as $pic_url){
        // 获取文件信息
        $ch = curl_init($pic_url);
        curl_setopt($ch, CURLOPT_HEADER, 0);
        curl_setopt($ch, CURLOPT_NOBODY, 0);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE );
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE );
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $fileInfo = curl_exec($ch);
        $httpinfo = curl_getinfo($ch);
        curl_close($ch);

        // 获取图片文件后缀
        $ext = strrchr($pic_url, '.');
        $filename = $dir . '/' . uniqid() . $ext;

        // 保存图片信息到文件
        $local_file = fopen($filename, 'w');
        if(false !== $local_file){
            if( false !== fwrite($local_file, $filecontent) ){
            fclose($local_file);
            }
        }
    }

}
```

## 4、获取当前客户端的 IP 地址，并判断是否在（111.111.111.111,222.222.222.222)

如果没有使用代理服务器：

$ip = $_SERVER['REMOTE_ADDR'];
使用透明代理

$ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
参考文章

https://www.cnblogs.com/rendd/p/6183094.html

## 5、nginx 的 log_format 配置如下：

```
log_format main ‘remoteaddr-remote_user [timelocal]"request”’
‘statusbody_bytes_sent “httpreferer""http_user_agent”
“upstreamresponsetime""request_time” “http_x_forwarded_for"';
```

从今天的 nginx log 文件 access.log 中：

- a、列出“request_time”最大的 20 行？
- b、列出早上 10 点访问量做多的 20 个 url 地址？

## 6、什么是 CSRF 攻击？XSS 攻击？如何防范？

CSRF：跨站请求伪造，可以通过通过判断来源和加 Token 的方式来防范。

XSS：跨站脚本攻击，可以通过对内容转义和过滤来防范,还有 CSP

## 7、应用中我们经常会遇到在 user 表随机调取 10 条数据来展示的情况，简述你如何实现该功能。

```
SELECT * FROM `table` WHERE id >= (SELECT FLOOR( MAX(id) * RAND()) FROM
`table` ) ORDER BY id LIMIT 1;
```

参考文章：

https://www.cnblogs.com/riasky/p/3367558.html

http://www.jb51.net/article/48801.htm

## 8、从扑克牌中随机抽 5 张牌，判断是不是一个顺子，即这 5 张牌是连续的

这个问题有个关键点，扑克牌，1-13 不能再多了。这就很简单了。用 PHP 来做，定义一个数组分别存着 1 到 13,拿出一个，置空一个，最后看下 这五个置空的 是不是连续的。这种情况不考虑抽出的顺序。

## 9、两条相交的单向链表，如何求它们的第一个公共节点

思想：



 

1. 如果两个链表相交，则从相交点开始，后面的节点都相同，即最后一个节点肯定相同；
2. 从头到尾遍历两个链表，并记录链表长度，当二者的尾节点不同，则二者肯定不相交；
3. 尾节点相同，如果 A 长为 LA，B 为 LB，如果 LA>LB,则 A 前 LA-LB 个先跳过

如果两个单向链表有公共的结点，也就是说两个链表从某一结点开始，它们的 m_pNext 都指向同一个结点。但由于是单向链表的结点，每个结点只有一个 m_pNext，因此从第一个公共结点开始，之后它们所有结点都是重合的，不可能再出现分叉。所以，两个有公共结点而部分重合的链表，拓扑形状看起来像一个 Y，而不可能像 X。
参考文献：

https://blog.csdn.net/wcyoot/article/details/6426436

https://blog.csdn.net/Lieacui/article/details/52046548

## 10、最长公共子序列问题 LCS，如有[1,2,5,11,32,15,77]和[99,32,15,5,1,77]两个数组，找到它们共同都拥有的数，写出时间复杂度最优的代码，不能用 array_intersect（这里有坑，需要去研究一下动态规划）。

## 11、linux 的内存分配和多线程原理

## 12、MYSQL 中主键与唯一索引的区别

主键：绝对不能有空值。唯一索引：可以有空值

参考：https://www.cnblogs.com/lonelyxmas/p/4594624.html

## 13、http 与 https 的主要区别

关键是 S 上。简而言之，https 建立连接后要先把 SSL 的证书发下去，有了公钥和私钥，就可以解密了。

参考：https://www.cnblogs.com/zyl-Tara/p/7079696.html



 

## 14、http 状态码及其含意

- 200 请求已成功，请求所希望的响应头或数据体将随此响应返回。
- 301 被请求的资源已永久移动到新位置。
- 302 请求的资源现在临时从不同的 URI 响应请求。
- 400 1、语义有误，当前请求无法被服务器理解。2、请求参数有误。
- 401 当前请求需要用户验证。
- 403 服务器已经理解请求，但是拒绝执行它。
- 404 请求失败，请求所希望得到的资源未被在服务器上发现。
- 500 服务器遇到了一个未曾预料的状况，无法完成对请求的处理，会在程序码出错时出现。
- 501 服务器不支持当前请求所需要的某个功能。无法识别请求的方法。
- 502 作为网关或者代理工作的服务器尝试执行请求时，从上游服务器接收到无效的响应。
- 503 由于临时的服务器维护或者过载，服务器当前无法处理请求。

参考：http://tool.oschina.net/commons?type=5

## 15、linux 中怎么查看系统资源占用情况

top、htop、free、uptime

## 16、SQL 注入的原理是什么？如何防止 SQL 注入

原理：第一 SQL 本身有问题（这个不是主要问题）。第二你写的 SQL 很有问题（这是最主要的）

防范：第一，绝对不要相信用户输入的任何东西。第二，预编译。现在的框架一般都会有 SQL 过滤的。

## 17、isset(null) isset(false) empty(null) empty(false)输出

PHP 入门问题,isset 和 empty 的区别

分别是 false, true, true, true



 

## 18、优化 MYSQL 的方法

第一，数据超过一定数量或者体积，请拆分表，垂直或者水平分（最有效果的优化）

第二，务必有自增主键。通过自增主键来查数据是最快的。

第三，常用的查询字段建立联合索引，写 SQL 一定要尊从最左原则，用到这个索引。

第四，不要把逻辑运算放到 sql 里。言外之意是，不要写太复杂的 SQL，你能写复杂的 SQL 你肯定也能通过 PHP 实现。

参考：https://cloud.tencent.com/developer/article/1004367

## 19、数据库中的事务是什么？

事务（transaction）是作为一个单元的一组有序的数据库操作。如果组中的所有操作都成功，则认为事务成功，即使只有一个操作失败，事务也不成功。如果所有操作完成，

事务则提交，其修改将作用于所有其他数据库进程。如果一个操作失败，则事务将回滚，该事务所有操作的影响都将取消。

## 20、写一个函数，尽可能高效的从一个标准 URL 中取出文件的扩展名

会 写正则最好。我反正不会正则，需要用的时候就百度。

## 21、参数为多个日期时间的数组，返回离当前时间最近的那个时间

遍历数组，求当前时间差，和第一个进行对比，小于第一个交换位置。最后取第一个即可。

## 22、echo、print、print_r 的区别

这三个放在一起，回答的点在于，print_r 是函数，echo、print 是结构语言。

至于他们具体的区别参考：https://www.cnblogs.com/xiaotaoing/p/6687368.html

## 23、http 协议的 header 中有哪些 key 及含义

这个问题，很难。一会半会说不清楚。如果做过 PHP restful 接口开发，也踩过这里面的坑，那应该是可以答出来常用的几个 KEY 的。

参考：https://blog.csdn.net/u014175572/article/details/54861813

## 24、二叉树前中后遍历代码

1.层序遍历 2.先序遍历 3.中序遍历 4.后序遍历

参考：https://blog.csdn.net/wk199351/article/details/65936001

## 25、PHP 的数组和 C 语言的数组结构上有何区别？

但从 PHP 来讲，考的是 PHP 数组的实现。可以简单的认为，PHP 的数组是 hash 桶+十字链表（实际上是数列 Array，列表 List，散列表/关联数组/字典 Hashtable 的聚合体）。优点是查询效率很高，遍历很方便，缺点是，占内存较多。（还是空间换时间的思路，毕竟现在内存又不值钱）

C 语言的数组，就是定长定类型的数列。

## 26、Redis 的跳跃表怎么实现的

跳跃表(skiplist)是一种有序数据结构，它通过在每个节点中维持多个指向其他节点的指针，从而达到快速访问节点的目的。

参考：https://blog.csdn.net/universe_ant/article/details/51134020



 

**27、哈希是什么？hash 冲突后，数据怎么存？**

**28、聚簇索引，聚集索引的区别？**

**29、B+Tree 是怎么进行搜索的**

**30、数组和 hash 的区别是什么？**

**31、写个函数，判断下面扩号是否闭合，左右对称即为闭合：**
**((()))，)(())，(()))，((((()))，(()())，()()**

**32、 找出数组中不重复的值[1,2,3,3,2,1,5]**

用普通方法，肯定很容易的。

**33、32 题你的时间复杂度是多少？有的情况下，你写了个算法，然后面试官会让你把你的算法的时间复杂度表达式写出来**

**34、PHP 的的这种弱类型变量是怎么实现的？**

考 zval 的，PHP 的八种类型，本质只有一个结构。

参考：https://blog.csdn.net/ohmygirl/article/details/41542445

**35、在 HTTP 通讯过程中，是客户端还是服务端主动断开连接？**

三次握手和四次挥手，以及他们每步的状态。

这个问题最好能一步到位回答的全面的。一般都是有客户端告诉服务端，我这边东西发完了，可以断连接了么。但是如果客户端发完 FIN 服务端没有回复，就会重试，直到超过超时时间，就断了。服务端也一样，超过时间，服务端就断了。

**36、PHP 中发起 http 请求有哪几种方式？它们有何区别？**

1. GET
2. POST
3. HEAD

4. PUT
5. DELETE
6. OPTIONS
7. TRACE
8. CONNECT

**37、有一颗二叉树，写代码找出来从根节点到 flag 节点的最短路径并打印出来，flag 节点有多个。比如下图这个树中的 6 和 14 是 flag 节点，请写代码打印 8、3、6 和 8、10、14 两个路径**

典型的二叉搜索树。大学数据结构的基础题。

参考：https://blog.csdn.net/BaiHuaXiu123/article/details/52488443

**38、有两个文件文件，大小都超过了 1G，一行一条数据，每行数据不超过 500 字节，两文件中有一部分内容是完全相同的，请写代码找到相同的行，并写到新文件中。PHP 最大允许内内为 255M。**

将文件拆分成若干个小文件，根据内容计算 hash 值，分散到不同文件。

**39、请写出自少两个支持回调处理的 PHP 函数，并自己实现一个支持回调的 PHP 函数**

array_map,array_filter, array_walk

**40、请写出自少两个获取指定文件夹下所有文件的方法（代码或思路）。**

核心方法是 scandir,核心思想是递归。

**41、请写出自少三种截取文件名后缀的方法或函数（PHP 原生函数和自己实现函数均可）**

```php
echo substr(strrchr($file, '.'), 1);

echo substr($file, strrpos($file, '.')+1);

$arr=explode('.', $file);
echo $arr[count($arr)-1];
```



 

```php
$arr=explode('.', $file);
echo end($arr);

echo strrev(explode('.', strrev($file))[0]);

echo pathinfo($file)['extension'];

echo pathinfo($file, PATHINFO_EXTENSION);
```

## 42、PHP 如何实现不用自带的 cookie 函数为客户端下发 cookie。对于分布式系统，如何来保存 session 值。

这个题有点绕。考的还是 COOKIE 和 SESSION 的基础知识。服务端通过 set-cookie 命令来通知客户端保存 cookie。

只要按照 domain path 过期时间等规则 用 header 函数就可以实现。

分布式系统 session，集中处理。按我们公司的架构，为了实现高可用和高容灾，提供一个分布式的验签服务。具体的可以看下 redis 的分布式服务架构。

## 43、请用 SHELL 统计 5 分钟内，nginx 日志里访问最多的 URL 地址，对应的 IP 是哪些？

## 44、写一段 shell 脚本实现备份 mysql 指定库（如 test)到指定文件夹并打包，并删除 30 天前的备份，然后将新的备份推送到远端服务器，完成后送邮件通知。

## 45、mysql 数据库中 innodb 和 myisam 引擎的区别

区别主要在数据和索引的存储结构和存储方式上，以及对于事务的支持。

参考：https://blog.csdn.net/chajinglong/article/details/56666771

## 46、从用户在浏览器中输入网址并回车，到看到完整的见面，中间都经历了哪些过程。

入门问题。这个问题有一个很大的坑，面试官可能会从这个问题下手问你一大堆问题。

以 PHP 为例：通常最简单的回答，从用户的电脑找到最近的 DNS 服务，然后解析到对应的 IP 然后双方开始 HTTP 连接，然后发送请求信息，服务器拿到请求信息



 

就开始准备回应的信息，中间要经过 nginx 转发到 frstCGI(PHP-FPM),然后 PHP 开始解析框架，解析请求头部，找到对应的 API，该查数据库查数据，该组装 HTML 组装 HTML，完事了就重新返回给用户。用户拿到返回数据，浏览器开始渲染页面，JS 开始加载。

## 47、如何分析一条 sql 语句的性能。

explain，具体的请百度。（基本很少用性能分析语句。MYSQL 的表设计上尽量冗余一部分字段，避免在 MYSQL 里处理大量的逻辑运算。我们是做 PHP 服务开发的，mysql 语句能简单尽量简单。逻辑运算的地方可以在 PHP 里做。）

## 48、ping 一个服务器 ping 不通，用哪个命令跟踪路由包？

```
linux:traceroute,windows:tracert
```

## 49、$a=[0,1,2,3]; $b=[1,2,3,4,5]; $a+=$b; var_dump($a)等于多少？

基础问题。本质还是考 PHP 数组的结构和特点。

结果是 01235。PHP 用数字索引和 STRING 索引差别还是很大的

参考：http://www.jb51.net/article/38593.htm

## 50、$a=[1,2,3]; foreach($a as &$v){} foreach($a as $v){} var_dump($a)等于多少;

122 此处有一坑。foreach 完之后，$index , $value 并不会消失保留最后一次赋值。 这里的第一次 foreach 之后，数组中最后一个元素变成引用，引用变量 $v 继续存在且指向数组的最后一个元素。第二次遍历，因为遍历变量名是 $v , 所以等于说每次遍历都将此次遍历的值修改成最后元素的值，直至到遍历最后一个元素（引用元素），因为此时数组的最后一个元素已被修改成上一个元素的值，最后一次赋值就是 自己==自己。 故最后一个等于倒数第二个

https://laravel-china.org/articles/7001/php-ray-foreach-and-references-thunder

## 51、数据库中的存放了用户 ID,扣费很多行，redis 中存放的是用户的钱包，现在要写一个脚本，将数据库中的扣费记录同步到 redis 中，每 5 分钟执行一次。请问要考虑哪些问题？

思路：生产者和消费者模式。这个问题也没有说其他的状态，比如数据库的数据会实时增加么？redis 中每个钱包是否有其他服务在读取或者写入啊。什么的。数据库和 REDIS 放一起，要么考数据一致性，要么考出现锁，导致效率降低。

## 52、MYSQL 主从服务器，如果主服务器是 innodb 引擎,从服务器是 myisam 引擎，在实际应用中，会遇到什么问题？

不知道，没用过，为什么这么设计？故意给自己找不愉快？

## 53、linux 中进程信号有哪些？

kill -l 很少用

## 54、redis 的底层实现

面试官这么样问你，你就反问他，你要的底层实现是字段的设计？内存分配管理？网络模型？数据同步？还是分布式的实现？（TIPS:面试就是两个人的博弈。面试官给出一个描述不清晰的问题，我们没必要回答。让他把问题讲清楚再思考怎么回复）

参考：https://cloud.tencent.com/developer/article/1004377

这篇文章 要多读几遍。

## 55、异步模型

问清楚是 IO 异步模型。还是 AJAX 这类的异步请求模型。差别非常大的。

参考：https://cloud.tencent.com/developer/article/1005481

狗东某风控研发必考题。

## 56、10g 文件，用 php 查看它的行数

粗暴一点的方法 ini_set('memory_limit','-1'); 先把当前内存限制解除了 然后直接逐行统计。时间会非常的久。

有更好的方法请留言。

## 57、有 10 亿条订单数据，属于 1000 个司机的，请取出订单量前 20 的司机

（TIPS）不要中招。不要用常用思路来处理，10 亿数据 你再怎么优化，全表求和，都是要死人的。

我们从设计上解决这个问题。只有一千个司机。我们可以做个简单哈希，分库分表，%求余数。保证这一千个司机分在一千个表里，每个人有每个人的单独表。引擎用 MYSAIM，求表中数据的总数，效率飞快，遍历一千张表，求最大前二十即可。

## 58、设计一个微信红包的功能

没做过。其实题目表达不清楚。如果做过微信公众号开发，知道微信事件模型的 XML 数据结构，应该会好做一点。

## 59、根据 access.log 文件统计最近 5 秒的 qps，并以如下格式显示，01 1000（难点在 01 序号）

```
tail -f access.log | awk -F '[' '{print $2}' | awk '{print $1}' | uniq -c
```

参考：https://blog.csdn.net/dong_007_007/article/details/78330337

## 60、php7 性能为什么提升这么高

不逼逼，直接参考：http://www.laruence.com/php-internal

鸟哥的文章要多读，多读。

## 61、遍历一个多维数组

递归。array_map 传入一个回调函数。

## 62、有这样一个字符串 abcdefgkbcdefab……随机长度，写一个函数来求 bcde 在这个字符串中出现的次数

substr_count（）；

## 63、有一个 1G 大小的一个文件，里面每一行是一个词，词的大小不超过 16 个字节，内存限制大小是 1M。返回频数最高的 100 个词

方法太多了，但是实现起来 各有各的问题。

我可能只会用 HASH 映射做。其他的，不会。

参考：第 64 题。

## 64、十道海量数据处理面试题与十个方法大总结

```
> https://blog.csdn.net/v_JULY_v/article/details/6279498
```

## 65、php 进程模型，php 怎么支持多个并发

守护进程模型（需要知道 php-fpm 的各种配置了）

参考：https://www.jianshu.com/p/542935a3bfa8

## 66、nginx 的进程模型，怎么支持多个并发

这个三言两语说不清楚。

参考：https://www.zhihu.com/question/22062795

## 67、php-fpm 各配置含义，fpm 的 daemonize 模式

php-fpm 的配置并不多，常用的就更少了。

参考：http://www.4wei.cn/archives/1002061

```
static - 子进程的数量是固定的（pm.max_children）
ondemand - 进程在有需求时才产生（当请求时，与 dynamic 相反，pm.start_servers 在服务启动时即启动
dynamic - 子进程的数量在下面配置的基础上动态设置：pm.max_children，
pm.start_servers，pm.min_spare_servers，pm.max_spare_servers
```

## 68、让你实现一个简单的架构，并保持高可用，两个接口，一个上传一条文本，一个获取上传的内容，你怎么来设计？要避免单机房故障，同时要让代码层面无感。

参考：分布式架构设计必备 CAP 原理。



 

## 69、两台 mysql 服务器，其中一台挂了，怎么让业务端无感切换，并保证正常情况下讲台服务器的数据是一致的

不是核心业务的话，先停写，把备机拉起来，查看两台机器的日志，进行数据补偿，开写。

如果是核心业务的话，现在所有的写操作都在正常的状态机器上。把好的这台机器的备机拉起来，当主机。

以上全是应急操作。实际上数据库的容灾设计要复杂的多。

面试官要是问你，备机的数据不一致怎么办，你要勇敢怼回去，你们每秒多少写入操作。按照百万级表，每秒 1000 的写入效率，正常的设计是，分布在 2 台机器上每台 500。这个级别的数据同步，出现差异的概率 可以忽略不计的。有一台出现问题，另一台也可以抗住。

（正常的操作，还是先停写，等数据一致，切换，开写。我们公司搞这些切换都是在凌晨 4.00 左右，核心业务的每秒写操作，只有十几个。前后耽搁不到 20 秒）。

## 70、http 协议具体的定义

这种题 有是很难回答的。太宽泛了，我们面试早就不问这种问题了。

参考：日本人写的《图解 HTTP》

## 71、什么是锁，怎么解决锁的问题

计算机原理学的，生产者消费者模型，银行家模型，都可以解决锁的问题。

## 72、rand 与 mt_rand 的区别

我实习的时候遇到这个坑。

说是 mt_rand 比 rand 快 4 倍。

在随机数区间不大的情况下并没有很大的效率差距。但是出现重复数的几率，rand 要比 mt_rand 高很多。

## 73、mysql 事务隔离是怎么实现的

通过各种行锁表锁，各种乐观锁悲观锁，排他锁实现的呀。

## 74、mysql 的锁怎么实现的

https://blog.csdn.net/alexdamiao/article/details/52049993

https://www.cnblogs.com/luyucheng/p/6297752.html

https://blog.csdn.net/tangkund3218/article/details/47704527

## 75、对称加密和非对称加密的方式

对称加密：我们俩共用一个秘钥，你加密，我解密。

非对称加密：我给你一个公钥，你加密完了，我还能有我的私钥把密文解开。但是你没有我的私钥。

扩展：椭圆加密算法。

## 76、10 瓶水，其中一瓶有毒，小白鼠喝完有毒的水之后,会在 24 小时后死亡,问:最少用几只小白鼠可以在 24 小时后找到具体是哪一瓶水有毒。

四只

二进制问题。薛定谔的老鼠。

一只老鼠有两个状态，死活，对应 01。假设老鼠的个数为 A，则有 2^A>=10; A=4；

思路很简单，十瓶药编号：0,1,10,11....1001;

0 不喝。第一只老鼠喝所有个位是 1 的：13579，第二只喝十位是 1 的，第三只和百位是 1 的，第四只喝千位是 1 的。

24 小时后，看下死了的是 1，活着的是 0。按老鼠的顺序乖乖站好.....假如第一只和第三只死了，那就是 0101，就是 5 有问题。

## 77、redis 是如何进行同步的，同步的方式，同步回滚怎么办，数据异常怎么办，同时会问 MYSQL 的同步方式和相关异常情况

redis 集群主从同步的简单原理

Redis 的复制功能是基于内存快照的持久化策略基础上的，也就是说无论你的持久化策略选择的是什么，只要用到了 Redis 的复制功能，就一定会有内存快照发生。

当 Slave 启动并连接到 Master 之后，它将主动发送一个 SYNC 命令( 首先 Master 会启动一个后台进程，将数据快照保存到文件中[rdb 文件] Master 会给 Slave 发送一个

Ping 命令来判断 Slave 的存活状态 当存活时 Master 会将数据文件发送给 Slave 并将所有写命令发送到 Slave )。

Slave 首先会将数据文件保存到本地 之后再将 数据 加载到内存中。

当第一次链接 或者是 故障后 重新连接 都会先判断 Slave 的存活状态 在做全部数据的同步， 之后只会同步 Master 的写操作(将命令发送给 Slave)

问题：

当 Master 同步数据时 若数据量较大 而 Master 本身只会启用一个后台进程 来对多个 Slave 进行同步， 这样 Master 就会压力过大， 而且 Slave 恢复的时间也会很慢！

redis 主从复制的优点：

(1)在一个 Redis 集群中，master 负责写请求，slave 负责读请求，这么做一方面通过将读请求分散到其他机器从而大大减少了 master 服务器的压力，另一方面 slave 专注于提供读服务从而提高了响应和读取速度。

(2)在一个 Redis 集群中，如果 master 宕机，slave 可以介入并取代 master 的位置，因此对于整个 Redis 服务来说不至于提供不了服务，这样使得整个 Redis 服务足够安全。

(3)水平增加 Slave 机器可以提高性能

参考：

- https://blog.csdn.net/hxpjava1/article/details/78347890/
- https://www.cnblogs.com/zhao-blog/p/6131524.html

## 78、怎么解决跨域

- JSONP



 

- 添加响应头，允许跨域
- 代理的方式

## 79、json 和 xml 区别,各有什么优缺点

（1）可读性方面：基本相同，XML 的可读性比较好；

（2）可扩展性方面：都具有良好的扩展性；

（3）编码难度方面：相对而言，JSON 的编码比较容易；

（4）解码难度：JSON 的解码难度基本为零，XML 需要考虑子节点和父节点；

（5）数据体积方面：JSON 相对于 XML 来讲，数据体积小，传递的速度比较快；

（6）数据交互方面：JSON 与 javascript 的交互更加方便，更容易解析处理，更好的数据交互；

（7）数据描述方面：XML 对数据描述性比较好；

（8）传输速度方面：JSON 的速度远远快于 XML。

参考：https://blog.csdn.net/java19880223/article/details/20054111

## 80、Trait 优先级

在 trait 继承中，优先顺序依次是：来自当前类的成员覆盖了 trait 的方法，而 trait 则覆盖了被继承的方法

## 81、a 引用 b，报错 c 里面类重复定义，循环引用会出现什么问题

## 82、下面员工 3 的薪水大于其主管的薪水，一条 SQL 找到薪水比下属低的主管

| id | username | salary |
|---|---|---|
| 1 | a | 3000 |
| 2 | b | 8000 |



 

| id | username | salary |
| --- | --- | --- |
| 3 | c | 5000 |
| 4 | d | 6000 |

```
SELECT a.*, b.*
FROM `user` as a
LEFT JOIN `user` as b ON a.pid = b.id AND a.salary > b.salary
WHERE b.id > 0;
```

## 82、在一个坐标系内有一个 N 个点组成的多边形,现在有一个坐标点,写代码或思路来判断这个点是否处于多边形内

## 83、数据库如果出现了死锁,你怎么排查,怎么判断出现了死锁?

https://www.cnblogs.com/huanyou/p/5775965.html

## 84、写一个一个程序来查找最长子串

http://www.jb51.net/article/128449.htm

## 85、分析一个问题:php-fpm 的日志正常,但客户端却超时了,你认为可能是哪里出了问题,怎么排查?

检查 nginx log，请求是否达到 nginx 和是否正常转发给 php-fpm

## 86、nginx 的工作流程是什么样的,可以画图描述

## 87、进程间通信方式有哪些

1)管道 管道分为有名管道和无名管道 无名管道是一种半双工的通信方式,数据只能单向流动,而且只能在具有亲缘关系的进程间使用.进程的亲缘关系一般指的是父子关系。无明管道一般用于两个不同进程之间的通信。当一个进程创建了一个管道,并调用 fork 创建自己的一个子进程后,父进程关闭读管道端,子进程关闭写管道端,这样提供了两个进程之间数据流动的一种方式。 有名管道也是一种半双工的通信方式,但是它允许无亲缘关系进程间的通信。



 

2)信号量 信号量是一个计数器,可以用来控制多个线程对共享资源的访问.,它不是用于交换大批数据,而用于多线程之间的同步.它常作为一种锁机制,防止某进程在访问资源时其它进程也访问该资源.因此,主要作为进程间以及同一个进程内不同线程之间的同步手段.

3)信号 信号是一种比较复杂的通信方式,用于通知接收进程某个事件已经发生.

4)消息队列 消息队列是消息的链表,存放在内核中并由消息队列标识符标识.消息队列克服了信号传递信息少,管道只能承载无格式字节流以及缓冲区大小受限等特点.消息队列是 UNIX 下不同进程之间可实现共享资源的一种机制,UNIX 允许不同进程将格式化的数据流以消息队列形式发送给任意进程.对消息队列具有操作权限的进程都可以使用 msget 完成对消息队列的操作控制.通过使用消息类型,进程可以按任何顺序读信息,或为消息安排优先级顺序.

5)共享内存 共享内存就是映射一段能被其他进程所访问的内存,这段共享内存由一个进程创建,但多个进程都可以访问.共享内存是最快的 IPC(进程间通信)方式,它是针对其它进程间通信方式运行效率低而专门设计的.它往往与其他通信机制,如信号量,配合使用,来实现进程间的同步与通信.

6)套接字：可用于不同及其间的进程通信

## 88、主从复制，从服务器会读取到主服务器正在回滚的数据吗？主数据库写成功，从服务器因为一些原因写失败，最后会出现什么情况？主从复制如果键冲突怎么办？

不会；主从数据不一致；正常是不会出现这种情景，具体看情况，是否可以修复，恢复到之前的时间点，然后追回同步。

## 89、事务有几种隔离级别？事务的隔离级别是怎么实现的?

- 读未提交（read-uncommitted）
- 不可重复读（read-committed）
- 可重复读（repeatable-read）
- 串行化（serializable）

https://www.cnblogs.com/huanongying/p/7021555.html

**90、什么是 B+树,请画 b+树的结构**

https://blog.csdn.net/qq_23217629/article/details/52510485

**91、mysql 中的字符集，客户端与数据库不一致，怎么办? MYSQL 中字符串到显示到界面,字符转换的过程是怎样的？数据库中的字符集是 latin1,你现在将 utf8 的字符串存到 latin1 字符集的数据库表,你能将 utf8 的字符串存进去吗？假如你说能存,追问:能否恢复?假如能,那怎么恢复?**

**92、写一段代码，找到所有子集合，如[a,b,c]的子集合有 {},{a},{b},{c},{ab},{ac},{abc}**

**93、['a'=>200,'b'=>100,'c'=>100],写一个自定义排序函数，按值降序,如果值一样，按键排序**

冒泡排序

**94、设计一个缓存系统，可以定期或空间占满之后自动删除长期不用的数据，不能使用用遍历。**

我当时的答案是用链表来存,缓存命中就将该缓存移到链表头,然后链表尾就都是冷数据了。 我记得之前是在哪里看过这个设计,但我忘记在连接了,请知道朋友的把连接贴上来。

**95、==和===的区别，写出以下输出："aa"==1,"bb"==0，1=="1"**

- == 等于，不需要对比数据类型
- === 全等，需要对比类型

false, true, true

**96、一个排序好的数组,将它从中间任意一个位置切分成两个数组,然后交换它们的位置并合并，合并后新数组元素**

**如:20,21,22,25,30,1,2,3,5,6,7,8,15,18,19,写一个查询函数来查找某个值是否存在。**

## 97、设计一个树形结构，再写一个函数对它进行层序遍历

## 98、'$var'和"$var"的区别

双引号串中的内容可以被解释而且替换，而单引号串中的内容总被认为是普通字符。

在单引号串中甚至反斜杠也失去了他的扩展含义（除了插入反斜杠\和插入单引号'）。所以，当你想在字串中进行变量代换和包 含\n（换行符）等转义序列时，你应该使用双引号。单引号串可以用在其他任何地方，脚本中使用单引号串处理速度会更快些。

## 99、self 和 static 的区别

static：如果在子类中重写了父类中的 static 方法、属性，父类就会去访问了子类的 static 方法

self: 是类内指针，不管子类有没有重写过父类中的方法、属性都指向本类的静态方法、属性

## 100、PHP 的协程以及用途

http://www.laruence.com/2015/05/28/3038.html

https://blog.csdn.net/gavin_new/article/details/54603490

## 101、描述 autoload 的机制

https://blog.csdn.net/zhihua_w/article/details/52723402

## 102、mysql 中字段类型各占几个字节：smallint、int、bigint、datetime、varchar(8)

- smallint 2 字节
- int 4 字节
- bigint 8 字节



 

- datetime 8 字节
- varchar(8) 8*3 字节

http://www.jb51.net/article/55853.htm

## 103、哪些属性唯一确定一条 TCP 连接

## 104、myisam 和 innodb 的区别，为什么 myisam 比 innodb 快，myisam 和 innodb 的索引数据结构是什么样的?innodb 主键索引和非主键索引的区别?其索引上存放的数据是什么样的？

区别主要在数据和索引的存储结构和存储方式上，以及对于事务的支持。

参考：https://blog.csdn.net/chajinglong/article/details/56666771

## 105、断开 TCP 连接时，timewait 状态会出现在发起分手的一端还是被分手的一端

为什么建立 TCP 连接需要三次握手？ 原因：为了应对网络中存在的延迟的重复数组的问题 例子： 假设 client 发起连接的连接请求报文段在网络中没有丢失，而是在某个网络节点长时间滞留了，导致延迟到达 server。本来这是一个已经失效的连接报文，但是 server 接收到这个连接报文之后，误认为 client 发起了新的连接，于是向 client 发送确认报文段。此时因为没有了连接的 3 次握手，client 不会对 server 的确认报文作出回应，也不会向 server 发送数据，server 就以为连接已经建立，一直在空等 client 的数据，这样 server 的这一部分网络资源就被浪费了。

为什么断开 TCP 连接需要进行四次握手 ？ 因为 TCP 连接是全双工的网络协议，允许同时通信的双方同时进行数据的收发，同样也允许收发两个方向的连接被独立关闭，以避免 client 数据发送完毕，向 server 发送 FIN 关闭连接，而 server 还有发送到 client 的数据没有发送完毕的情况。所以关闭 TCP 连接需要进行四次握手，每次关闭一个方向上的连接需要 FIN 和 ACK 两次握手。

TIME_WAIT 状态的意义

在 TCP 连接中，当被动关闭连接的一方(图中 client)发送的 FIN 报文到达时，被动关闭连接的一方会发送 ACK 确认报文，并且进入 TIME_WAIT 状态，并且等待 2MSL 时间段(MSL:maximum segment life)。这么做有下述两个原因：

被动关闭连接的一方(图中的 server)在一段时间内没有收到对方的 ACK 确认数据包，会重新发送 FIN 数据包，因而主动关闭连接的一方需要停留在等待状态以处理



 

对方重新发送的 FIN 数据包。否则他会回应一个 RST 数据包给被动关闭连接的一方，使得对方莫名其妙。

在 TIME_WAIT 状态下，不允许应用程序在当前 ip 和端口上和之前通信的 client(这个 client 的 ip 和端口号不变)建立一个新的连接。这样就能避免新的连接收到之前的 ip 和端口一致的连接残存在网络中的数据包。这也是 TIME_WAIT 状态的等待时间被设置为 2MSL 的原因，以确保网络上当前连接两个方向上尚未接收的 TCP 报文已经全部消失。

https://www.cnblogs.com/zhoudayang/p/6012257.html

## 106、AWK 各种数据分析考得非常多，要多练习，题目不再一一写了

## 107、redis 中集合、有序集合、hyperLog、hash 的数据结构是啥样的

key value

## 108、描述一下:一个请求到达 nginx 的全部处理过程（nginx 自身会调用哪些逻辑）、然后怎么与 php 通信，中间的流程是什么样的等等？

https://www.jianshu.com/p/df89b530db89

https://blog.csdn.net/xiajun07061225/article/details/9309273

## 109、nginx 和 php-fpm 的相关配置,随便问里面各种参数啥意思

php-fpm 可以通过 tcp socket 和 unix socket 两种方式实现。

https://blog.csdn.net/koastal/article/details/52303316

## 110、假如有一张地图,如下图,"-"代表海洋、"+"代表陆地,用你最擅长的方式,取出陆地的坐标。

```
--++----++--+++---
-++++----+++++++--
-+++----++++------
-----++------++++-
---++++++-----+++-
-----++------+++-
```

比如上图在数组中表示成,1 表示成陆地,0 表示海洋:

```
[
    [0,0,1,1,0,0,0,1,1....],
    [0,1,1,1.....],
]
```

写个算法取出所有陆地的坐标,并按块放到一起,如地图上左上角第一个陆地的坐标是:

```
[
    [0,2],[0,3],
    [1,1],[1,2],[1,3],[1,4],
    [2,1],[2,2],[2,3]
]
```

## 111、Jsonp 的实现原理，你还知道哪些跨域方式？

- JSONP
- 添加响应头，允许跨域
- 代理的方式

## 112、如果某个博客通过判断 referer 方式来进行图片防盗链，如何破解？

curl 设置来源地址来欺骗对方服务器验证

## 113、简述 mysql 查询优化的本质，并举 2 个例子

## 114、设计一个秒杀系统，如何保证商品不超卖？

https://blog.csdn.net/zhoudaxia/article/details/38067003

## 115、单例模式的优点是什么？抽象类是什么？ 还了解哪些设计模式？

单例模式又称为职责模式，它用来在程序中创建一个单一功能的访问点，通俗地说就是实例化出来的对象是唯一的。 所有的单例模式至少拥有以下三种公共元素：

1. 它们必须拥有一个构造函数，并且必须被标记为 private
2. 它们拥有一个保存类的实例的静态成员变量

3. 它们拥有一个访问这个实例的公共的静态方法 单例类不能再其它类中直接实例化，只能被其自身实例化。它不会创建实例副本，而是会向单例类内部存储的实例返回一个引用。

抽象的类不能被实例化。任何一个类，如果它里面至少有一个方法是被声明为抽象的，那么这个类就必须被声明为抽象的。被定义为抽象的方法只是声明了其调用方式（参数），不能定义其具体的功能实现。

https://www.cnblogs.com/kangxl/p/6347179.html

工厂模式 适配器模式