新浪

php spl
autoload
file_get_contents
判断两个数组$a、$b
tp 和 Smarty 的区别
linux 命令查看 Apache 打开的端口 netstatus -anpl |grep 80
apache 和 nginx 区别

nginx 相对于 apache 的区别是：
轻量级，同样起 web 服务，比 apache 占用更少的内存及资源
抗并发，nginx 处理请求是异步非阻塞的，而 apache 则是阻塞型的，在高并发下 nginx 能保持低资源低消耗高性能
高度模块化的设计，编写模块相对简单
社区活跃，各种高性能模块出品迅速
apache 相对于 nginx 的优点：
rewrite ，比 nginx 的 rewrite 强大
模块超多，基本想到的都可以找到
少 bug ，nginx 的 bug 相对较多
存在就是理由，一般来说，需要性能的 web 服务，用 nginx 。如果不需要性能只求稳定，那就 apache 吧。后者的各种功能模块实现得比前者，例如 ssl 的模块就比前者好，可配置项多。这里要注意一点，epoll(freebsd 上是 kqueue）网络 IO 模型是 nginx 处理性能高的根本理由，但并不是所有的情况下都是 epoll 大获全胜的，如果本身提供静态服务的就只有寥寥几个文件，apache 的 select 模型或许比 epoll 更高性能。
当然，这只是根据网络 IO 模型的原理作的一个假设，真正的应用还是需要实测了再说的。
)
...
京东

笔试大题：
输入整数 n,求 m，m>9，m 中各个数位的乘积=n 的最小整数;如 n=36，m=49;
二叉树前序遍历的非递归实现（本文的总结）
求第 n 个数，这个序列满足(2^i)*(3^j)*(5^k),前 7 个为：2,3,4,5,6,8,10...
整体不是特别难，除了第三道附加算法题时间没来得及做，其他两道和选择题自我感觉还不错

一面：
PHP 有多线程么（没有，但可以通过其他方式实现），怎样理解 PHP
对于静态页面文件，是放在 ngnix 端，还是 Server 端 (ngnix 处理静态页面性能比 apache 要快三倍以上，所以直接放在 ngnix)
php 写接口，通信方式，socket、http 通信实现

两台计算机如何通信，浏览器输入 url 后，一直到页面呈现，中间服务器都经过了哪些流程
第一步：客户机提出域名解析请求,并将该请求发送给本地的域名服务器。

第二步：当本地的域名服务器收到请求后,就先查询本地的缓存,如果有该纪录项,则本地的域名服务器就直接把查询的结果返回。

第三步：如果本地的缓存中没有该纪录,则本地域名服务器就直接把请求发给根域名服务器,然后根域名服务器再返回给本地域名服务器一个所查询域(根的子域)的主域名服务器的地址。

第四步：本地服务器再向上一步返回的域名服务器发送请求,然后接受请求的服务器查询自己的缓存,如果没有该纪录,则返回相关的下级的域名服务器的地址。

第五步：重复第四步,直到找到正确的纪录。

用 php 做客户端接口应注意什么问题（数据加密、json 传输等）
json 格式数据有哪些特点（并不能保证安全，对于所有语言都是统一的 key-value 处理规范）
验证码安全，google 破解其他站点
md5 怎样逆置，其他加密方式的加密、逆置原理
验 证码的原理。验证码的实现原理，如果验证码是存储在服务器 session，那如果正在输入的时候，服务器与客户端断开连接，又连上去(session 已经 失效)，提交后怎样判断验证码正确性(

在客户端 js 中构造和服务器端相同的 sessionid，提交时与服务器端 sessionid 匹配)
数据库主键怎样理解，它是索引么
建立数据库应该注意哪些（表冗余、主键、外键、索引、字段...）
数据库事务是什么 （事务处理可以确保除非事务性单元内的所有操作都成功完成，否则不会永久更新面向数据的资源。
)
如果正在下单状态，将用户银行卡状态修改为正在使用状态 1，下一步付款、修改订单状态时，突然断网怎么办 （面试官期待答案：不是马上回滚，而是在这个地方尝试重试几次，若还不成

功就 rollback）
请求量比较大时应从哪方面优化提高性能（缓存、页面静态化、sql 优化、表结构、水平分割、垂直分割）
Linux 常用命令 （文件操作命令、vim 命令、系统命令）
Javascript 和 jquery 的区别
学校活动、个人性格、工作地域调整、薪资等...
腾讯后台开发面

2.core 文件是什么，有什么用？

答：程序崩溃以后，会把最后的栈信息存在 core 文件里，方便程序员了解程序崩溃前最后的栈信息。

追问：如果程序 core 了，但没产生 core 问价，是何原因？

答：没有打开 core 文件生成的开关。ulimit -c unlimit

追问：不用 core 文件，程序出了问题产生信号是否知道？

答：当时蒙了，现在回想起来，就是内核向进程发信号嘛。没答出来。

3.共享内存，不使用同步方式，是否可以安全读写？

答：这一题答偏了，不知怎么扯到 fork 上去了。这一题应该是不行，共享内存属于临界区，应该要同步，否则两个进程同时操作一个内存区就出问题了。可以使用读写锁来同步。

4.fork 后，子进程保留了父进程的什么？

答：子进程的内存区是父进程的副本，堆栈等都会继承过来，还有打开的文件描述符等。其实还有很多，比如实际用户 ID,有效用户 ID，当前工作目录，存储映射等等。

5.共享内存除了文件映射还有什么方式？

答：共享内存对象映射。

追问：二者有什么区别？

答：不知道。

6.tcp 怎么实现流量控制？

答：对端告知窗口大小。本端传递的数据量小于窗口大小。（更好的说法是告诉对端本地的窗口的大小，对端传递的数据量必须小于该窗口）

追问：怎么告知窗口大小？

答：在 ACK 的报文里。

7.编程题（没搞出来，诶~）

问：一个超长字符串表示的十进制数（大于 2^32)，转化为十六进制的字符串？

赶集网面试总结

面试渠道：小伙伴 @蔚然 内推技术实习岗
面试时间：8 月 14 日 下午 13:30

第一面：
面试方式：2 对 1　（技术 leader 和前端负责人）
过程叙述：
问的问题很广，php 方面、算法数据结构、设计模式、前端、数据库、Linux 等。
内容大概有：
当字符串为 null 时，isset(false)和 empty(true)的区别
==和===的区别

===是恒等计算符　同时检查表达式的值与类型

==是比较运算符号　不会检查条件式的表达式的类型

PHP 魔术方法有哪些，作用是什么
__construct()构造函数是目前为止最经常使用的函数。在创建对象时，可以在构造函数中做一些初始化工作。可以为构造函数定义任意多个参数，只要在实例化时传入对应个数的参数即可。构造函数中出现的任何异常都会阻止对象的创建。

__desctruct()

析构函数通常在对象被销毁时调用，析构函数不接收任何参数。经常在析构函数中执行一些清理工作，比如关闭数据库连接等。

魔术方法__get()在我们尝试访问一个不存在的属性时会被调用。它接收一个参数，该参数表示访问属性的名字，并且将该属性的值返回。

__set()魔术方法在我们尝试修改一个不可访问的属性时会被调用，它接收两个参数，一个表示属性的名字，一个表示属性的值。

__isset()魔术方法在对一个不可访问的属性调用 isset()方法时会被调用，它接收一个参数，表示属性的名字。它应该返回一个布尔值，用来表示该属性是否存在。

__unset()魔术方法在调用 unset()函数销毁一个不能访问的属性时会被调用，它接收一个参数，表述属性的名字。

__toString()在我们将对象当作字符串一样使用时会被调用，它不接收任何参数。该方法允许我们定义对象的表现形式。
__clone()魔术方法__clone()可以解决上面的问题。当对一个对象使用 clone 关键字时，该魔术方法会被调用。

__sleep()

魔术方法__sleep()在对一个对象序列化时(调用 serialize())会被调用。它不接收任何参数，而且应该返回一个包含所有应该被序列化的属性的数组。在该魔术方法中，也可以执行一些其他操作。
有一点要注意的是，不要再该函数中进行任何的析构操作，因为这可能会影响正在运行的对象。

魔术方法__wakeup()在对存储的对象反序列化时会被调用。它不接收任何参数，也没有任何返回值。可以用它来处理在序列化时丢失的数据库连接或资源。

魔术方法__call()在调用不存在或不可访问的方法时会被调用。它接收两个参数，一个是调用的方法的名字，一个是包含函数参数的数组。我们可以使用这种方法调用子对象中得同名函数。

用过哪些数组函数，

```
array_values($arr);        //获得数组的值
array_keys($arr);          //获得数组的键名
array_flip($arr);          //数组中的值与键名互换（如果有重复前面的会被后面的覆盖）
array_search('PHP',$arr); //检索给定的值，加 true 则是严格类型检查
array_reverse($arr);       //将数组中的元素翻转(前后顺序)
in_array("apple", $arr);  //在数组中检索 apple
array_key_exists("apple", $arr); // 检索给定的键名是否存在数组中
array_count_values($arr);        // 统计数组中所有值出现的次数
array_unique($arr);
```

常见的排序算法和查找算法有哪些，数据结构的树、链表、堆、栈，是自己实现过还是只看过
说一下快速排序的最好和最坏时间复杂度 （nlogn　 n 的平方）
快速排序是对冒泡排序的优化说以最好时间复杂度为 nlogn，最坏为 n 的平方
简述快速排序的原理　面试前最好先熟悉一下常用算法和数据结构，另外不可忽略了时间复杂度

二分查找的原理和时间复杂度（logn）时间复杂度第一次回答错了

你熟悉的设计模式有哪些，手写一个设计模式（当时我写的是单例）

```
class Singleton(){
   private static $instance;
   public static function getInstance(){
      if(null==self::$instance){
         slef::$instance=new Db();
```

```
        }
        return self::$instance;
    }
    private function __destruct(){}
    private function __cloe{};
}


abstract class Parent(){

}
class childer1 extends Parent(){}
class childer2 extends Parent(){}

class Destory(){
    public function create(){
        if($status==1){
            return new childer1();
        }else{
            return new childer2();
        }
    }
}
```

final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。

finally 是异常处理语句结构的一部分，表示总是执行。

finalize 是 Object 类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。

2 开头 （请求成功）表示成功处理了请求的状态代码。

200 （成功） 服务器已成功处理了请求。 通常，这表示服务器提供了请求的网页。

201 （已创建） 请求成功并且服务器创建了新的资源。

202 （已接受） 服务器已接受请求，但尚未处理。

203 （非授权信息） 服务器已成功处理了请求，但返回的信息可能来自另一来源。

204 （无内容） 服务器成功处理了请求，但没有返回任何内容。

205 （重置内容） 服务器成功处理了请求，但没有返回任何内容。

206 （部分内容） 服务器成功处理了部分 GET 请求。

3 开头 （请求被重定向）表示要完成请求，需要进一步操作。 通常，这些状态代码用来重定向。

300 （多种选择） 针对请求，服务器可执行多种操作。 服务器可根据请求者 (user agent) 选择一项操作，或提供操作列表供请求者选择。

301 （永久移动） 请求的网页已永久移动到新位置。 服务器返回此响应（对 GET 或 HEAD 请求的响应）时，会自动将请求者转到新位置。

302 （临时移动） 服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。

303 （查看其他位置） 请求者应当对不同的位置使用单独的 GET 请求来检索响应时，服务器返回此代码。

304 （未修改） 自从上次请求后，请求的网页未修改过。 服务器返回此响应时，不会返回网页内容。

305 （使用代理） 请求者只能使用代理访问请求的网页。 如果服务器返回此响应，还表示请求者应使用代理。

307 （临时重定向） 服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。

4 开头 （请求错误）这些状态代码表示请求可能出错，妨碍了服务器的处理。

400 （错误请求） 服务器不理解请求的语法。

401 （未授权） 请求要求身份验证。 对于需要登录的网页，服务器可能返回此响应。

403 （禁止） 服务器拒绝请求。

404 （未找到） 服务器找不到请求的网页。

405 （方法禁用） 禁用请求中指定的方法。

406 （不接受） 无法使用请求的内容特性响应请求的网页。

407 （需要代理授权） 此状态代码与 401（未授权）类似，但指定请求者应当授权使用代理。

408 （请求超时） 服务器等候请求时发生超时。

409 （冲突） 服务器在完成请求时发生冲突。 服务器必须在响应中包含有关冲突的信息。

410 （已删除） 如果请求的资源已永久删除，服务器就会返回此响应。

411 （需要有效长度） 服务器不接受不含有效内容长度标头字段的请求。

412 （未满足前提条件） 服务器未满足请求者在请求中设置的其中一个前提条件。

413 （请求实体过大） 服务器无法处理请求，因为请求实体过大，超出服务器的处理能力。

414 （请求的 URI 过长） 请求的 URI（通常为网址）过长，服务器无法处理。

415 （不支持的媒体类型） 请求的格式不受请求页面的支持。

416 （请求范围不符合要求） 如果页面无法提供请求的范围，则服务器会返回此状态

代码。

417 （未满足期望值） 服务器未满足"期望"请求标头字段的要求。

5 开头（服务器错误）这些状态代码表示服务器在尝试处理请求时发生内部错误。 这些错误可能是服务器本身的错误，而不是请求出错。

500 （服务器内部错误） 服务器遇到错误，无法完成请求。

501 （尚未实施） 服务器不具备完成请求的功能。 例如，服务器无法识别请求方法时可能会返回此代码。

502 （错误网关） 服务器作为网关或代理，从上游服务器收到无效响应。

503 （服务不可用） 服务器目前无法使用（由于超载或停机维护）。 通常，这只是暂时状态。

504 （网关超时） 服务器作为网关或代理，但是没有及时从上游服务器收到请求。

505 （HTTP 版本不受支持） 服务器不支持请求中所用的 HTTP 协议版本。

数据库中的索引有哪些，数据库引擎区别（MyIsam 和 InnoDB）

1. InnoDB 不支持 FULLTEXT 类型的索引。
2. InnoDB 中不保存表的具体行数，也就是说，执行 select count(*) from table 时，InnoDB 要扫描一遍整个表来计算有多少行，但是 MyISAM 只要简单的读出保存好的行数即可。注意的是，当 count(*)语句包含 where 条件时，两种表的操作是一样的。
3. 对于 AUTO_INCREMENT 类型的字段，InnoDB 中必须包含只有该字段的索引，但是在 MyISAM 表中，可以和其他字段一起建立联合索引。
4. DELETE FROM table 时，InnoDB 不会重新建立表，而是一行一行的删除。
5. LOAD TABLE FROM MASTER 操作对 InnoDB 是不起作用的，解决方法是首先把 InnoDB 表改成 MyISAM 表，导入数据后再改成 InnoDB 表，但是对于使用的额外的 InnoDB 特性（例如外键）的表不适用。

叙述联合索引，和主键的区别（把主键和索引搞混了，囧..）

主键，是一种特殊的唯一索引，在一张表中只能定义一个主键索引，主键用于唯一标识一条记录，使用关键字 PRIMARY KEY 来创建。
索引可以覆盖多个数据列，如像 INDEX(columnA, columnB)索引，这就是联合索引。

联合主键，顾名思义就是多个主键联合形成一个主键组合，体现在联合。
(主键原则上是唯一的，别被唯一值所困扰。)
索引可以极大的提高数据的查询速度，但是会降低插入、删除、更新表的速度，因为在执行这些写操作时，还要操作索引文件。

数据库表优化方法，建表的注意事项和原则，常见数据库字段类型

第一范式：1NF 是对属性的原子性约束，要求属性具有原子性，不可再分解；
第二范式：2NF 是对记录的惟一性约束，要求记录有惟一标识，即实体的惟一性；
第三范式：3NF 是对字段冗余性的约束，即任何字段不能由其他字段派生出来，它要求字段没有冗余.

普通索引：最基本的索引，没有任何限制
唯一索引：与"普通索引"类似，不同的就是：索引列的值必须唯一，但允许有空值。
主键索引：它 是一种特殊的唯一索引，不允许有空值。
全文索引：仅可用于 MyISAM 表，针对较大的数据，生成全文索引很耗时好空间。
组合索引：为了更多的提高 mysql 效率可建立组合索引，遵循”最左前缀“原则。

MyISAM 中索引检索的算法为首先按照 B+Tree 搜索算法搜索索引，如果指定的 Key 存在，则取出其 data 域的值，然后以 data 域的值为地址，读取相应数据记录。在 MyISAM 中，主索引和辅助索引（Secondary key）在结构上没有任何区别，只是主索引要求 key 是唯一的，而辅助索引的 key 可以重复。
InnoDB 的数据文件本身就是索引文件。InnoDB 的辅助索引 data 域存储相应记录主键的值而不是地址。聚集索引这种实现方式使得按主键的搜索十分高效，但是辅助索引搜索需要检索两遍索引：首先检索辅助索引获得主键，然后用主键到主索引中检索获得记录。

sql 注入有哪些，如何有效防止

所谓 SQL 注入，就是通过把 SQL 命令插入到 Web 表单提交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的 SQL 命令。具体来说，它是利用现有应用程序，将（恶意）的 SQL 命令注入到后台数据库引擎执行的能力，它可以通过在 Web 表单中输入（恶意）SQL 语句得到一个存在安全漏洞的网站上的数据库，而不是按照设计者意图去执行 SQL 语句。比如先前的很多影视网站泄露 VIP 会员密码大多就是通过 WEB 表单递交查询字符暴出的，这类表单特别容易受到 SQL 注入式攻击.
防护
归纳一下，主要有以下几点：
1.永远不要信任用户的输入。对用户的输入进行校验，可以通过正则表达式，或限制长度；对单引号和双"-"进行转换等。
2.永远不要使用动态拼装 sql，可以使用参数化的 sql 或者直接使用存储过程进行数据查询存取。
3.永远不要使用管理员权限的数据库连接，为每个应用使用单独的权限有限的数据库连接。
4.不要把机密信息直接存放，加密或者 hash 掉密码和敏感的信息。
5.应用的异常信息应该给出尽可能少的提示，最好使用自定义的错误信息对原始错误信息进行包装
6.sql 注入的检测方法一般采取辅助软件或网站平台来检测，软件一般采用 sql 注入检测工具 jsky，网站平台就有亿思网站安全平台检测工具。MDCSOFT SCAN 等。采用 MDCSOFT-IPS 可以有效的防御 SQL 注入，XSS 攻击等。

用过哪些缓存技术，thinkphp 的缓存是局部缓存 还是完全缓存
Memcaech 和 radis 的区别

Cookie 和 Session 区别
禁用 Cookie 后，Session 还可以用吗，有什么解决方案

可以使用，可以通过 URL 传输 sessionID

常用 Linux 命令（我当时从文件操作和 vim 两方面回答）

1. man 对你熟悉或不熟悉的命令提供帮助解释

eg:man ls 就可以查看 ls 相关的用法

注：按 q 键或者 ctrl+c 退出，在 linux 下可以使用 ctrl+c 终止当前程序运行。

2. ls 查看目录或者文件的属*，列举出任一目录下面的文件

eg: ls /usr/man

ls -l

a.d 表示目录(directory)，如果是一个"-"表示是文件，如果是 l 则表示是一个连接文件(link)

b.表示文件或者目录许可权限.分别用可读(r)，可写(w)，可运行(x)。

3. cp 拷贝文件

eg: cp filename1 filename2 //把 filename1 拷贝成 filename2

cp 1.c netseek/2.c //将 1.c 拷到 netseek 目录下命名为 2.c

4. rm 删除文件和目录

eg: rm 1.c //将 1.c 这个文件删除

5. mv 移走目录或者改文件名

eg: mv filename1 filename2 //将 filename1 改名为 filename2

mv qib.tgz ../qib.tgz //移到上一级目录

6. cd 改变当前目录 pwd 查看当前所在目录完整路径

eg: pwd //查看当前所在目录路径

cd netseek //进入 netseek 这个目录

cd //退出当前目录

7. cat，more 命令

将某个文件的内容显示出来。两个命令所不同的是:cat 把文件内容一直打印出来，而 more 则分屏显示

eg; cat>1.c //就可以把代码粘帖到 1.c 文件里，按 ctrl+d 保存代码。

cat 1.c 或 more 1.c //都可以查看里面的内容。

gcc -o 1 1.c //将 1.c 编译成.exe 文件，我们可以用此命编译出代码。

8.chmod 命令 权限修改 用法：chmod 一位 8 进制数 filename。

eg: chmod u+x filenmame //只想给自己运行，别人只能读

//u 表示文件主人， g 表示文件文件所在组。 o 表示其他人 ;r 表可读，w 表可写，x 表可以运行

chmod g+x filename //同组的人来执行

9. clear，date 命令

clear:清屏，相当与 DOS 下的 cls;date:显示当前时间。

10. mount 加载一个硬件设备

用法:mount [参数] 要加载的设备 载入点

eg: mount /dev/cdrom

cd /mnt/cdrom //进入光盘目录

11. su 在不退出登陆的情况下，切换到另外一个人的身份

用法: su -l 用户名(如果用户名缺省，则切换到 root 状态)

eg:su -l netseek (切换到 netseek 这个用户，将提示输入密码)

12.whoami，whereis，which，id

//whoami:确认自己身份

//whereis:查询命令所在目录以及帮助文档所在目录

//which:查询该命令所在目录(类似 whereis)

//id:打印出自己的 UID 以及 GID。(UID:用户身份唯一标识。GID:用户组身份唯一标识。每一个用户只能有一个唯一的 UID 和 GID)

eg: whoami //显示你自己登陆的用户名

whereis bin 显示 bin 所在的目录，将显示为：/usr/local/bin

which bin

13. grep，find

grep:文本内容搜索;find:文件或者目录名以及权限属主等匹配搜索

eg: grep success * /*查找当前目录下面所有文件里面含有 success 字符的文件

14. kill 可以杀死某个正在进行或者已经是 dest 状态的进程

eg; ps ax

15. passwd 可以设置口令

16. history 用户用过的命令

eg: history //可以显示用户过去使用的命令

17. !! 执行最近一次的命令

18. mkdir 命令

eg: mkdir netseek //创建 netseek 这个目录

19. tar 解压命令

eg: tar -zxvf nmap-3.45.tgz //将这个解压到 nmap-3.45 这个目录里

20. finger 可以让使用者查询一些其他使用者的资料

eg: finger //查看所用用户的使用资料

finger root //查看 root 的资料

:q 不保存退出

:q! 不保存强制性退出

:w 保存编辑

:w filename 存入文件 filename 中

:w! filename 强制性存入文件 filename 中

:wq / :x / ZZ 保存退出

:wq! 强制保存然后退出

是否了解 ngnix，和 apache 作用相同
常见 apache 配置是否熟悉，apache 中的 rewrite 作用
php 开发过程中用过哪些编译器（eclipse、phpstrom、zend，一般的 sublimeText）
对前端的 html、css、js 是否熟悉，用过哪些前端框架　vue,angular

第一面结束后问了有什么问题，我问了公司开发过程中使用的技术架构（lnmp）。然后然我等一下，当时心情比较放松，因为感觉除了数据库方面回答不太流 利，其他都还可以。在会

议室正纳闷接下来会发生什么，一会儿来了一个气场十足的大牛。接下来的面试有点紧张...

第二面：
面试方式：1 对 1
过程叙述：
　　问的问题很深，感觉平时做项目太偏应用，底层的知识比较欠缺
　　内容大概有：
解释了一下为什么大学是五年制(囧,每次回答这个问题都要从很古老的一段伤心往事说起)
平时学习的渠道（我说刚开始基础是从书上学习，后面都是项目驱动，然后人直接说了项目驱动的弊端...）
索引的最左前缀原则 （有点熟悉，好像在哪看到过，但是真想不起来了）

为什么要用 ThinkPHP，和直接写 php 比有哪些优缺点
是否阅读过 tp 源码
既然学习 thinkphp，解释一下 tp 中的 hook 机制吧（只在新版本更新时候看过，不太熟悉啊，紧张...）
什么是 reset ful （呃...这是什么，真心不知道）
解释一下 php 中的 fastcgi，那 cgi 是什么，cgi 和 fastcgi 的区别 （不太熟悉，唉，硬着头皮上吧）
刚才第一面问过你设计模式了，什么是共享模式 （晕啊，设计模式那么多，好紧张，想不起来了）
解释一下 PHP 的 namespace
定义命名空间 可以调用个域下面的下面所申明的 class, interface, const

你平时开发用的 php 版本是什么 （顿时好轻松，5.5 啊）

三，再容我休息几天嘛。接下来回复 offer，心情比较纠结，到底去不去，想想现在也没什么事，就先实习 着再继续准备校招吧。准备入职材料。

百度

笔试题型

百度技术岗的笔试题，是每个大部门自己出题的，所以，每年笔试完都会有数套笔试题在网上流传。
考察面主要包括：
1. 基本要求
语言（主要是 C/C++）、计组原理、操作系统原理、计算机网络。如指针运算、字节对齐、函数调用栈帧结构、内存管理、进程调度、网络协议七层模型等。通常以问答题形式出现。
2. 数据结构及算法
查 找（典型如二分查找）、排序（典型如快排、外排）、树（典型如 2 叉树的前/中/后序遍历、trie 树）。人品好的话，是题目明确告诉你"请写出 2 叉树中序 遍历的算法伪码并分析复杂度"

，这种题目还算简单。不走运的话，题目相对抽象，比如"给定一个存放几亿个整数值的文件，设计一种算法，在满足 xx 空间复杂 度或 xx 时间复杂度的前提下，对这些整数做 xx 处理"，这种题

目是比较难对付的。所以，大家要有心理准备。这部分通常是写算法伪码。
3. 系统设计题
文字描述一段实际的业务场景，要求设计一个系统，能多快好省地实现 xxx 功能。
这种题目比较发散，通常，大部分校招学生对这类设计题都不可能有啥经验。基本是凭借自己对系统，尤其是计算机组成原理或操作系统的理解，触类旁通，自由发挥，做到有理有据，逻辑

第一波:众荟网技术部
cookie 禁用后 session 是否还能使用？
（回答：可以）接着问：可以通过哪些方式？
Session URL 重写,保证在客户端禁用或不支持 COOKIE 时,仍然可以使用 Session。

解释__call()，__construct()，__isset()，__get()

获取中文字符串的长度，实现截取中文字符串
Mb_strlen($str,'UTF-8'); mb_substr($str,$i,1,'UTF-8');

如何设计数据库

垂直分割和水平分割依据是什么
说出常用的数据库索引，并作相应的解释
说出常用的数据库引擎
接着问：MyISAM 和 INNODB 的区别
然后是一个读程序写结果（这道题不难，考的是 PHP 中的念想对象相关的知识）
git 常用命令，git 的分支操作命令

创建分支 $ git branch 分支名
推送到远程 $ git push origin 分支名

git 如何回退版本
git reset -hard

第二波：e 袋洗
打印出昨天的日期 2015-08-18 08:08:18;
date('Y-m-d H:i:s',strtotime('-1 day',time()));

写出三种常用的三种开源的数据库
Mysql ,nosql,oracl
写出三种以上的数据库引擎
MyIsam,InnoDb,Bdb,Memory

用 js 写出 3 种创建 image 标签的写法
写出你见过的 PHP 文件的后缀

排序方法有哪些，写出一种排序方法
PHP 遍历一个目录下的目录子目录和文件（提示：可以递归）
解释以下协议：SMTP、POP3、HTTP、FTP、DNS
如何实现即时通信？给出至少给出两种方案
在网站注册登录的基础上，第三方接入登录数据库如何设计，消息推送如何实现？推送到哪？
zendFramework
PHP7 有哪些新技术
你通过哪些渠道获取新技术
还有一道数据库方面的问题(简单)
正则表达匹配邮箱和 url
nginx、FPM
解释下 session 的机制
这家面试感觉有种被虐的感觉，问的好多东西我都没有接触过，感觉很受单击。。。。不过这又让认识到我的不足。。。感谢 e 袋洗
第三波：汇通天下物流
写出你常用的 Linux 命令
写出你知道的魔术方法并解释
写出常用的设计模式，并实现其中的一个
多服务器如何实现 session 共享
js 实现单击按钮 A(id='a')修改按钮 B(id='b')的显示为 loading...并禁用按钮 A,B
正则表达式匹配邮箱
include 和 require 的区别
MVC 优点和缺点
解释：什么是面向对象？
一道数据库设计题，当时让写 sql 忘得差不多了，但是感觉写的还行
最后一题：N 个数，写一个高效的算法判断是否存在两个数的和是 P
。。。。额。。。忘了。。总得来说答的还行

第四波：壹号车
>一面：技术

写出斐波那契数列中第 N 个数的的值（要求：递归和非递归分别实现）
用 MVC（tp）描述登录的实现
数据库方面的问题：一个用户信息表，实现按指定方式（从当前日期开始：按月，按季度，按年。。）显示出即将过生日的人都是谁，（当用 between 时候，跨年是是否会存在问题？）
。。。。。
>二面：技术（这一面都是数据结构和算法）

一组数字要求奇数在前，偶数在后要求空间复杂度 O（1）；
多维数组$arr[a][b][c]现在换种形式封装一个函数 test($arr,$path)参数 path 就是数组的深度例如：a,b,c
。。。
>三面：复试(CTO)

ORM
一个一维数组下标 0-99，随机函数生成 1-100 的数插入数组要求不能重复（要求高效）
。。。。

第五波：易车网
这是面试中最简单的一家了，感觉面的不错，改天在总结
第六波：360

这个部门是云引擎部门，首先做了一套面试题，然后技术面，面了很久，到 HR 面了但是 hr 下班了，不过不太喜欢哪个部门，拿到 offer
第七波：楚楚街
>技术面：

先说几个魔术方法并解释
PHP 数组底层是什么数据结构？
hash 的时间复杂度是多少？
PHP5.3 之前有一个请求时候产生的 bug 知道是什么吗？
hash 如何解决冲突？，
1、开放地扯法
2、再哈希法
3、链地址法
4、建立一个公共溢出区

线性探测后 hash 的时间复杂度是多少？

知道什么是原子性吗？解释下
一个场景：当一个接口用来处理访问次数，请问如何统计请求是次数，（Redis 的原子性函数）
了解 git 吗？

>hr 面：

谈了下我什么时候能入职，来北京多长时间了，有没有收到其他公司的 offer，她给 offer 我没有马上同意接受，说下周一给回复，薪资很诱人，拿到 offer

8.新浪微博
>一面：

很和蔼的一个 程序员，拿到我的简历后问，你简历上的这些项目哪些是让你印象深刻的或者说让你学到很多的东西的，简单说一下
说一下常用的 PHP 数组函数并讲一下都是干嘛用的，然后又让写了一些常用的字符串处理函数
一道简单的算法题：合并两个有序数组

```
function merge($arrA,$arrB){
    $a_i=$b_j=0;
    $len_a=count($arrA);
    $len_b=count($arrB);
    while($a_i<$len_a&&$b_i<$len_b){
        if($arrA[$i]<$arrB[$j]){
            $arrC[]=$arrA[$i++];
        }else{
            $arrC[]=$arrB[$j++];
        }
    }
    while($a_i<$len_a){
        $arrC[]=$arrA[$i++];
    }
    while($b_i<$len_b){
        $arrC[]=$arrB[$j++];
    }
    return $arrC;
}
```

又一道简单的算法题：汉诺塔（这不是重点，写完后让你讲讲处理流程，最后竟然又让我在纸上画出一步步如何挪的，三个盘子、四个盘子、五个盘子!!!! 真是服了）
讲一下什么是递归，递归用来处理哪些问题？
然后是一些数据库的题问的很杂，这里就不一一总结了
这一面的问题问的挺广泛地 ，问的很杂，具体的一些细节想不起来了这是 5 天前面的今天才总结。。。这一面我感觉面的很不错
>二面：经理

很有气场的一个人，那人嚼着口香糖长的又高又胖当时那十足的气场还真把我给镇住了。见到我后二话不说上去写了个 PHP 语句：
<? echo "aa",$_GET['r'],'"bb" ?>然后问我：这样写有没有问题？都是有哪些问题一一说出来

php.ini 中如何配置短标签？属性点名字是什么？short_open_tags
说一下你对 memchace 和 redis 的理解

然后问了 PHP 的一些扩展和 php.ini 文件的配置
mysql 单表的最大容量是多少？

char(10)和 int(10)的含义
GBK 和 UTF8 各占多少个字符？
如何优化高并发大流量的数据库？
//数据库读写分离，给表建立外键，使用表分割技术（水平分割，垂直分割）
现有表 A(id,url,time) 现需要查询出 A 表中每个用户发的访问第一个 url 并同时创建表 B,表 B 的结构和表 A 完全相同内容为查询的结果，要求用一条 SQL 语句实现
新浪微博的主页可以做哪些优化？如果让你想你能想到哪些？
。。。
这一面面的很吃力，但是总体还是不错的，给了口头 offer 让回去等 hr 电话
>hr

hr 是在面试两天后才给我打的电话，是中午 1 点多那是我正睡觉，谈了薪资说 150/1 天，当时我的心哇凉哇凉的。。我说我有个同学（赵亚飞童鞋）也是在微博人家都是 200/1 天，hr 说那可能是

产品我说他就是开发的，hr 说她不清楚。。。
然后他要我的身份证号录信息，我说我在考虑考虑就给她挂了。。。。新浪再见

9.百度音乐
>一面：技术

好吧，先来个简单的自我介绍
看了简历后就问那个项目你印象最深说一下，讲讲项目的特色
memchache、redis 的理解
PHP 数组函数都是有哪些你知道的，写一下；字符串函数都是有哪些，写一下
写了一大堆后他从中挑了几个让我解释什么意思，然后又给场景如何利用函数实现，easy
搭建过 lamp 环境吗？简单说说搭建流程
说说常用的 linux 命令，问的还真是多
如何反转数组？不用 PHP 的函数要求时间复杂度最低
二叉树的特性都是有哪些？说一下
写一下树的遍历，前序遍历（我用的递归），非递归会写吗？然后又写了非递归

然后是一大堆的数据库方面的问题。。。
>二面：leader

PHP 底层的扩展是如何加载的？
redis 的底层是如何实现的
php.ini 中常用配置都是有哪些？说你知道的

php.ini 中的 safe_mode 是干什么用的
一道算法题：字符串中出现次数最多的子串，以及出现次数
一道算法题：n 阶楼梯，一次可以上一阶、上二阶、上三阶问共有多少种上法（剑指 offer 上的）
>三面：经理

说给实习 offer，等公司分出去后再发校招 offer，回去等 hr 电话，好吧今天是周五
未完待续~~~~

---------- 华丽的分割线

参考答案
[php] view plain copy
```
<?php
/**
 * Created by PhpStorm.
 * User: qishou
 * Date: 2015/8/29
 * Time: 9:29
 */
header("content-type:text/html;charset=utf-8");
/*================================ 众 荟 网 技 术 部 ====================================*/

//截取中文字符串
$str = "中国 hello world !";
echo substr($str,0,6)."<br/>";
echo mb_substr($str,0,6,'utf8')."<br/>";
echo strlen($str)."<br/>";
echo mb_strlen($str,'utf-8')."<br/>";

//cookie 禁用后 session 是否还能使用？
//答：可以通过 url,表单进行提交

//如何设计数据库
//1.为数据库选择合适的存储引擎
//2.遵循三范式建立合理的表结构
//3.添加适当的索引

//垂直分割和水平分割的依据是什么？
//答：根据实际的业务需求

//常用的数据库索引
```

```
//1.主键索引
//2.唯一索引
//3.联合索引
//4.普通索引

//常用的数据库引擎
//1.MYISAM
//2.INNODB
//3.heap
//4.BDB

//MYISAM 和 INNODB 的区别
//1.MYISAM 不支持事务，innodb 支持事物
//2.myisam 是表级锁，innodb 是行级锁
//3.msIsam 支持全文索引，innodb 不支持全文索引
//4.myisam 常用于 select 较多的数据库，innodb 常用于更新较多的数据库

//git 如何回退历史版本
//git log 查看版本历史
//git reset --hard 39458493859;

/*==============================e 袋洗=====================================*/
// 打印出昨天的的日期
echo date('Y-m-d H:i:s',strtotime("-1 day"))."<br/>";

//写出三种以上常用的开源数据库
//mysql , sqllite, BDB ,firebird

//写出三种以上的数据库引擎
//myisam, innodb, BDB, heap

//js 写出三种创建 image 标签的方式
//var image    = new image();
//var image = document.createElemetById('image'); //createElementById();用来创建对象
//image.innerHTML("<image src=''>");

//写出你所见过的 PHP 文件的后缀
//.php , .inc , .conf , .tpl ,

//写出你所用过的 PHP 的扩展
//mb_string , PDO ， mysql , GD, socket ,curl

//排序都是有哪些，写出一种排序
//排序有：冒泡排序，简单选择排序，直接插入排序，shell 排序，堆排序，归并排序，快速
```

排序

```
//例如：快速排序
function quick_sort($array){
	$length = count($array);
	if($length<=1){
		return $array;
	}
	$privotkey = $array[0];
	$left_array = array();
	$right_array = array();
	for($i=0;$i<$length;$i++){
		if($array[$i]<=$privotkey){
			$left_array[] = $array[$i];
		}else{
			$right_array[] = $array[$i];
		}
	}
	$left_array = quick_sort($left_array);
	$right_array = quick_sort($right_array);

	return array_merge($left_array,array($privotkey),$right_array);
}

//PHP 遍历指定目录下所有文件
function get_all_file($path){
	$files = array();
	if(is_dir($path)){
		if($handle = opendir($path)){
			while($file = readdir($handle)){
				if($file!='.' && $file!='..'){
					$child_path = $path.'/'.$file;
					if(is_dir($child_path)){
						$files[$file] = get_all_file($child_path);
					}else{
						$files[] = $child_path;
					}
				}
			}
		}
		closedir($handle);
		return $files;
	}
}
$path = "f:";
```

```
//echo "<pre>";
//print_r(get_all_file($path));

//解释下列网络协议
//1.SMTP:简单邮件协议
//2.POP3：邮局协议
//3.HTTP：超文本协议
//4.FTP：文件传输协议
//5.DNS：域名解析服务

/*<script>
	var image1 = new Image(); //面向对象的方法
	var image2 = document.createElement("image"); //document.createElement()方法
	//image1.innerHTML('<image src="'');

	//单击 A 改变 B 并禁用 AB
	var btn_A = document.getElementById('A');
	var btn_B = document.getElementById('B');

	btn_A.onclick = function(){
		//btn_B.setAttribute('text','cc');
		btn_B.innerHTML = "loading...";
		btn_A.setAttribute('disabled','false');
		btn_B.setAttribute("disabled",'false');
	}

</script>*/

//正则表达式匹配邮箱和 URL
$email = 'qishouzhang@marchsoft.cn.com';
$url = 'http://baidu.com.cn';
$grep_email = '/^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]+)+$/';
var_dump(preg_match($grep_email,$email,$matches));
var_dump($matches[0]);
$grep_url = '/^http:\/\/[\w]+(\.[\w]+)+[\S]*/';
var_dump(preg_match($grep_url,$url,$matches));
var_dump($matches[0]);


/*============================== 汇 通 天 下 物 流 ==================================*/
//写出你常用的 Linux 命令
//ls , cat ,vi ,mv ,cp , rm, chmod, find , ps, top,sudu
clear ：清屏命令
```

文件管理命令：

```
1   mkdir：创建一个目录     如：mkdir      csvtpy
2   rm：删除文件
     rmdir：删除目录
3   cp：复制文件      如：cp  1.py 12.py       //把 1.py 文件复制为
12.py 文件
4    mv：修改文件名     如：mv 1.py  11.py    //把 1.py 文件名改为
11.py 文件名
5    head：显示文件头部
        用法：head 【选项】【文件(一个或多个)】
        主要选项：-n：显示文件头 n 行内容
        例如：head -n 2 2 .py     //显示 2.py 文件的前 2 行内容
6    tail：显示文件尾部     用法同 head
7    ls：显示当前文件名
8    cat：显示文本文件的全部内容
9    less：分屏显示文件内容，就是另开一个窗口显示文件内容
10  pwd：显示当前目录
11  rz：上传文件命令
12  sz：下载文件命令 sz filename   (注意 sz，rz 只是对应文件，而不是
文件夹)
13  tar：压缩或解压文件或文件夹     如：tar -zpcv -f
/root/test.tar.gz  test        //将 test 文件夹压缩为
test.tar.gz，       再如：tar -zpcv -f test1.tar.gz  test1  //将文
件 test1 压缩   解压如：tar -zxv -f test2.tar.gz -C asd   //将
test2.tar.gz 解压到 asd 文件夹下
```

Vim 文本编辑器命令：

```
1    vim 文件名（1.py）：全屏幕进入 vim 文本编辑器，编辑 1.py 文件，如
果没有 1.py 就会创建进入 1.py
2    从 vim 切换到终端：
    ESC 键（使 vim 进入命令模式）
    :w      将缓冲区写入文件，即保存修改
    :wq      保存修改并退出
    :x      保存修改并退出
    :q      退出，如果对缓冲区进行过修改，则会提示
    :q!      强制退出，放弃修改
3    修改文本命令：
    i：在当前位置的字符之前进入插入模式
 //   a：在当前位置的字符之后进入插入模式
 //   c：更改当前位置的字符并进入插入模式键入替换字符
```

```
    o：在当前行的下方另起一行以插入文本
    cw：删除当前字且光标之后的字符，进入插入模式
    dd：删除当前行。可以使用重复 计数删除多行
    x：删除光标位置的字符，可以使用重复 计数删除多行
    p：将最后一个删除文本或字符放在当前字符处
    xp：交换光标位置的字符和它右边的字符
4    编辑/切换多个文件
    vim 一次性打开多个文件      如：vim 1.py 2.py
    ：e 文件名    切换文件 如：：e 2.py
5    在 vim 编辑器中与 shell 交互
    使用 ：！命令：直接在！后面输入所要执行的命令
    如：：！ hostname     显示当前主机名
6  用 vim 给文件加密解密
    添加密码：
    使用 vim 进入文件后，输入：X（注意是大写），然后敲回车，这时系统会提示输入密码
    输入密码：*******
    请再输入一次：*******
    保存后退出，这个文件就已经加密
    用 cat 或 more 查看文件内容，显示为乱码；用 vim/vi 重新编辑这个文件，会提示输入密码，如果输入的密码不正确，同样会显示为乱码
    解除密码：
    1    用 vim/vi 打开文件如 text.txt，要输入正确的密码，然后在编辑时，将密码设置为空，方法是输入下面的命令：      ：set key=    然后直接回车，：wq 保存文件后，文件已经解密了。
    2  或者这样也行：在正确打开文件后用“:X”指令，然后给一个空密码也可以。保存用“wq!”保存。
    两种方法实际上效果是一样的

linux 的一些快捷键：
    Ctrl+Alt+F2~F6：从图形界面转换到终端
    Ctrl+Alt+F1：从终端转到图形界面
    Ctrl+Alt+F8：用户（系统）锁定
```

```
//写出常用的魔术方法
//__construct(),__destruct(),__call(),__isset(),__get(),__clone(),__sleep(),__weakup()

//多服务器如何实现 session 共享？
//答：存数据库(当然还有其他的方式上)

//单击按钮 A 修改按钮 B 的显示为 loading...且禁用按钮 A 和按钮 B
/*var btnA = document.getElementById('A');
```

```
var btnB = document.getElementById('B');
btnA.onclick = function(){
    btnB.innerHTML = "loading";
    btnA.setAttribute('disabled','true');
    btnB.setAttribute('disabled','true');
}*/

//正则表达式匹配邮箱
preg_match('/^[a-zA-Z0-9_-]+@[a-zA-Z_-]+(\.[a-zA-Z0-9_-]+)+$/',$email,$matches);
var_dump($matches[0]);

//include 和 require 的区别
//1.include 包含的文件如果出错了，给出一个警告但不会终止脚本
//2.require 包含的文件如果出错了，给出一个致命错误会终止脚本
//3.include 什么时候使用什么时候引入，require 通常在文件的开始处
//4.require 不能放到 if 条件中，include 可以放到 if 条件中

//MVC 的优缺点
//优点：
//1.快速开发
//2.分工明确
//3.松耦合，重用性高
//缺点：
//1.需要维护大量的文件
//2.增加了结构的复杂性

//最后一题：N 个数，写一个高效的算法判断是否存在两个数的和是 P
function check($array,$p){
    $length = count($array);
    for($i=0;$i<$length;$i++){
        echo "a";
        for($j=$i+1;$i<$length;$j++){
            if($array[$i]+$array[$j] == $p){
                echo "存在！ $p = {$array[$i]}+ {$array[$j]}<br/>";
                return ;
            }
        }
    }
    echo "不存在！ <br/>";
}
$array = range(1,10);
shuffle($array);
//var_dump($array);
//check($array,112);
```

```
/*==============================壹号车==================================*/
//斐波那契数列的递归和非递归实现
//斐波那契数列-递归
function feibo($n){
	if($n<0)return;
	if($n<2)return $n;
	for($i=2;$i<=$n;$i++){
		return feibo($n-1)+feibo($n-2);
	}
}
var_dump(feibo(6));

//斐波那契数列-非递归
function feibo2($n){
	if($n<0)return;
	if($n<2)return $n;
	$x=0;$y=1;$z=0;
	for($i=2;$i<=$n;$i++){
		$z = $x+$y;
		$x = $y;
		$y = $z;
	}

	return $z;
}
var_dump(feibo2(6));

//一组数字要求奇数在前偶数在后
function adjust(&$array){
	$length = count($array);
	$pre = 0;
	$end = $length-1;
	while($pre<$end){
		while($pre<$end && $array[$pre]%2!=0){
			$pre++;
		}
		while($pre<$end && $array[$end]%2==0){
			$end--;
		}
		list($array[$pre],$array[$end]) = array($array[$end],$array[$pre]);
	}
}
```

```
$array = range(1,10);
shuffle($array);

//var_dump($array);
//adjust($array);
//var_dump($array);

//多维数组取值的其他方式例如;$array[a][b][c] => get_array_value($array,$path)
function get_array_value($array,$path){
	$path = explode(',',$path);
	$length = count($path);
	if($length<=0)return;
	$tmp = array();
	for($i=1;$i<$length;$i++){
		$tmp = $array[$path[$i-1]]; //上一维度
		$tmp = $tmp[$path[$i]]; //当前维度
		$array[$path[$i]] = $tmp; //当前维度用作下一次循环的开始
	}
	return $tmp;
}
$array = array('a'=>array('b'=>array('c'=>array('d'=>'This is a test!'),'c1'=>'c1')));
$path = 'a,b,c,d';
var_dump(get_array_value($array,$path));

//随机生成 100 个元素，要求不能重复，尽可能高效
function insert_value(){
	$array = array_fill(0,100,0);
	for($i=0;$i<100;$i++){
		$val = rand($i,100);//key 生成的值
		while(in_array($val,$array)){
			$val = rand(1,100);//key 生成的值
		}
		$array[$i] = $val;
	}
	return $array;
}
//var_dump(insert_value());


/*==============================360==================================*/
//1.PHP 的构造方法和析构方法是什么，如何显示的调用析构方法
//__construct() ;构造方法用来初始化对象
//__destruct():对象销毁前进入该方法
//用 unset():显示的调用__destruct()析构方法
```

//2.PHP 打印出前一天的
echo date('Y-m-d H:i:s',strtotime('-1 day'))."
";

//3.以下打印出什么
$a1 = null;
$a2 = false;
$a3 = 0;

PHP 面试题汇总
PHP
1、写出五种以上你使用过的 PHP 的扩展的名称（提示：常用的 PHP 扩展）
mysql、gd2、pdo、curl、mbstring、soap 等，在 php.ini 中可以找到。尽量多了解一些扩展，了解他们的功能（能做什么)。
php 通过使用 php_ming 库（Ming 库）快速生成 Flash 动画

1. 了解 MVC 模式吗？请写出三种以上目前 PHP 流行的 MVC 框架名称（不区分大小写）
MVC 是 Model（模型）、View(视觉）、Controll（控制器）的缩写。
MVC(Model-View-Controller)介绍
模型(Model): 应用程序的模型部分关心的是欲显示的数据的细节。模型通常关注的是应用程序的业务逻辑部分，关注的是如何使用数据库来读取和保存数据。
视图(View): 视图关心的是用户显示的部分，它通常是 HTML。
控制器(Controller)：控制器将特定的模型和视图结合起来，保证将正确的数据显示到页面上。
常用的 MVC 框架：
Zend Framework
FleaPHP

qeephp
CakePHP
ThinkPHP
CI

YII
大家有时间可以对框架进入多一些的深入了解。

3、用 PHP 打印出前一天的时间格式是 2008-2-8 18:00:10

echo “昨天:”, date(“Y-m-d H:i:s”, strtotime(“-1 day”)), “
”;

echo “昨天:”, date(“Y-m-d H:i:s”, mktime (date(“H”), date(“i”), date(“s”), date(“m”), date(“d”)-1, date(“Y”))), “
”;

date('Y-m-d H:i:s',Time () -24*3600)

参考 strtotime.php mktime.php

4、echo(),print(),print_r()的区别 var_dump()
echo 与 print:
它们都不是真正的函数，是一种语法结构（也有说 print 是函数，echo 不是）。

echo 和 print 后面都可不用加(),如: echo ‘ok’; print ‘ok’;
运行速度 echo 稍快一些，因为 echo 并不返回值，print 返回一个值 int(1)。
结论：
1、一般用 echo，除非三元运算时。$a=5; ($a==5) ? print ’5′ : print 0;
2、echo 后一般不要跟()。
print_r 是递归打印，主要用于输出数组对象。

2. print 只能有一个参数，所以不能不能用”,”,而 echo 可以。

3. Sprintf 以一定的格式 格式化一个字符串
参考 echo_print_print_r.php

5、能够使 HTML 和 PHP 分离开使用的模板
其实 PHP 本身就是一种模版引擎。参考 require.php

常用的模板引擎：smarty，还有 PHPLib,FastTemplate,Savant 等。
模板引擎列表：http://www.sitepoint.com/forums/showthread.php?t=123769

6、如何实现字符串翻转？
可用内置函数 strrev。如果不准用 PHP 内置函数的就自己写:

参考 strrev2.php

$STR = 'abc';

$STR{0}

Hello->olleH

7、$a = “hello”;

$b = &$a;

unset($b);

$b = “world”;

what is $a?
参考 references.php

此题的目的是要深刻理解引用。

通常，在将一个变量的值赋给另外一个变量的时候，先产生原变量的一个副本，然后再将它保存在内存的其他地方。如:
$a = 5;

$b = $a;
首先产生$a 的一个副本，然后再将它保存到$b 中。如果随后改变$a 的值，$b 的值不会改变：

$a = 7; //这时$b 仍然是 5
可以使用引用操作符&来避免这样的副本。如：
$a = 5;
$b = &$a;
$b = 7; //这时$a 和$b 都会是 7。 这行也可以换成$b = 7;
引用是非常有趣的。请记住，引用就像一个别名，而不是一个指针。$a 和$b 都指向了内存的相同地址。可以通过重置它们来改变所指向的地址。如下所示：
unset($a);
重置并不会改变$b 的值，但是只可以破坏$a 和值 7 保存在内存中的连接。

可以搜索： 引用 site:php.net 进一步深入了解 PHP 的引用。
也可以直接访问这个页面：http://www.php.net/manual/zh/language.references.php

http://www.php.net/manual/zh/language.references.unset.php

当 unset 一个引用，只是断开了变量名和变量内容之间的绑定。这并不意味着变量内容被销毁了。例如：

```
<?php
$a = 1;
$b =& $a;
unset($a);
?>
```

不会 unset $b，只是 $a。

8、实现中文字串截取无乱码的方法。
a.可以用正则
b.用 mb_substr()

参考 mb_substr.php

北京 abc substr

9、$a = 1;

$x = &$a;

$b = $a++;

what is $b? $x?

参考 references2.php

10、$array = array();
$x = empty($array);

what is $x? true or false
参考 empty.php

什么样的内容为空？

empty()、isset()、is_null
11、用 PHP 写出显示客户端 IP 与服务器 IP 的代码

得到服务器端的 IP：
//gethostbyname() 参考 gethostbyname.php 有时候得不到。
$_SERVER['SERVER_ADDR'];
得到客户端的 IP：
$_SERVER['REMOTE_ADDR'];
参考 ip.php

12、某内容管理系统：用户提交内容后，系统生成静态 HTML 页面；写出实现的基本思路，最好写出相关代码。

用户提交内容后，将内容加在最终页面模板上，然后另存为 HTML 页面（创建 HTML 页面，将内容和页面模板写入）。
代码：

```
$tpl->assign('vars', $vars);
$static_html = = $tpl->fetch（'tpl/index.html'）; //Smarty 的 fetch 方法
$fp = fopen('html/index.html', 'w');

fwrite($fp, $static_html);
```

另一种 ob_start

createHtml.php

13、写出以下程序的输出结果

```
$b=201;

$c=40;

$a=$b>$c?4:5;

echo $a;
```

参考 3yuan.php

14、写出以下程序的输出结果

```
$str="cd";

$$str="hotdog"; // $cd = 'hotdog'

$$str.="ok";

echo $cd;
```

参考$$.php

15、在 PHP 中 error_reporting 这个函数有什么作用？
设定 php 脚本的错误报告级别

error_reporting(6143)的作用是设定 php 脚本的错误报告级别为”所有错误”。
ini_set(‘display_errors’, 1); //作用是在显示 PHP 脚本错误，相当于修改 php.ini 中的

error_reporting

定义和用法

error_reporting() 设置 PHP 的报错级别并返回当前级别。

语法

error_reporting(report_level)

如果参数 level 未指定，当前报错级别将被返回。下面几项是 level 可能的值：

值 常量 描述
1 E_ERROR Fatal run-time errors. Errors that can not be recovered from. Execution of the script is halted
2 E_WARNING Non-fatal run-time errors. Execution of the script is not halted
4 E_PARSE Compile-time parse errors. Parse errors should only be generated by the parser
8 E_NOTICE Run-time notices. The script found something that might be an error, but could also happen when running a script normally
16 E_CORE_ERROR Fatal errors at PHP startup. This is like an E_ERROR in the PHP core
32 E_CORE_WARNING Non-fatal errors at PHP startup. This is like an E_WARNING in the PHP core
64 E_COMPILE_ERROR Fatal compile-time errors. This is like an E_ERROR generated by the Zend Scripting Engine
128 E_COMPILE_WARNING Non-fatal compile-time errors. This is like an E_WARNING generated by the Zend Scripting Engine
256 E_USER_ERROR Fatal user-generated error. This is like an E_ERROR set by the programmer using the PHP function trigger_error()
512 E_USER_WARNING Non-fatal user-generated warning. This is like an E_WARNING set by the programmer using the PHP function trigger_error()
1024 E_USER_NOTICE User-generated notice. This is like an E_NOTICE set by the programmer using the PHP function trigger_error()
2048 E_STRICT Run-time notices. PHP suggest changes to your code to help interoperability and compatibility of the code
4096 E_RECOVERABLE_ERROR Catchable fatal error. This is like an E_ERROR but can be caught by a user defined handle (see also set_error_handler())
8191 E_ALL All errors and warnings, except level E_STRICT (E_STRICT will be part of E_ALL as of PHP 6.0)
例子

任意数目的以上选项都可以用”或”来连接（用 OR 或 |），这样可以报告所有需要的各级别错误。例如，下面的代码关闭了用户自定义的错误和警告，执行了某些操作，然后恢复到原始的

报错级别：

```
<?php

//禁用错误报告

error_reporting(0);

//报告运行时错误

error_reporting(E_ERROR | E_WARNING | E_PARSE);

//报告所有错误

error_reporting(E_ALL);

?>
```

16、简述如何得到当前执行脚本路径，包括所得到参数。
访问 http://temp.com/phpinfo.php?id=1
echo $_SERVER['SCRIPT_URL']; //得到/phpinfo.php

echo $_SERVER["SCRIPT_URI"]; //得到 http://temp.com/phpinfo.php

echo $_SERVER["SCRIPT_FILENAME"]; //得到 F:/www/Temp/phpinfo.php

echo $_SERVER["REQUEST_URI"]; //得到/phpinfo.php?id=1

echo $_SERVER["SCRIPT_NAME"]; //得到/phpinfo.php

参考 server.php http://lesson.com/test/server.php?id=1

17、有一个网页地址 http://bbs.91lamp.com/index.php ,如何得到它的 html 内容
file_get_contents()

4. T 数组函数 arsort 的作用是__。

对数组进行逆向排序并保持索引关系

/*

rsort — 对数组逆向排序

sort — 对数组排序

*/
参考 arsort.php

19、执行程序段<?php echo 8%(-2) ?>将输出__0。
参考%.php
20、语句 include 和 require 都能把另外一个文件包含到当前文件中，它们的区别是__；为了避免多次包含同一文件，可以用语句__来代替它们。
发生异常时 include 产生警告，程序继续执行；require 产生致命错误，程序停止往下执行。
一般推荐使用 require（更能调试错误）。
require_once()/include_once()

require 重复调用会多次加载你引用的文件；require_once 只加载一次，而不管你实际上调用了多少次，主要用于复杂的文件包含关系。
例如 b 包含 a，c 包含 a，但同时 c 又包含了 b，那么如果用 require 的话可能会导致两次加载 a，这时应使用 require_once。

实际开发过程中：如果确定某个文件只会被包含一次，那么用 require，否则用 require_once。
因为 require 不需要检测文件是否被包含过，比 require_once 的执行效率要高。

21、一个函数的参数不能是对变量的引用，除非在 php.ini 中把__设为 on。
allow_call_time_pass_reference
quote.php

22、在 PHP 中，heredoc 是一种特殊的字符串，它的结束标志必须__。
结束标识符所在的行不能包含任何其它字符除”;”

参考 heredoc.php

23、有一数组 $a=array(3,2,4,9,8);请将其重新排序，按从小到大的顺序列出。
可用冒泡法进行排序：
冒泡排序的基本概念是：依次比较相邻的两个数，将小数放在前面，大数放在后面。即首先

比较第 1 个和第 2 个数，将小数放前，大数放后。然后比较第 2 个数和第 3 个数，将小数放前，大数放

后，如此继续，直至比较最后两个数，将小数放前，大数放后，此时第一趟结束，在最后的数必是所有数中的最大数。重复以上过程， 仍从第一对数开始比较（因为可能由于第 2 个数和第 3 个

数的交换，使得第 1 个数不再小于第 2 个数），将小数放前，大数放后，一直比较到最大数前的一对相邻 数，将小数放前，大数放后，第二趟结束，在倒数第二个数中得到一个新的最大数。如

此下去，直至最终完成排序。

参考 array1.php array2.php

24、写出 session 的运行机制。

session 创建时，是否会在服务端记录一个 cookie?cookie 里面的内容是什么？

session 机制是一种服务器端的机制，服务器使用一种类似于散列表的结构（也可能就是使用散列表）来保存信息。

当 程序需要为某个客户端的请求创建一个 session 的时候，服务器首先检查这个客户端的请求里是否已包含了一个 session 标识-称为 sessionid，如果已包含一个 sessionid 则说明以前已经为

此客户端创建过 session，服务器就按照 sessionid 把这个 session 检索出来使用（如果检索不到，可能会新建一个），如果客户端请求不包含 sessionid，则为此客户端创建一个 session 并且

生成一 个与此 session 相关联的 sessionid，sessionid 的值应该是一个既不会重复，又不容易被找到规律以仿造的字符串，这个 sessionid 将被在本次响应中返回给客户端保存。

保存这个 sessionid 的方式可以采用 cookie，这样在交互过程中浏览器可以自动的按照规则把这个标识发给服务器。一般这个 cookie 的名字都是类似于 SEEESIONID。

由 于 cookie 可以被人为的禁止，必须有其他机制以便在 cookie 被禁止时仍然能够把 sessionid 传递回服务器。经常被使用的一种技术叫做 URL 重写，就是把 sessionid 直接附加在 URL 路径的后

面，附加方式也有两种，一种是作为 URL 路径的附加信息，表现形式为 http://…..

/xxx;SEEESIONID=ByOK3vjFD75aPnrF7C2HmdnV6QZcEbzWoWiBYEnLerjQ99zWpBng!-145788764

另一种是作为查询字符串附加在 URL 后面，表现形式为 http:// … ../xxx?SEEESIONID=ByOK3vjFD75aPnrF7C2HmdnV6QZcEbzWoWiBYEnLerjQ99zWpBng!-145788764

为了在整个交互过程中始终保持状态，就必须在每个客户端可能请求的路径后面都包含这个 SEEESIONID。

参考：
session 运行机制:理解 session 机制：
http://bbs.91lamp.com/detail-526-1.html
抛开 cookie 使用 session：
http://www.91lamp.com/html/document/php/200808/16-1800.html
cookie 与 session：
http://www.91lamp.com/html/document/php/200808/16-1797.html

25、Cookie 的原理及使用？

Coøkie 是网站保存在浏览器客户端的信息，也就是说保存在访客的机器里的变量，一般随着 HTTP 头发送到服务器端。在 Cookie 生效之后及失效之前，客户每次发 出页面请求的时候（包括 PHP

页面和静态 html 页面），都会把 Cookie 一块发送到服务器，只要我们针对它进行相应的处理，就可以实现变量”追随”。

cookie 可以跨越子域名。
比如我们在 xiaofeicn.com 下面注册个个 cookie，那么可以在 bbs.xiaofeicn.com 上读取到该 cookie。
session 不可以跨越子域名：
比如我们在 xiaofeicn.com 下面注册个个 session，那么不可以在 bbs.xiaofeicn.com,www.xiaofeicn.com 上读取到该 session。

a． 设置一个 Cookie 变量

设置一个 Cookie 变量，PHP 使用的函数是： int setcookie(string name, string value, int expire,

string path, string domain, int secure);

其中 name 是 Cookie 变量名称标识，你在 PHP 中将可以象使用普通变量名一样来用它引用 Cookie 变量。value 是 Cookie 变量的初始 值，expire 表示该 Cookie 变量的有效时间；path 为该

Cookie 变量的相关路径；domain 表示 Cookie 变量的网站；secure 则需在 https 的安全传输时才有效。 例如我们要设置一个变量 username，它的值是字符串”bluewind”，我们可以这

么写代码： setcookie (“username”,”bluewind”); //这两个参数是 setcookie 必要的。

我们还想给这个变量设置有效时间来限制操作超时等，比如说 10 分钟： setcookie (“username”,”bluewind”, 600000); //有效时间的单位是毫秒。

注意：setcookie 和 header 函数一样，需要放在任何能向客户端输出的语句之前。

b． 销毁一个变量

销毁 Cookie 变量只要将它的 value 设为空（””）就可以了，如想销毁上面那个变量只要再写一次： setcookie (“username” ,””);

就可以了。这常用作安全退出之用。

c． Cookie 的有效范围和生存期

Cookie 的有效范围（也就是说在这个范围的页面都能得到这个 Cookie 变量）默认的是该目录及其子目录，当然你可以用 setcookie 的 path 和 domain 参数进行修改。如果你不对 cookie 的

expire 进行设置（参见 1. 设置一个 Cookie 变量中的例子），那么当你离开网站的页面，cookie 也同时得到自动销毁。 http://www.netscape.com/newsref/std/cookie_spec.html 是

cookie 原创者 Netscape 所提供的完整介绍信息。

26、PHP 的意思（英文全称、含义）

php 是 Hypertext Preprocessor 的缩写，php 是一种内嵌 HTML 的脚本语言。PHP 的独特语法混合了 c,java 和 perl 及 PHP 式的新语法。这门语言的的目标是让网页开发人员快速的写出动态的网页。

Personal HomePage tools

Hypertext Preprocessor Hypertext Preprocessor …

27、foo()和@foo()之间有什么区别?
foo() 会执行这个函式，任何解译错误、语法错误、执行错误都会在页面上显示出来。

@foo() 在执行这个函数时，会隐藏所有上述的错误讯息。

很多应用程式都使用 @mysql_connect() 和 @mysql_query 来隐藏 mysql 的错误讯息，我认为这是很严重的失误，因为错误不该被隐藏，你必须妥善处理它们，可能的话解决它们。

28、如何声明一个名为”myclass”的没有方法和属性的类?

class myclass

{

}

29、如何实例化一个名为”myclass”的对象?

$myclass = new myclass;

30、你如何访问和设置一个类的属性?

$myclass->username = ‘andy’;

31、GD 库是做什么用的?
GD 函式库用来做什么？

这 个可能是我最喜欢的函式库，自从 PHP 4.3.0 版本后 GD 便内建在 PHP 系统中。这个函式库让你处理和显示各式格式的图档，它的另一个常见用途是制作所图档。GD 以外的另一个选择

是 ImageMagick，但这个函式库并不内建于 PHP 之中，必须由系统管理员安装在伺服器上。MagickWand

32、指出一些在 PHP 输出一段 HTML 代码的办法。
嗯，你可以使用 PHP 中任何一种输出语句，包括 echo、print、printf，大部分人都使用如下例的 echo：

echo “My string $variable”;

你也可以使用这种方法：
1
Heredoc

echo <<<END
This text is written to the screen as output and this $variable is parsed too. If you wanted you can have HTML tags in here as well. The END; remarks

must be on a line of its own, and can’t contain any extra white space.
END;

33、下面哪个函数可以打开一个文件，并对文件进行读和写操作? c

(a) fget() (b) file_open() (c) fopen() (d) open_file()

34、下面哪个选项没有将 john 添加到 users 数组中? bd

(a) $users[] = ‘john’;

(b) array_add($users, ‘john’);

(c) array_push($users, ‘john’, ‘andy’);

(d) $users ||= ‘john’;

35、

如何使用下面的类,并解释下面什么意思?

class Mymd5

{

function get_md5($str)

{

```
            $str=md5(md5($str).”xingmo”);

            return $str;

        }

}
```

参考 mymd5.php

36、用哪一个函数检测一个变量是否定义过?是否为空的函数是?是否为 NULL?

isset()、 empey、is_null()

要深刻理解这几个的含义。

37、$arr = array(‘james’, ‘tom’, ‘symfony’); 请打印出第一个元素的值 ac

a.echo $arr[0];

b.echo $arr{1};

c. $arr2 = array_shift($arr); echo $arr2;
array_pop

38、请将数组的值用’,’号分隔并合并成字符串输出。如何将一个以’,’隔开的字符串分割成数组?
参考 implode.php 把数组变成字符串
要掌握 implode 和 explode 的用法。

39、$a = ‘abcdef’; 请打印出$a 的第一个字母。
echo $a{0};
echo $a[0]; // 不建议用这种方式

substr($a, 0, 1);

最好是用{}。

40、PHP 可以和 sql server/oracle 等数据库连接吗?

可以。可以用 PDO 连接。

41、请写出 php5 的构造函数和析构函数

```
function __construct()

{
}
function __destruct()

{

}
```

42、写一个函数，尽可能高效的，从一个标准 url 里取出文件的扩展名

例如: http://www.sina.com.cn/abc/de/fg.php?id=1 需要取出 php 或 .php
参考 url1.php

43、求两个日期的差数，例如 2007-2-5 ~ 2007-3-6 的日期差数（天数）。
思路 1：先用 strtotime 转换成 unix 时间戳，然后相减，除以 86400.
思路 2：先用 mktime 转换成 unix 时间戳，然后相减，除以 86400.

参考 time1.php

44、请写一个函数，实现以下功能：
字符串”open_door” 转换成 “OpenDoor”、”make_by_id” 转换成 “MakeById”。

思路：
1)将’_’替换成’ ‘;
2)使用 ucwords()将各单词首字母大写；
3)去掉空格;str_replace()

rename_for_val.php

46、如果模板是用 smarty 模板。怎样用 section 语句来显示一个名为$data 的数组。比如：

```
$data = array(
        0 => array( ‘id’=>8, ‘name’=>’name1′ ),
        1 => array( ‘id’=>10, ‘name’=>’name2′ ),
        2 => array( [id]=15 [name]=’name3′ )
)
```

写出在模板页的代码？ 若用 foreach 语句又要怎样显示呢？
section 和 foreach 循环二维数组的基本功。section 和 foreach 循环一维数组？ section 和 foreach 循环三维数组？

Section 句式

```
{section name=customer loop=$data }
id: {$data[customer]}<br />
{/section}
```

47、不用新变量直接交换现有两个变理的值。
考算法的基本功。

```
$a = ‘welcome’;

$b = ‘beijing’;

$a = ‘|’.$a.’|’.’|’.$b.’|’; //|welcome||beijing|
$b = str_replace( ‘|’.$b.’|’, ”, $a);

$b = trim($b, ‘|’);


$a = str_replace( ‘|’.$b.’|’, ”, $a);

$a = trim($a, ‘|’);

echo $a;
```

echo $b;

$a=5;$b=3;怎么交换两个数字？

$a=$a+$b; // 8

$b=$a-$b; //5

$a=$a-$b; //3

5. PHP 数字金额转中文大写格式，同时说明思路(考数组掌握)。
15001.25 壹万伍仟零壹元贰角伍分

$daixi = array (‘零’,’壹’,”);

$str{0}

6. 写一个函数，能够遍历一个文件夹下的所有文件和子文件夹。
Readdir opendir scandir

参考 dir.php

50、表单中 get 与 post 提交方法的区别?

a、 Get 方法通过 URL 请求来传递用户的数据，将表单内各字段名称与其内容，以成对的字符串连接，置于 action 属性所指程序的 url 后，如 http://www.domain.com/test.asp?

name=51js&password=51js，数据都会直接显示在 url 上，就像用户点击一个链接一样；Post 方法通过 HTTP post 机制，将表单内各字段名称与其内容放置在 HTML 表头(header)内一起传

送给服务器端交由 action 属性能所指的程序处理，该程序会通过标准输入(stdin)方式，将表单的数据读出并加以处理

b、 Get 方式需要使用 $_GET 来取得变量的值；而 Post 方式通过 $_POST 来访问提交的内容

c、 Get 方式传输的数据量非常小，一般限制在 2 KB 左右，但是执行效率却比 Post 方法

好；而 Post 方式传递的数据量相对较大，它是等待服务器来读取数据，不过也有字节限制，这是

为了避免对服务器用大量数据进行恶意攻击。可在 php.in 中 对 post_max_size 进行设置。

建议：除非你肯定你提交的数据可以一次性提交，否则请尽量用 Post 方法

d、 Get 方式提交数据，会带来安全问题，比如一个登陆页面，通过 Get 方式提交数据时，用户名和密码将出现在 URL 上，如果页面可以被缓存或者其他人可以访问客户这台机器，就可以

从历史记录获得该用户的帐号和密码，所以表单提交建议使用 Post 方法；Post 方法提交的表单页面常见的问题是，该页面如果刷新的时候，会弹出一个对话框。

建 议 ： 出 于 安 全 性 考 虑 ， 建 议 最 好 使 用 Post 提 交 数 据 **********在 B/S 应用程序中，前台与后台的数据交互，都是通过 HTML 中 Form 表单完成的。Form 提供了两

种数据传输的方式——get 和 post。虽然它们都是数据的提交方式，但是在实际传输时确有很大的不同，并且可能会 对数据产生严重的影响。虽然为了方便的得到变量值，Web 容器已经屏蔽了

二者的一些差异，但是了解二者的差异在以后的编程也会很有帮助的。

Form 中的 get 和 post 方法，在数据传输过程中分别对应了 HTTP 协议中的 GET 和 POST 方法。二者主要区别如下：

a、Get 是用来从服务器上获得数据，而 Post 是用来向服务器上传递数据。

b、 Get 将表单中数据的按照 variable=value 的形式，添加到 action 所指向的 URL 后面，并且两者使用”?”连接，而各个变量之间使用”& amp;”连接；Post 是将表单中的数据放在 form 的数据

体中，按照变量和值相对应的方式，传递到 action 所指向 URL。

c、 Get 是不安全的，因为在传输过程，数据被放在请求的 URL 中，而如今现有的很多服务器、代理服务器或者用户代理都会将请求 URL 记录到日志文件中，然后 放在某个地方，这样就可能会

有一些隐私的信息被第三方看到。另外，用户也可以在浏览器上直接看到提交的数据，一些系统内部消息将会一同显示在用户面前。 Post 的所有操作对用户来说都是不可见的。

d、Get 传输的数据量小，这主要是因为受 URL 长度限制；而 Post 可以传输大量的数据，所以在上传文件只能使用 Post（当然还有一个原因，将在后面的提到）。

e、Get 限制 Form 表单的数据集的值必须为 ASCII 字符；而 Post 支持整个 ISO10646 字符集。

f、Get 是 Form 的默认方法。

*.Post 传输数据时，不需要在 URL 中显示出来，而 Get 方法要在 URL 中显示。

*.Post 传输的数据量大，可以达到 2M，而 Get 方法由于受到 URL 长度的限制,只能传递大约 1024 字节.

*.Post 顾名思义,就是为了将数据传送到服务器段,Get 就是为了从服务器段取得数据.而 Get 之所以也能传送数据,只是用来设计告诉服务器,你到底需要什么样的数据.Post 的信息作为 http 请求

的内容，而 Get 是在 Http 头部传输的。

51、session 与 cookie 的区别?

session 是服务器端缓存，cookie 是客户端缓存。

cookie 机制采用的是在客户端保持状态的方案，而 session 机制采用的是在服务器端保持状态的方案。

52、PHP 支持的数据类型有八种,以下被支持的有：

string

int

float

Bool

array

NULL

resource

object

A、array

B、floating-point numbers(double)

C、integer

D、date

E、string

[A B C E]

PHP 的变量属于松散数据类型，在计算时动态(dynamic)决定。如果要强制设置变量的数据类型的话，可以利用 settype()

函数。或利用 c 语言的强制转型方式(type casting)。

53、假定要使用 Apache+Php 的配置，并将 php3 编译成 Apache 的一个模块。那麼以下 httpd.conf 文件的语句是必须的：【C】

A、AddModule mod_php3.c

B、LoadModule php3_module libexec/libphp3.so

C、AddType application/x-httpd-php3 .php3

D、setup

E、make install

54、以下程序:

<HTML>

<HEAD>

```
<TITLE></TITLE>

<HEAD>

<BODY>

<?php

$num1 = 15;

$num2 = $num1;

echo “<p>$num2</p>”;

$num2 = &$num1;

$num2 = 20;

echo “<p>$num1</p>”;

?>

</BODY>

</HTML>
```

程序输出为:[ ]

A、15

B、35

C、20

D、5

AC

55、以下程序

```
<?php
$str1 = “01″;
$str1++;
$str1 += 1; //$str1 = $str1 + 1;
echo “<p>\$str1 => $str1</p>”;
?>
```

程序输出为：[ ]

A、$str1 => 01

B、$str1 => 2

C、$str1 => 03

D、$str1 => 3

E、$str1 => 1

D

参考 math1.php

56、全局变量与局部变量

```
$a=1;
sum()
{

echo $a;
}
sum();
```

程序输出为：[ ]

A、1

B、10

C、100

D、1000

E、空值

E

57、PHP 的控制语句

```
<?php
$a = 3;
$b = $a++;
if ($a > $b)
{
echo “a 比 b 大”;
}
elseif ($a == $b)
{
echo “a 等于 b”;
}
else
{
```

echo “a 比 b 小”;

}

?>

输出结果为：[ ]

A、a 比 b 大

B、a 等于 b

C、a 比 b 小

D、”a 比 b 小”

E、无输出

A

58、PHP 对字符串的处理程序

$name=”Jollen”;

echo ‘Name:$name’;

echo “Name:$name”;

输出结果为：[ ]

A、Name:Jollen

Name:Jollen

B、Name:Jollen

Name:$name

C、Name:$name

Name:Jollen

D、Name:$name

Name:$name

E、Name:”Jollen”

Name:Jollen

C

此题考单引号与双引号的基本功。

59、下面建立与 MySQL Server 的连接语法正确的是：[ ]

A、$link=connect(“host_name”,”user_name”,”password”);

B、$link=mysql_connect(“host_name”,”user_name”,”password”);

C、$link=mysqlconnect(“host_name”,”user_name”,”password”);

D、$link=mysql_pconnect(“host_name”,”user_name”,”password”);

E、$link=pconnect(“host_name”,”user_name”,”password”);

BD

60、rawurlencode()的作用是？
按照 RFC 1738 对 URL 进行编码

返 回字符串，此字符串中除了 -_. 之外的所有非字母数字字符都将被替换成百分号（%）后跟两位十六进制数。这是在 RFC 1738 中描述的编码，是为了保护原义字符以免其被解释为特殊的

URL 定界符，同时保护 URL 格式以免其被传输媒体（像一些邮件系统）使用字符转换时弄乱。

与 urlencode()的区别：
urlencode:
返回字符串，此字符串中除了 -_. 之外的所有非字母数字字符都将被替换成百分号（%）后跟两位十六进制数，空格则编码为加号（+）。此编码与 WWW 表单 POST 数据的编码方式是一样的

，同时与 application/x-www-form-urlencoded 的媒体类型编码方式一样。由于历史原因，此编码在将空格编码为加号（+）方面与 RFC1738 编码（参见 rawurlencode()）不同。

61、请说明在 php.ini 中 safe_mode 开启之后对于 PHP 系统函数的影响?

safe_mode 是唯一 PHP_INI_SYSTEM 属性，必须通过 php.ini 或 httpd.conf 来设置。要启用 safe_mode，只需修改 php.ini： safe_mode = On 或者修改 httpd.conf，定义目录：

Options FollowSymLinks php_admin_value safe_mode 1

重启 apache 后 safe_mode 就生效了。启动 safe_mode，会对许多 PHP 函数进行限制，特别是和系统相关的文件打开、命令执行等函数。

默认情况下，所有操作文件的函数将只能操作与脚本 UID 相同的文件。

注意：如果在 linux 中启用了 safe_mode，那么如果要在一个目录中创建一个目录，比如要在/upload 中创建一个 20081202，那么/upload 目录所有者必须是 apache 的所有者。

62、PHP5 中魔术方法函数有哪几个，请举例说明各自的用法

__sleep
__wakeup
__toString
__set_state
__construct,
__destruct
__call,
__get,
__set,
__isset,
__unset
__clone
__autoload

7. What does <? echo count (“123″) ?> print out? D
A) 3
B) False

C) Null
D) 1
E) 0

65、What is the value of $a?
<?php
$a = in_array(’01′, array(’1′)) == var_dump(’01′ == 1);
?>
A) True
B) False

B

66、What is the value of $result in the following PHP code?
<?php
function timesTwo($int) {
$int = $int * 2;
}
$int = 2;
$result = timesTwo($int);
?>;
Answer: NULL

67、The code below _____ because _____.
<?php
class Foo {
?>
<?php
function bar() {
print “bar”;
}
}
?>
A) will work, class definitions can be split up into multiple PHP blocks.
B) will not work, class definitions must be in a single PHP block.
C) will not work, class definitions must be in a single file but can be in multiple PHP blocks.
D) will work, class definitions can be split up into multiple files and multiple PHP blocks.

68、When turned on, _____ will _____ your script with different variables from HTML forms and cookies. //D
A) show_errors, enable

B) show_errors, show
C) register_globals, enhance
D) register_globals, inject

69、What will be the output of the following PHP code:
<?php
echo count(strlen(“http://php.net“));
?>
Answer: 1

71、What is the difference between “print()” and “echo()”?

Answer: print is a function,echo is a language construct

72、写出以下程序的运行结果
$aa = null;
$bb = false;
If($aa= = = $bb)
{
    Echo ‘相同’;
}
Else
{
echo’不相同’;
}

8. zend optimizer 是什么
用优化代码的方法来提高 php 应用程序的执行速度，且可以解密 Zend Guard 加密过后代码，使之能够正常运行

74、有三个 php 文件位于同一目录下，内容为
a.php:——-
<?php function fa() { echo “in Function A\n”; }?>

b.php:——-
<?php include_once ‘a.php’; ?>
<?php function fb() { fa(); echo “in Function B\n”; } ?>

c.php:——-
<?php include ‘a.php’; ?>
<?php include ‘b.php’; ?>
<?php fa(); fb(); ?>

使用浏览器访问 c.php，请问是否存在问题。
如果存在问题，请指出修正方法并写出浏览器查看效果
如果不存在问题，请写出浏览器查看效果

9. 将字符 09 转换成十进制数字。
Intval.php

77、What would the following code print to the browser? Why?
复制内容到剪贴板代码:

```
$num = 10;
function multiply(){
$num = $num * 10;
}
multiply();
echo $num;
```

79、以下哪一个函式可以把浏览器转向到另一个页面？ (B)

A) redir()

这不是一个 PHP 函式，会引致执行错误。

B) header()

这个是正确答案，header() 用来插入卷头资料，可以用来使浏览器转向到另一个页面，例如：

header(“Location: http://www.search-this.com/”);

C) location()

这不是一个 PHP 函式，会引致执行错误。

D) redirect()

这不是一个 PHP 函式，会引致执行错误。

80、isset()和 empty()的区别

两者都是测试变量用的。但是 isset()是测试变量是否被赋值，而 empty()是测试一个已经被赋值的变量是否为空。 如果一个变量没被赋值就引用在 php 里是被允许的,但会有 notice 提示。

如果一个变量被赋空值，$foo=””或者$foo=0 或 者$foo=false,那么 empty($foo)返回真，isset($foo)也返回真，就是说赋空值不会注销一个变量。要注销一个变量，可以用 unset($foo)或

者$foo=NULL。

```
$a4 = '';
$a5 = '0';
$a6 = 'null';
$a7 = array();
$a8 = array(array());

echo empty($a1) ? 'true' : 'false';echo "<br/>"; //true
echo empty($a2) ? 'true' : 'false';echo "<br/>"; //true
echo empty($a3) ? 'true' : 'false';echo "<br/>"; //true
echo empty($a4) ? 'true' : 'false';echo "<br/>"; //true
echo empty($a5) ? 'true' : 'false';echo "<br/>"; //true
echo empty($a6) ? 'true' : 'false';echo "<br/>"; //false
echo empty($a7) ? 'true' : 'false';echo "<br/>"; //true;
echo empty($a8) ? 'true' : 'false';echo "<br/>"; //false;

//3.shell 脚本中获取当前命令的所有参数

//4.使用 awk 命令输出/etc/passwd 下的所有用户
//awk -F":" '{print $7}' /etc/passwd | uniq -c

//5.svn 对比 index.php 版本 1000 和 1001 的区别
//svn diff -r1000:1001

//6.请使用 vim 命令替换文本中所有的 hello 为 world;
//%s/hello/world/g

//7.定义 vim 快捷键 mm,一键删除所有行
//方法 1:       按 ggdG
//方法 2:       :%d

//10.写出程序的执行结果
error_reporting(E_ALL);
$a1 = array('1.0.3.1111','1.1.3.1111','2.0.3.1111','2.1.3.1111');
var_dump(in_array(1,$a1));    //true
var_dump(in_array('2',$a1));//false
var_dump(in_array(2,$a1,true)); //false
var_dump(in_array('1',$a1,true));//false
```

```
//11.写出程序的输出结果
$a1 = "1	a	3	d";
$b1 = explode('\t',$a1);
$b2 = explode("\t",$a1);
echo in_array('a',$b1) ? 1 : 0;echo "<br/>"; //0
echo in_array("d",$b2) ? 1 : 0;echo "<br/>";//1 ?不能明白为啥执行结果为：0
echo in_array("1",$b1) ? 1 : 0;echo "<br/>";//0
echo in_array(3,$b2) ? 1 : 0;echo "<br/>";//1 ?不明白为哈结果为：0

//12.请用原生 js 输出浏览器类型
//nagotive.userAgent;

//13.设计一个函数，打开一个页面并获取页面中所有的邮箱
function open_page($path,$grep){
	if(is_file($path)){
		$handle = file_get_contents($path);	//打开文件
		preg_match_all($grep,$handle,$matches);//过滤邮箱
		return $matches[0];//返回匹配的结果
	}
}
$path = "mymst.html";
$grep_email = '/[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]+)+/';
var_dump(open_page($path,$grep_email));

//14.遍历指定目录的所有文件
function get_all_files($path){
	$files = array();
	if(is_dir($path)){
		if($handle = opendir($path)){
			while($file = readdir($handle)){
				if($file!='.' && $file!='..'){
					$child_path = $path.'/'.$file;
					if(is_dir($child_path)){
						$files[$file] = get_all_files($child_path);
					}else{
						$files[] = $child_path;
					}
				}
			}
		}
		closedir($handle);
		return $files;
	}
```

```
}
//var_dump(get_all_files('f:'));

//14. 写一个最简单的 RBAC 中的所有数据表及关键字段
//1.用户表（user_id）
//2.角色表(role_id)
//3.权限表(node_id，action)
//4.用户角色表（id,role_id,user_id）
//5.角色权限关联表（id,role_id,node_id）

//15.HTTP 请求报文和应答报文
/*
请求头:
Accept: text/html,image(浏览器可以接收的类型)
Accept-Charset: ISO-8859-1(浏览器可以接收的编码类型)
Accept-Encoding: gzip,compress(浏览器可以接收压缩编码类型)
Accept-Language: en-us,zh-cn(浏览器可以接收的语言和国家类型)
Host: www.it315.org:80(浏览器请求的主机和端口)
If-Modified-Since: Tue, 11 Jul 2000 18:23:51 GMT(某个页面缓存时间)
Referer: http://www.it315.org/index.jsp(请求来自于哪个页面)
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)(浏览器相关信息)
Cookie：(浏览器暂存服务器发送的信息)
Connection: close(1.0)/Keep-Alive(1.1)(HTTP 请求的版本的特点)
Date: Tue, 11 Jul 2000 18:23:51 GMT(请求网站的时间)

响应头：
Location: http://www.it315.org/index.jsp(控制浏览器显示哪个页面)
Server:apache tomcat(服务器的类型)
Content-Encoding: gzip(服务器发送的压缩编码方式)
Content-Length: 80(服务器发送显示的字节码长度)
Content-Language: zh-cn(服务器发送内容的语言和国家名)
Content-Type: image/jpeg; charset=UTF-8(服务器发送内容的类型和编码类型)
Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT(服务器最后一次修改的时间)
Refresh: 1;url=http://www.it315.org(控制浏览器 1 秒钟后转发 URL 所指向的页面)
Content-Disposition: attachment; filename=aaa.jpg(服务器控制浏览器发下载方式打开文件)
Transfer-Encoding: chunked(服务器分块传递数据到客户端）
Set-Cookie:SS=Q0=5Lb_nQ; path=/search(服务器发送 Cookie 相关的信息)
Expires: -1(服务器控制浏览器不要缓存网页，默认是缓存)
Cache-Control: no-cache(服务器控制浏览器不要缓存网页)
Pragma: no-cache(服务器控制浏览器不要缓存网页)
Connection: close/Keep-Alive(HTTP 请求的版本的特点)
Date: Tue, 11 Jul 2000 18:23:51 GMT(响应网站的时间)
*/
```

```
//16.POST 和 GET 的区别，以及 POST 常用的数据封装格式
//post 和 get 的区别：
/*
1.get 请求的参数附加到 url 中，post 不会显示在 url 中
2.get 请求的参数有长度限制，post 没有长度限制
3.get 请求类型只能为 ASCII ,post d 对数据类型没有限制
4.get 请求可以被缓存可以存书签，post 请求不可以缓存不可以存书签
*/
//post 常用的数据封装格式
/*
1.
2.json
3.text/html
4.multipart/form-data
*/

//17.简述 SQL 注入与 XSS 攻击的原理
//SQL 注入：通过出传入非法参数，和 sql 语句拼接后，构成对数据库进行攻击的语句（SQL 注入攻击中以 SQL 语句作为用户输入，从而达到查询/修改/删除数据的目的）
//XSS（跨站脚本攻击）：将恶意代码植入到用户使用的页面中（通过插入恶意脚本，实现对用户游览器的控制）


/*============================== 豆 果 美 食 (*)====================================*/
//file()//返回函数
//eval() //把一个字符串当做代码执行

//统计数组中字符出现的次数，并按升序将统计后的元素进行排序
$array = array(4,6,3,'a','d',5,7,3,7,9,'e','s','c',9,'w','x',2,'x');
$tmp = array_count_values($array);
arsort($tmp);
//var_dump($tmp);

//计算两个路径的相对路径
function relative_path($path1,$path2){
    $path1 = explode('/',dirname($path1));
    $path2 = explode('/',dirname($path2));
    $length = count($path2);
    for($i=0;$i<$length;$i++){
        if($path2[$i] != $path1[$i]){
            break;
        }
    }
```

```
    if($i == 0){    //不在同一个目录下
        $split = array();
    }
    if($i!=0 && $i<$length){
        $split = array_fill('..',0,$length-$i);
    }
    if($i==$length){
        $split = array('.');
    }

    return implode('/',array_merge($split,array_slice($path1,$i)));
}

//从 URL 取出.php 为扩展名的文件名如：index.php
$url1 = 'http://club.sohu.com/abc/de/index.html';
$url2 = 'http://club.sohu.com/ef/hi/page.php?id=1';
$url3 = 'http://club.sohu.com/xy/read.php?key=123&board=456';

var_dump(pathinfo(parse_url($url2)['path'],PATHINFO_BASENAME));
var_dump(basename(parse_url($url3)['path']));

//匹配 URL:/^http:\/\/[\w]+(\.[\w]+)+$/


/*================================ 好 收 网 （ * ）
====================================*/

//1.列举 PHP 的变量类型
//四种标量：integer    string    float    boolean
//两种复合：array      object
//两种特殊：resource   null

//2.简述 PHP 的常用函数类型，并写出常用的函数
//                数 组 函 数 ：
sort,rsort,asort,arsort,array_num,array_count_value,array_pop,array_push,array_shift,array_uns
hift,shuffle
//字符串函数：substr,strlen,str_split,explode,implode,md5,strstr,strpos,strrev
//数学函数：max,min,rand,abs
//日期函数：data,strtotime,mktotime

//3.列举链接 mysql 数据库方法的实现
//mysql 连接：mysql_connect($host,$user,$pwd);
//mysqli 连接：new mysqli($host,$user,$pwd,$db);
```

```
//4.PHP 实现创建多级目录的函数
function create_dir($path,$mode = 0777){
    if(is_dir($path)){
        echo "目录已经存在！ <br/>";
        return ;
    }
    if(mkdir($path,$mode,true)){
        echo '目录创建成功！ <br/>';
    }else{
        echo '目录创建失败！ <br/>';
    }
}

//5.谈谈你对分层的理解
//首先视图接受用户输入请求，然后将请求传递给控制器，
//控制器再调用某个模型来处理用户的请求，
//在控制器的控制下，再将处理后的结果交给某个视图进行格式化输出给用户.这是经典的 MVC 设计执行的基本流程。
//MVC 不仅实现了功能模块和显示模块的分离，同时它还提高了系统的可维护性、可扩展性和组件的可复用性，是一个优秀的创建软件的途径。

//7.接口和抽象类的理解
//1.抽象类是特殊的类，接口是特殊的抽象类
//2.抽象类是单继承 extends，接口可以多实现 implements
//3.抽象不能被实例化
//4.接口没有构造函数，抽象类可以有构造函数
//5. 接 口 中 的 方 法 默 认 都 是 public 类 型 的 ， 而 抽 象 类 中 的 方 法 可 以 使 用 private,protected,public 来修饰

//8.写出知道的设计模式，和对设计模式的理解
//简单工场模式，工厂方法模式，单例模式，策略模式，观察者模式，命令模式，适配器模式

//1.写出 url 组成部分
//URL 链接的组成部分：协议 主机 端口 路径 四个部分组成

//2.写出常用状态码及含义
//http 状态码的标志：200 请求成功，502 网关错误，500 服务器内部错误，404 资源找不到

//3.cookie 和 session 的区别
//1.cookie 是存储在客户端的，session 是存储在服务器端的，都是用来解决 HTTP 协议无状态的
//2.基于 cookie 的 session，需要 cookie 传递 session_id,进行会话跟踪
```

//4.写出 web 的请求流程
//web 的请求流程：
//　　建立 TCP 连接-》
//　　Web 浏览器向 Web 服务器发送请求命令 -》Web 浏览器发送请求头信息 -》
//　　Web 服务器应答-》Web 服务器发送应答头信息 -》Web 服务器向浏览器发送数据 -》
//　　Web 服务器关闭 TCP 连接

//5.myqsl 的数据表引擎有哪些？区别是什么？
//mysqlde 的索引 myisam ,innodb,BDB,heap
//MYISAM 和 INNODB 的区别
//1.MYISAM 不支持事务，innodb 支持事物
//2.myisam 是表级锁，innodb 是行级锁
//3.mslsam 支持全文索引，innodb 不支持全文索引
//4.myisam 常用于 select 较多的数据库，innodb 常用于更新较多的数据库

//6.mysql 的索引类型有哪些？创建索引的优缺点，如何创建索引？
//常见索引类型有：
//1.主键索引
//2.唯一索引
//3.联合索引
//4.普通索引
//创建索引的有优缺点：
//优点：
//创建索引可以大大提高系统的性能。
//第一，　通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。
//第二，　可以大大加快数据的检索速度，这也是创建索引的最主要的原因。
//第三，　可以加速表和表之间的连接，特别是在实现数据的参考完整性方面特别有意义。
//第四，　在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间。
//第五，　通过使用索引，可以在查询的过程中，使用优化隐藏器，提高系统的性能。
//
//缺点：
//第一，　创建索引和维护索引要耗费时间，这种时间随着数据量的增加而增加。
//第二，　索引需要占物理空间，除了数据表占数据空间之外，每一个索引还要占一定的物理空间，如果要建立聚簇索引，那么需要的空间就会更大。
//第三，　当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，这样就降低了数据的维护速度。

//如何创建索引：
//一般来说，应该在这些列上创建索引。
//第一，　在经常需要搜索的列上，可以加快搜索的速度；
//第二，　在作为主键的列上，强制该列的唯一性和组织表中数据的排列结构；
//第三，　在经常用在连接的列上，这些列主要是一些外键，可以加快连接的速度；
//第四，　在经常需要根据范围进行搜索的列上创建索引，因为索引已经排序，其指定的

范围是连续的；
//第五，　在经常需要排序的列上创建索引，因为索引已经排序，这样查询可以利用索引的排序，加快排序查询时间；
//第六，　在经常使用在 WHERE 子句中的列上面创建索引，加快条件的判断速度。

今天接到了阿里的电话面试，感觉很是激动，趁着还有些印象我就总结一下吧，分享给有需要的同学！
简单的自我介绍

毫无特色的自我介绍：我叫 XXX….balabala…感觉不到一分钟
说一下二分查找的思想

前提::线性表中记录必须是有序的,线性表采用顺序存储
原理:在有序表中,取中间记录作为比较对象,若给定的值和中间记录的关键字相同,则查找成功;如果查找对象小于中间记录的关键字,则在中间记录的左半部分继续查找;否则在右半区查找.
时间复杂度: O(log(n))
如何判断链表中是否有环

原理：用两个指针，pSlow，pFast，就是一个慢一个快
慢的一次跳一步，
快的一次跳两步，
什么时候快的追上慢的了（就是 pSlow == pFast || pSlow->next == pFast），就表示有环。
代码如下：

```
#include <stdio.h>

typedef struct Node{
    int val;
    Node *next;
}Node,*pNode;

//判断是否有环
bool isLoop(pNode pHead)
{
    pNode fast = pHead;
    pNode slow = pHead;
    //如果无环，则 fast 先走到终点//当链表长度为奇数时，fast->Next 为空//当链表长度为偶数时，fast 为空 while( fast != NULL && fast->next != NULL){
        fast = fast->next->next;
        slow = slow->next;
        //如果有环，则 fast 会超过 slow 一圈 if(fast == slow){
            break;
        }
```

```
        }

        if(fast == NULL || fast->next == NULL    ){
            return false;
        }else{
            return true;
        }
}

//计算环的长度
int loopLength(pNode pHead){
        if(isLoop(pHead) == false)
            return 0;
        pNode fast = pHead;
        pNode slow = pHead;
        int length = 0;
        bool begin = false;
        bool agian = false;
        while( fast != NULL && fast->next != NULL){
            fast = fast->next->next;
            slow = slow->next;
            //超两圈后停止计数，挑出循环 if(fast == slow && agian == true)
                break;

            //超一圈后开始计数 if(fast == slow && agian == false){
                begin = true;
                agian = true;
            }

            //计数 if(begin == true)
                ++length;
        }
        return length;
}


//求出环的入口点
Node* findLoopEntrance(pNode pHead){
        pNode fast = pHead;
        pNode slow = pHead;
        while( fast != NULL && fast->next != NULL){
            fast = fast->next->next;
            slow = slow->next;
            //如果有环，则 fast 会超过 slow 一圈 if(fast == slow){
```

```
                break;
            }
        }
        if(fast == NULL || fast->next == NULL)
            return NULL;
        slow = pHead;
        while(slow != fast){
            slow = slow->next;
            fast = fast->next;
        }

        return slow;
}
```

说一下 cookie 和 session 的区别

简单的描述：cookie 存在客户端，session 存在服务器端。cookie 分两种：一种叫会话 cookie 是没有设置过期时间，当关闭浏览器后 cookie 将消失；另一种是设置了过期时间叫持久 cookie，

这种 cookie 存储在磁盘中关闭浏览器后 cookie 不会消失。
就安全性来说 cookie 存在客户端的磁盘上对用户透明 session 存在服务器端而且 sessionID 是加密的相对来说 seesion 较安全。
用户的会话通过 cookie 中存储的 session id 来和服务器端的 session 进行关联从而保持正常的会话。
如果把 cookie 禁掉 session 还能用吗？

通过 url 传值，把 session id 附加到 url 上（缺点：整个站点中不能有纯静态页面，因为当是纯静态页面 session id 将无法继续向后传了）
通过隐藏表单，把 session id 放到表单的隐藏文本框中同表单一块提交过去（缺点：不适用<a>标签这种直接跳转的非表单的情况）
直接配置 php.ini 文件,将 php.ini 文件里的 session.use_trans_sid= 0 设为 1,
用文件、数据库等形式保存 Session ID，在跨页过程中手动调用
session 都是可以存在哪？

数据库、缓存、文件
PHP 如何获取客户端的 IP 地址?

$_SERVER[‘REMOTE_ADDR’] :通过全局数组来获得
getenv(‘REMOTE_ADDR’) :通过环境变量来获得
用$_SERVER 获取的 IP 地址有什么问题？

当客户机使用代理的时候获取不到真实的 IP 地址
isset()和 array_key_exists()有什么区别？

对于数组值的判断不同，对于值为 null 或”或 false,isset 返回 false，array_key_exists 返回 true
执行效率不同，isset 是内建运算符，array_key_exists 是 php 内置函数，isset 要快一些
当用 isset 访问一个不存在索引数组值时，不会引起一个 E_NOTICE 的 php 错误消息
array_key_exists 会调用 get_defined_vars 判断数组变量是否存在，isset 不用
HTTP 请求头和相应头信息都有哪些？

请求头:
Accept: text/html,image/*(浏览器可以接收的类型)
Accept-Charset: ISO-8859-1(浏览器可以接收的编码类型)
Accept-Encoding: gzip,compress(浏览器可以接收压缩编码类型)
Accept-Language: en-us,zh-cn(浏览器可以接收的语言和国家类型)
Host: www.it315.org:80(浏览器请求的主机和端口)
If-Modified-Since: Tue, 11 Jul 2000 18:23:51 GMT(某个页面缓存时间)
Referer: http://www.it315.org/index.jsp(请求来自于哪个页面)
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)(浏览器相关信息)
Cookie：(浏览器暂存服务器发送的信息)
Connection: close(1.0)/Keep-Alive(1.1)(HTTP 请求的版本的特点)
Date: Tue, 11 Jul 2000 18:23:51 GMT(请求网站的时间)
响应头：
Location: http://www.it315.org/index.jsp(控制浏览器显示哪个页面)
Server:apache tomcat(服务器的类型)
Content-Encoding: gzip(服务器发送的压缩编码方式)
Content-Length: 80(服务器发送显示的字节码长度)
Content-Language: zh-cn(服务器发送内容的语言和国家名)
Content-Type: image/jpeg; charset=UTF-8(服务器发送内容的类型和编码类型)
Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT(服务器最后一次修改的时间)
Refresh: 1;url=http://www.it315.org(控制浏览器 1 秒钟后转发 URL 所指向的页面)
Content-Disposition: attachment; filename=aaa.jpg(服务器控制浏览器发下载方式打开文件)
Transfer-Encoding: chunked(服务器分块传递数据到客户端）
Set-Cookie:SS=Q0=5Lb_nQ; path=/search(服务器发送 Cookie 相关的信息)
Expires: -1(服务器控制浏览器不要缓存网页，默认是缓存)
Cache-Control: no-cache(服务器控制浏览器不要缓存网页)
Pragma: no-cache(服务器控制浏览器不要缓存网页)
Connection: close/Keep-Alive(HTTP 请求的版本的特点)
Date: Tue, 11 Jul 2000 18:23:51 GMT(响应网站的时间)
说一下 HTTP 常用的状态码并解释涵义

一些常见的状态码为：
200 – 服务器成功返回网页
304 – 未修改
403 – 服务器拒绝请求
404 – 请求的网页不存在
502 – 网关错误
503 – 服务器超时

详解：常用的 HTTP 状态码
HTTP 常见的请求类型都有哪些？区别是什么？

get：请求的数据随 HTTP 请求头发过去
post:请求的数据在 HTTP 请求头发过去之后再发过去
get 方法的数据大小是有一定限制的。而且发送的数据容易被人看到。>而 post 就没有这些特点
说一下 PHP 中的 error_reporting 是干嘛用的

error_reporting() 设置 PHP 的报错级别并返回当前级别。
说一下错误界别都是有哪些，如何自定义错误处理级别？

E_WARNING 非致命的 run-time 错误。不暂停脚本执行。
E_NOTICE Run-time 通知。脚本发现可能有错误发生，但也可能在脚本正常运行时发生
E_USER_ERROR 致命的用户生成的错误。
E_USER_WARNING 非致命的用户生成的警告。
E_USER_NOTICE 用户生成的通知。
E_RECOVERABLE_ERROR 可捕获的致命错误。
E_ALL 所有错误和警告，除级别 E_STRICT 以外
自定错误处理
PHP 的默认错误处理程序是内建的错误处理程序。可以使用 set_error_handler(“函数名称”)来自定义错误处理使其仅应用到某些错误，
说一下面向对象中类的概念,并简单举例说明一下

按专业的概念说就是“具有相同属性和服务的同一类事物的抽象”。类是一个抽象的概念。例如：人类。
常用的数据库引擎有哪些

我的回答：MYISAM 和 InnoDB。然后就抛出了下一个问题
MYISAM 和 InnoDB 的区别是什么？

MYISAM 不支持事务 InnoDB 支持事务
MYISAM 是表级锁 由于是表级锁，对高并发性的 update 效率较低；InnoDB 是行级锁由于是行级锁，对高并发性的 update 效率相对较高
MYISAM 支持全文索引 InnoDB 不支持全文索引
MYISAM 使用用于大量 select 操作的数据库，InnoDB 适合用于执行大量的 INSERT 或 UPDATE 操作的数据库
如何优化数据库？

创建相应的索引：索引有助于提高检索性能，但过多或不当的索引也会导致系统低 效。因为用户在表中每加进一个索引，数据库就要做更多的工作。过多的索引甚至会导致索引碎片。
分表：针对每个时间周期产生大量的数据，可以考虑采用一定的策略将数据存到多个数据表中
分库：是将系统按照模块相关的特征分布到不同的数据中，以提高系统整体负载能力，如：

读写分离
sql 优化:
in 和 not in 也要慎用，因为 IN 会使系统无法使用索引,而只能直接搜索表中的数据
尽量避免在 where 子句中对字段进行表达式操作，这将导致引擎放弃使用索引而进行全表扫描
SELECT * FROM T1 WHERE F1/2=100
应改为:
SELECT * FROM T1 WHERE F1=100*2
充分利用连接条件,在某种情况下,两个表之间可能不只一个的连接条件,这时在 WHERE 子句中将 连接条件完整的写上，有可能大大提高查询速度。
例：
SELECT SUM(A.AMOUNT) FROM ACCOUNT A,CARD B WHERE A.CARD_NO = B.CARD_NO 改成
SELECT SUM(A.AMOUNT) FROM ACCOUNT A,CARD B WHERE A.CARD_NO = B.CARD_NO AND A.ACCOUNT_NO=B.ACCOUNT_NO
如何对比查看两条 sql 语句的执行效果？

可以收用 explain 进行查看。例如：explain select * from user_info;
根据数据设计数据库（考虑优化）:“学生”,“成绩”,“课程”

这个就没什么好说的了，就是平时创建数据库的那一套东西什么索引、外键、等等
Linux 中统计一个文件中指定字符出项的次数

grep -o ‘搜索的字符’ file | wc -l

一面（技术面）:
面试:题 1：你用的是什么服务器 Apache OR Nginx，说说二者的区别。
我：我用的是 Apache,因为之前做的项目都小中型项目，所以用 Apache 会比较稳点，二者的区别主要是两者的运行机制不同，对于中小型的项目来说不涉及到高并发量的数据请求 Apache 处理起

来会很轻松并且很稳定。最核心的区别在于 apache 是同步多进程模型，一个连接对应一个进程；nginx 是异步的，多个连接（万级别）可以对应一个进程 ；Apache 在处理动态有优势，Nginx 并

发性比较好，CPU 内存占用低，如果 rewrite 频繁，那还是 Apache
拓展问题：1、select 和 epull 的区别？
2、Apache 解析 PHP 的三种方式？哪种方式最稳定？（把 PHP 配置到 Apache 模块中最稳定）
答： CGI、fastCGI 和 Apache 的模块化配置
3、fastCGI 和 CGI 的区别？（最大的区别是 fastCGI 能独立运行，而 CGI 只能依托于 Apache 运行）
答：fastCGI 是一种常驻内存的 CGI，只需要加载一次，解决了 CGI 程序每次都要重新进行 fork 一个新
的进程。 最大的区别是 fastCGI 能独立运行，而 CGI 只能依托于 Apache 运

行。
面试题 2：MySQL 都有什么引擎？
我：MySQL 引擎有很多，比较常用的有 MyISAM 和 InnoDB.
拓展问题：1、MyISAM 和 Inno DB 区别
2、MySQL 的存储方式是什么？（b+tree）
3、MyISAM 和 InnoDB 的存储方式有什么不同？（MyISAM 是堆表）
4、MyISAM 比 InnoDB 在 select 时快么，为什么？
INNODB 在做 SELECT 的时候，要维护的东西比 MYISAM 引擎多很多:
5、InnoDB 为什么必须要建主键索引？
6、InnoDB 是什么级别的锁，数据库的隔离级别有哪些？
数据库的隔离级别
7、怎样理解 InnoDB 的行级锁？
8、平时什么情况下用到事务？怎样理解事务？

面试题 3：PHP 用什么函数连接 MySQL？
我：mysql_connect('数据库名'，'用户名'，'密码')；
拓展问题：1、PHP 连接 MySQL 的扩展有哪些？它们有什么区别？（mysql , mysqli , PDO）
2、什么是 PDO？
3、PHP 怎么执行 sql 语句？（使用 query()函数）用什么来得到查询的结果集？(mysql_fetch_array()或
者 mysql_fetch_row()) 得到的结果集是什么格式的？（数组）
还有关于缓存，计算机网络，数据结构，php 基础的知识（数组，字符串处理，文件等）反正挺多的。
二面（技术总监面）：
1、谈项目问题：介绍了大学期间做的项目（自己负责的那一块）；
2、怎样设计一张用户表，说说各个字段的数据类型，需要在什么字段上建立索引？
3、MyISAM 和 InnoDB 区别？
4、什么是全文索引？全文索引的原理是什么？
让我提问：1、贵公司的技术团队有多少人？ PHP 开发的有多少？
2、如果到公司了我能做点什么？
3、实习生的待遇怎么样？
4、实习生转正的名额有多少？
5、如果实习结束留到公司转正的工资范围是多少？

三面（HR 面）：
1、介绍一下自己的情况。
2、介绍了一下公司的整体的业务模块和流程。
3、问我自己想要的实习工资是多少？（我从住房、车费、吃饭等多个方面进行分析）
4、聊聊自己的人生规划。挺开心的。

新浪 php 面试题

1. echo count("abc"); 输出什么？

2. 用 PHP 写出显示客户端 IP 与服务器 IP 的代码

3. error_reporting(2047)什么作用？

error_reporting 设定错误讯息回报的等级
2047 我记得应该是 E_ALL。

php.ini 文件中有许多配置设置。您应当已经设置好自己的 php.ini 文件并把它放在合适的目录中，就像在 Linux 上安装 PHP 和 Apache 2 的文档说明中所示的那样（请参阅 参考资料）

。在调试 PHP 应用程序时，应当知道两个配置变量。下面是这两个变量及其默认值：
display_errors = Off
error_reporting = E_ALL

E_ALL 能从不良编码实践到无害提示到出错的所有信息。E_ALL 对于开发过程来说有点太细，因为它在屏幕上为一些小事（例如变量未初始化）也显示提示，会搞糟浏览器的输出
所以不建议使用 2047，最好把默认值改为：error_reporting = E_ALL & ~E_NOTICE

4. echo，print()和 print_r()有什么区别？

5. 打开 php.ini 中的 Safe_mode，会影响哪些参数？至少说出 6 个。

6. 写个函数来解决多线程同时读写一个文件的问题。

7. 请写一个函数验证电子邮件的格式是否正确（要求使用正则）
北京兼职招聘：http://job.aftjob.com/job/category-34.html
8. 考 SQL 语句的题，题太长了，实在不好回忆了。

9. MySQL 数据库，一天一万条以上的增量，怎么优化？

10. 写出一种排序算法（要写出代码），并说出优化它的方法。

11. 写个函数用来对二维数组排序。

12. 写 5 个不同的自己的函数，来截取一个全路径的文件的扩展名，允许封装 php 库中已有的函数。

13. 一群猴子排成一圈，按 1，2，…，n 依次编号。然后从第 1 只开始数，数到第 m 只,把它踢出圈，从它后面再开始数，再数到第 m 只，在把它踢出去…，如此不停的进行下去，直到最后只剩

下一只猴子为止，那只猴子就叫做大王。要求编程模拟此过程，输入 m、n, 输出最后那个

大王的编号。

PHP:
1.$_GET,$_POST,$_REQUEST 都是做什么用的

2.
if(strpos($str, 'a') == false) {}
这个语句有什么 bug?

3.isset($foo),!empty($foo),($foo)
作用分别是什么?

4.根据数组:
$a = array(
'a'=>'apple',
'b'=>'banan'
);
输出一个下拉框

5.优化下列函数
function text($str) {
preg_replace('/a/', 'z', $str);
preg_replace('/b/', 'z', $str);
preg_replace('/c/', 'z', $str);
return $str;
}
http://www.aftjob.com/search.php?mod=forum
6.如何处理 SQL 注入

MySQL
7.如何读取一个表的表类型,以及读取一个表中字段的类型.

8.如果这三个字段(a,b,c)都有索引,下列 SQL 语句那个更快
select ... from ... where c=...
select ... from ... where a=... and c=...
select ... from ... where b=... and c=...
(SQL 语句应该没记错,这道和网上流传的英文版面试题差不多)

9.忘了,好像是问的 MySQL 的 MyISAM 有什么优点?

10.记得不太清楚,大致是根据字段,取出记录中的第一个月,最后一天?

11.多台 MySQL 服务器,如何同步数据
我回答:没用过- -,只用过 SQLServer 的数据订阅
1. 写一个函数，尽可能高效的，从一个标准 url 里取出文件的扩展名
例如: http://www.phpddt.com/abc/de/fg.php?id=1 需要取出 php 或 .php

答：我是直接用 PHP 内置函数搞定的，不重复造轮子，估计出题者也是想考察基础知识，主要是解析 url 和一个返回文件信息的函数（扩展：取得文件后缀名的多种方法）：

```
<?php
    /** by www.phpddt.com */
    $url = "http://www.phpddt.com/abc/de/fg.php?id=1";
    $path = parse_url($url);
    echo pathinfo($path['path'],PATHINFO_EXTENSION);  //php
?>
```

2. 在 HTML 语言中，页面头部的 meta 标记可以用来输出文件的编码格式，以下是一个标准的 meta 语句
<META http-equiv='Content-Type' content='text/html; charset=gbk'>
请使用 PHP 语言写一个函数，把一个标准 HTML 页面中的类似 meta 标记中的 charset 部分值改为 big5
请注意:
(1) 需要处理完整的 html 页面，即不光此 meta 语句
(2) 忽略大小写
(3) ' 和 " 在此处是可以互换的
(4) 'Content-Type' 两侧的引号是可以忽略的，但 'text/html; charset=gbk' 两侧的不行
(5) 注意处理多余空格

答：表示我正则表达式（PHP 正则详解）忘记差不多了，弄了半天。

```
<?php
    /** http://www.phpddt.com */
    $html = "<meta http-equiv='Content-Type' content='text/html; charset=gbk'>";
    //匹配标准的 meta 标签
    $pattern = "/<meta\s+http-equiv=(\'|\")?Content-Type(\'|\")?\s+content=(\'|\")text\/html;\s+charset=(.*)(\'|\")>/i";
    $replacement = "<meta http-equiv='Content-Type' content='text/html; charset=big5'>";
    $result = preg_replace($pattern, $replacement, $html);
    echo htmlspecialchars($result);
?>
```

3. 写一个函数，算出两个文件的相对路径

如 $a = '/a/b/c/d/e.php';

$b = '/a/b/12/34/c.php';

计算出 $b 相对于 $a 的相对路径应该是 ../../c/d 将()添上

答案：

```
<?php
    /** by www.phpddt.com */
    $a = '/a/b/c/d/e.php';
    $b = '/a/b/13/34/c.php';
    echo getRelativePath($a, $b); //"../../12/34/"
    function getRelativePath($a,$b){
        $a2array = explode('/', $a);
        $b2array = explode('/', $b);
        $relativePath      = '';
        for( $i = 1; $i <= count($b2array)-2; $i++ ) {
            $relativePath .= $a2array[$i] == $b2array[$i] ? '../' : $b2array[$i].'/';
        }
        return $relativePath;
    }
?>
```

4.写一个函数，能够遍历一个文件夹下的所有文件和子文件夹。

答：这个我之前就在博客中写过（PHP 文件遍历及文件拷贝），只是实现的方法很多，效率不一定最高

```
/*
 *@blog    http://www.phpddt.com
 */
function listDir($dir = '.'){
    if ($handle = opendir($dir)) {
        while (false !== ($file = readdir($handle))) {
            if($file == '.' || $file == '..'){
                continue;
            }
            if(is_dir($sub_dir = realpath($dir.'/'.$file))){
                echo 'FILE in PATH:'.$dir.':'.$file.'<br>';
                listDir($sub_dir);
            }else{
                echo 'FILE:'.$file.'<br>';
            }
```

```
            }
            closedir($handle);
        }
    }

    listDir('e:\www\abc');
```

5.简述论坛中无限分类的实现原理。

答：无限极分类，那么应该是考察递归函数吧！

第一步：建立测试数据库：

```
CREATE TABLE `category` (
  `id` smallint(5) unsigned NOT NULL auto_increment,
  `fid` smallint(5) unsigned NOT NULL default '0',
  `value` varchar(50) NOT NULL default '',
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

第二步：插入测试数据：

```
INSERT INTO `category` ( `fid`, `value`) VALUES
(0, 'PHP 点点通博客 PHPDDT.COM'),
(1,'a'),
(1,'b'),
(2,'c'),
(2,'d'),
(4,'e')
```

第三步：递归输出分类：

```
<?php
/** by www.phpddt.com */
$conn = mysql_connect("localhost", "root", "mckee");
mysql_select_db("test",$conn);
mysql_query("set names utf8");
$sql = "SELECT * FROM category";
$res = mysql_query($sql);
while($row = mysql_fetch_assoc($res)){
    $arr[] = array($row[id],$row[fid],$row[value]);
```

```
}
getCate(0);
function getCate($fid = 0) {
    global $arr;
    for ($i = 0; $i < count($arr); $i++) {
        if ($arr[$i][1] == $fid) {
            echo $arr[$i][2] . "<br>";
            getCate($arr[$i][0]); //递归
        }
    }
}
?>
```

6.设计一个网页，使得打开它时弹出一个全屏的窗口，该窗口中有一个文本框和一个按钮。用户在文本框中输入信息后点击按钮就可以把窗口关闭，而输入的信息却在主网页中显示！

答案：尼玛。都没明白出这题目是干嘛的，新浪工程师脑子进水了吗？考察 js 的 window 对象？亲们告诉我？

index.html

```
<html>
    <head>
        <title>by www.phpddt.com</title>
    </head>
    <body>
        <h1></h1>
        <script type="text/javascript">
            open('fullwin.html');
        </script>
    </body>
</html>
```

fullwin.html

```
<html>
    <head>
        <title>by www.phpddt.com</title>
    </head>
    <body>
        <script type="text/javascript">
```

```
                window.moveTo(0, 0);
                window.resizeTo(window.screen.width, window.screen.height);
                var s = prompt('请输入：');
                window.opener.document.getElementsByTagName('h1')[0].innerText = s;
                window.close();
            </script>
        </body>
    </html>
```

前天接到了新浪乐居（http://www.leju.com/）的面试邀请函了，昨天去面试了，经历了笔试、技术面，技术总监面，HR 面共 3 个半小时。今天上午打电话说给实习的 offer，说要下周三入职

，我说现在我在面试等回去给她回复。

面试总体给我的感受是新浪乐居很侧重数据库的知识，问的都是 Mysql 很深入的知识点。

一面（技术面）：

面试:题 1：你用的是什么服务器 Apache OR Nginx，说说二者的区别。

我：我用的是 Apache,因为之前做的项目都小中型项目，所以用 Apache 会比较稳点，二者的区别主要是两者的运行机制不同，对于中小型的项目来说不涉及到高并发量的数据请求 Apache 处理起

来会很轻松并且很稳定。最核心的区别在于 apache 是同步多进程模型，一个连接对应一个进程；nginx 是异步的，多个连接（万级别）可以对应一个进程 ；Apache 在处理动态有优势，Nginx 并

发性比较好，CPU 内存占用低，如果 rewrite 频繁，那还是 Apache

拓展问题：1、select 和 epull 的区别？

2、Apache 解析 PHP 的三种方式？哪种方式最稳定？（把 PHP 配置到 Apache 模块中最稳定）

答： CGI、fastCGI 和 Apache 的模块化配置

3、fastCGI 和 CGI 的区别？（最大的区别是 fastCGI 能独立运行，而 CGI 只能依托于 Apache 运行）

答：fastCGI 是一种常驻内存的 CGI，只需要加载一次，解决了 CGI 程序每次都要重新进行 fork 一个新

的进程。 最大的区别是 fastCGI 能独立运行，而 CGI 只能依托于 Apache 运行。

面试题 2：MySQL 都有什么引擎？

我：MySQL 引擎有很多，比较常用的有 MyISAM 和 InnoDB.

拓展问题：1、MyISAM 和 Inno DB 区别

2、MySQL 的存储方式是什么？（b+tree）

3、MyISAM 和 InnoDB 的存储方式有什么不同？（MyISAM 是堆表）

4、MyISAM 比 InnoDB 在 select 时快么，为什么？

5、InnoDB 为什么必须要建主键索引？

6、InnoDB 是什么级别的锁，数据库的隔离级别有哪些？

7、怎样理解 InnoDB 的行级锁？

8、平时什么情况下用到事务？怎样理解事务？

面试题 3：PHP 用什么函数连接 MySQL？
我：mysql_connect('数据库名'，'用户名'，'密码')；
拓展问题：1、PHP 连接 MySQL 的扩展有哪些？它们有什么区别？（mysql , mysqli , PDO）
2、什么是 PDO？
3、PHP 怎么执行 sql 语句？（使用 query()函数）用什么来得到查询的结果集？(mysql_fetch_array()或者 mysql_fetch_row()) 得到的结果集是什么格式的？（数组）
还有关于缓存，计算机网络，数据结构，php 基础的知识（数组，字符串处理，文件等）反正挺多的。
二面（技术总监面）：
1、谈项目问题：介绍了大学期间做的项目（自己负责的那一块）；
2、怎样设计一张用户表，说说各个字段的数据类型，需要在什么字段上建立索引？
3、MyISAM 和 InnoDB 区别？
4、什么是全文索引？全文索引的原理是什么？
让我提问：1、贵公司的技术团队有多少人？ PHP 开发的有多少？
2、如果到公司了我能做点什么？
3、实习生的待遇怎么样？
4、实习生转正的名额有多少？
5、如果实习结束留到公司转正的工资范围是多少？

三面（HR 面）：
1、介绍一下自己的情况。
2、介绍了一下公司的整体的业务模块和流程。
3、问我自己想要的实习工资是多少？（我从住房、车费、吃饭等多个方面进行分析）
4、聊聊自己的人生规划。挺开心的。
什么是 OAuth 授权？

一、什么是 OAuth 协议
OAuth(开放授权)是一个开放标准。
允许第三方网站在用户授权的前提下访问在用户在服务商那里存储的各种信息。
而这种授权无需将用户提供用户名和密码提供给该第三方网站。
OAuth 允许用户提供一个令牌给第三方网站，一个令牌对应一个特定的第三方网站，同时该令牌只能在特定的时间内访问特定的资源。

二、OAuth 的原理和授权流程
OAuth 的认证和授权的过程中涉及的三方包括：
服务商：用户使用服务的提供方，一般用来存消息、储照片、视频、联系人、文件等(比如 Twitter、Sina 微波等)。
用　户：服务商的用户
第三方：通常是网站，该网站想要访问用户存储在服务商那里的信息。

比如某个提供照片打印服务的网站，用户想在那里打印自己存在服务商那里的网络相册。
在认证过程之前，第三方需要先向服务商申请第三方服务的唯一标识。
OAuth 认证和授权的过程如下:
1、用户访问第三方网站网站，想对用户存放在服务商的某些资源进行操作。
2、第三方网站向服务商请求一个临时令牌。
3、服务商验证第三方网站的身份后，授予一个临时令牌。
4、第三方网站获得临时令牌后，将用户导向至服务商的授权页面请求用户授权，然后这个过程中将临时令牌和第三方网站的返回地址发送给服务商。
5、用户在服务商的授权页面上输入自己的用户名和密码，授权第三方网站访问所相应的资源。
6、授权成功后，服务商将用户导向第三方网站的返回地址。
7、第三方网站根据临时令牌从服务商那里获取访问令牌。
8、服务商根据令牌和用户的授权情况授予第三方网站访问令牌。
9、第三方网站使用获取到的访问令牌访问存放在服务商的对应的用户资源
PHP 面试题汇总
PHP
1、写出五种以上你使用过的 PHP 的扩展的名称（提示：常用的 PHP 扩展）
mysql、gd2、pdo、curl、mbstring、soap 等，在 php.ini 中可以找到。尽量多了解一些扩展，了解他们的功能（能做什么)。
php 通过使用 php_ming 库（Ming 库）快速生成 Flash 动画

了解 MVC 模式吗？请写出三种以上目前 PHP 流行的 MVC 框架名称（不区分大小写）
MVC 是 Model（模型）、View(视觉）、Controll（控制器）的缩写。
MVC(Model-View-Controller)介绍
模型(Model): 应用程序的模型部分关心的是欲显示的数据的细节。模型通常关注的是应用程序的业务逻辑部分，关注的是如何使用数据库来读取和保存数据。
视图(View): 视图关心的是用户显示的部分，它通常是 HTML。
控制器(Controller)：控制器将特定的模型和视图结合起来，保证将正确的数据显示到页面上。
常用的 MVC 框架：
Zend Framework
FleaPHP
qeephp
CakePHP
ThinkPHP
CI
YII
大家有时间可以对框架进入多一些的深入了解。

3、用 PHP 打印出前一天的时间格式是 2008-2-8 18:00:10
echo “昨天:”, date(“Y-m-d H:i:s”, strtotime(“-1 day”)), “
”;
echo “昨天:”, date(“Y-m-d H:i:s”, mktime (date(“H”), date(“i”), date(“s”), date(“m”), date(“d”)-1, date(“Y”))), “
”;
date(‘Y-m-d H:i:s’,Time () -24*3600)

参考 strtotime.php mktime.php

4、echo(),print(),print_r()的区别 var_dump()
echo 与 print:
它们都不是真正的函数，是一种语法结构（也有说 print 是函数，echo 不是）。
echo 和 print 后面都可不用加(),如: echo ‘ok’; print ‘ok’;
运行速度 echo 稍快一些，因为 echo 并不返回值，print 返回一个值 int(1)。
结论：
1、一般用 echo，除非三元运算时。$a=5; ($a==5) ? print ’5′: print 0;
2、echo 后一般不要跟()。
print_r 是递归打印，主要用于输出数组对象。
print 只能有一个参数，所以不能不能用”,”,而 echo 可以。
Sprintf 以一定的格式 格式化一个字符串
参考 echo_print_print_r.php

5、能够使 HTML 和 PHP 分离开使用的模板
其实 PHP 本身就是一种模版引擎。参考 require.php
常用的模板引擎：smarty，还有 PHPLib,FastTemplate,Savant 等。
模板引擎列表：http://www.sitepoint.com/forums/showthread.php?t=123769

6、如何实现字符串翻转？
可用内置函数 strrev。如果不准用 PHP 内置函数的就自己写:
参考 strrev2.php

$STR = ‘abc’;
$STR{0}
Hello->olleH

7、$a = “hello”;
$b = &$a;
unset($b);
$b = “world”;
what is $a?
参考 references.php
此题的目的是要深刻理解引用。

通常，在将一个变量的值赋给另外一个变量的时候，先产生原变量的一个副本，然后再将它保存在内存的其他地方。如:
$a = 5;
$b = $a;
首先产生$a 的一个副本，然后再将它保存到$b 中。如果随后改变$a 的值，$b 的值不会改变：
$a = 7; //这时$b 仍然是 5
可以使用引用操作符&来避免这样的副本。如：
$a = 5;

$b = &$a;
$b = 7; //这时$a 和$b 都会是 7。 这行也可以换成$b = 7;
引用是非常有趣的。请记住，引用就像一个别名，而不是一个指针。$a 和$b 都指向了内存的相同地址。可以通过重置它们来改变所指向的地址。如下所示：
unset($a);
重置并不会改变$b 的值，但是只可以破坏$a 和值 7 保存在内存中的连接。

可以搜索： 引用 site:php.net 进一步深入了解 PHP 的引用。
也可以直接访问这个页面：http://www.php.net/manual/zh/language.references.php

http://www.php.net/manual/zh/language.references.unset.php

当 unset 一个引用，只是断开了变量名和变量内容之间的绑定。这并不意味着变量内容被销毁了。例如：
<?php
$a = 1;
$b =& $a;
unset($a);
?>
不会 unset $b，只是 $a。
8、实现中文字串截取无乱码的方法。
a.可以用正则
b.用 mb_substr()
参考 mb_substr.php

北京 abc substr

9、$a = 1;
$x = &$a;
$b = $a++;
what is $b? $x?
参考 references2.php

10、$array = array();
$x = empty($array);
what is $x? true or false
参考 empty.php
什么样的内容为空？
empty()、isset()、is_null
11、用 PHP 写出显示客户端 IP 与服务器 IP 的代码
得到服务器端的 IP：
//gethostbyname() 参考 gethostbyname.php 有时候得不到。
$_SERVER['SERVER_ADDR'];
得到客户端的 IP：

$_SERVER['REMOTE_ADDR'];
参考 ip.php

12、某内容管理系统：用户提交内容后，系统生成静态 HTML 页面；写出实现的基本思路，最好写出相关代码。
用户提交内容后，将内容加在最终页面模板上，然后另存为 HTML 页面（创建 HTML 页面，将内容和页面模板写入）。
代码：

```
$tpl->assign('vars', $vars);
$static_html = = $tpl->fetch（'tpl/index.html'）; //Smarty 的 fetch 方法
$fp = fopen('html/index.html', 'w');
fwrite($fp, $static_html);
```

另一种 ob_start
createHtml.php

13、写出以下程序的输出结果

```
$b=201;
$c=40;
$a=$b>$c?4:5;
echo $a;
```

参考 3yuan.php

14、写出以下程序的输出结果

```
$str="cd";
$$str="hotdog"; // $cd = 'hotdog'
$$str.="ok";
echo $cd;
```

参考$$.php

15、在 PHP 中 error_reporting 这个函数有什么作用？
设定 php 脚本的错误报告级别
error_reporting(6143)的作用是设定 php 脚本的错误报告级别为”所有错误”。
ini_set（'display_errors', 1); //作用是在显示 PHP 脚本错误，相当于修改 php.ini 中的

error_reporting

定义和用法
error_reporting() 设置 PHP 的报错级别并返回当前级别。
语法
error_reporting(report_level)

如果参数 level 未指定，当前报错级别将被返回。下面几项是 level 可能的值：

值

常量

描述

1

E_ERROR

Fatal run-time errors. Errors that can not be recovered from. Execution of the script is halted

2

E_WARNING

Non-fatal run-time errors. Execution of the script is not halted

4

E_PARSE

Compile-time parse errors. Parse errors should only be generated by the parser

8

E_NOTICE

Run-time notices. The script found something that might be an error, but could also happen when running a script normally

16

E_CORE_ERROR

Fatal errors at PHP startup. This is like an E_ERROR in the PHP core

32

E_CORE_WARNING

Non-fatal errors at PHP startup. This is like an E_WARNING in the PHP core

64

E_COMPILE_ERROR

Fatal compile-time errors. This is like an E_ERROR generated by the Zend Scripting Engine

128

E_COMPILE_WARNING

Non-fatal compile-time errors. This is like an E_WARNING generated by the Zend Scripting Engine

256

E_USER_ERROR

Fatal user-generated error. This is like an E_ERROR set by the programmer using the PHP function trigger_error()

512

E_USER_WARNING

Non-fatal user-generated warning. This is like an E_WARNING set by the programmer using the PHP function trigger_error()

1024

E_USER_NOTICE

User-generated notice. This is like an E_NOTICE set by the programmer using the PHP function trigger_error()

2048

E_STRICT

Run-time notices. PHP suggest changes to your code to help interoperability and compatibility of

the code
4096
E_RECOVERABLE_ERROR
Catchable fatal error. This is like an E_ERROR but can be caught by a user defined handle (see also set_error_handler())
8191
E_ALL
All errors and warnings, except level E_STRICT (E_STRICT will be part of E_ALL as of PHP 6.0)
例子
任意数目的以上选项都可以用”或”来连接（用 OR 或 |），这样可以报告所有需要的各级别错误。例如，下面的代码关闭了用户自定义的错误和警告，执行了某些操作，然后恢复到原始的报错级别：

```
<?php
//禁用错误报告
error_reporting(0);
//报告运行时错误
error_reporting(E_ERROR | E_WARNING | E_PARSE);
//报告所有错误
error_reporting(E_ALL);
?>
```

16、简述如何得到当前执行脚本路径，包括所得到参数。
访问 http://temp.com/phpinfo.php?id=1

```
echo $_SERVER['SCRIPT_URL']; //得到/phpinfo.php
echo $_SERVER["SCRIPT_URI"]; //得到 http://temp.com/phpinfo.php
echo $_SERVER["SCRIPT_FILENAME"]; //得到 F:/www/Temp/phpinfo.php
echo $_SERVER["REQUEST_URI"]; //得到/phpinfo.php?id=1
echo $_SERVER["SCRIPT_NAME"]; //得到/phpinfo.php
```

参考 server.php http://lesson.com/test/server.php?id=1

17、有一个网页地址 http://bbs.91lamp.com/index.php ,如何得到它的 html 内容
file_get_contents()

T 数组函数 arsort 的作用是__。
对数组进行逆向排序并保持索引关系

```
/*
rsort — 对数组逆向排序
sort — 对数组排序
*/
```

参考 arsort.php

19、执行程序段<?php echo 8%(-2) ?>将输出__0。

参考%.php
20、语句 include 和 require 都能把另外一个文件包含到当前文件中，它们的区别是__；为了避免多次包含同一文件，可以用语句__来代替它们。
发生异常时 include 产生警告，程序继续执行；require 产生致命错误，程序停止往下执行。
一般推荐使用 require（更能调试错误）。
require_once()/include_once()

require 重复调用会多次加载你引用的文件；require_once 只加载一次，而不管你实际上调用了多少次，主要用于复杂的文件包含关系。
例如 b 包含 a，c 包含 a，但同时 c 又包含了 b，那么如果用 require 的话可能会导致两次加载 a，这时应使用 require_once。

实际开发过程中：如果确定某个文件只会被包含一次，那么用 require，否则用 require_once。因为 require 不需要检测文件是否被包含过，比 require_once 的执行效率要高。

21、一个函数的参数不能是对变量的引用，除非在 php.ini 中把__设为 on。
allow_call_time_pass_reference
quote.php

22、在 PHP 中，heredoc 是一种特殊的字符串，它的结束标志必须__。
结束标识符所在的行不能包含任何其它字符除”;”
参考 heredoc.php

23、有一数组 $a=array(3,2,4,9,8);请将其重新排序，按从小到大的顺序列出。
可用冒泡法进行排序：
冒泡排序的基本概念是：依次比较相邻的两个数，将小数放在前面，大数放在后面。即首先比较第 1 个和第 2 个数，将小数放前，大数放后。然后比较第 2 个数和第 3 个数，将小数放前，大数放后，如此继续，直至比较最后两个数，将小数放前，大数放后，此时第一趟结束，在最后的数必是所有数中的最大数。重复以上过程， 仍从第一对数开始比较（因为可能由于第 2 个数和第 3 个数的交换，使得第 1 个数不再小于第 2 个数），将小数放前，大数放后，一直比较到最大数前的一对相邻 数，将小数放前，大数放后，第二趟结束，在倒数第二个数中得到一个新的最大数。如此下去，直至最终完成排序。
参考 array1.php array2.php

24、写出 session 的运行机制。

session 创建时，是否会在服务端记录一个 cookie?cookie 里面的内容是什么？
session 机制是一种服务器端的机制，服务器使用一种类似于散列表的结构（也可能就是使用散列表）来保存信息。
当 程序需要为某个客户端的请求创建一个 session 的时候，服务器首先检查这个客户端的请求里是否已包含了一个 session 标识-称为 sessionid，如果已包含一个 sessionid 则说明以前已经为此客户端创建过 session，服务器就按照 sessionid 把这个 session 检索出来使用（如果检索不到，可能会新建一个），如果客户端请求不包含 sessionid，则为此客户端创建一个 session 并且生成一 个与此 session 相关联的 sessionid，sessionid 的值应该是一个既不会重

复，又不容易被找到规律以仿造的字符串，这个 sessionid 将被在本次响应中返回给客户端保存。
保存这个 sessionid 的方式可以采用 cookie，这样在交互过程中浏览器可以自动的按照规则把这个标识发给服务器。一般这个 cookie 的名字都是类似于 SEEESIONID。

由 于 cookie 可以被人为的禁止，必须有其他机制以便在 cookie 被禁止时仍然能够把 sessionid 传递回服务器。经常被使用的一种技术叫做 URL 重写，就是把 sessionid 直接附加在 URL 路径的后面，附加方式也有两种，一种是作为 URL 路径的附加信息，表现形式为 http:// … ../xxx;SEEESIONID=ByOK3vjFD75aPnrF7C2HmdnV6QZcEbzWoWiBYEnLerjQ99zWpBng!-145788764
另 一 种 是 作 为 查 询 字 符 串 附 加 在 URL 后 面 ， 表 现 形 式 为 http:// … ../xxx?SEEESIONID=ByOK3vjFD75aPnrF7C2HmdnV6QZcEbzWoWiBYEnLerjQ99zWpBng!-145788764

为了在整个交互过程中始终保持状态，就必须在每个客户端可能请求的路径后面都包含这个 SEEESIONID。

参考：
session 运行机制:理解 session 机制：
http://bbs.91lamp.com/detail-526-1.html
抛开 cookie 使用 session：
http://www.91lamp.com/html/document/php/200808/16-1800.html
cookie 与 session：
http://www.91lamp.com/html/document/php/200808/16-1797.html

25、Cookie 的原理及使用？
Cookie 是网站保存在浏览器客户端的信息，也就是说保存在访客的机器里的变量，一般随着 HTTP 头发送到服务器端。在 Cookie 生效之后及失效之前，客户每次发 出页面请求的时候（包括 PHP 页面和静态 html 页面），都会把 Cookie 一块发送到服务器，只要我们针对它进行相应的处理，就可以实现变量”追随”。

cookie 可以跨越子域名。
比如我们在 xiaofeicn.com 下面注册个个 cookie，那么可以在 bbs.xiaofeicn.com 上读取到该 cookie。
session 不可以跨越子域名：
比 如 我 们 在 xiaofeicn.com 下 面 注 册 个 个 session ， 那 么 不 可 以 在 bbs.xiaofeicn.com,www.xiaofeicn.com 上读取到该 session。

a． 设置一个 Cookie 变量
设置一个 Cookie 变量，PHP 使用的函数是： int setcookie(string name, string value, int expire,
string path, string domain, int secure);
其中 name 是 Cookie 变量名称标识，你在 PHP 中将可以象使用普通变量名一样来用它

引用 Cookie 变量。value 是 Cookie 变量的初始 值，expire 表示该 Cookie 变量的有效时间；path 为该 Cookie 变量的相关路径；domain 表示 Cookie 变量的网站；secure 则需在 https 的安全传输时才有效。 例如我们要设置一个变量 username，它的值是字符串”bluewind”，我们可以这么写代码： setcookie (“username”,” bluewind”); //这两个参数是 setcookie 必要的。

我们还想给这个变量设置有效时间来限制操作超时等，比如说 10 分钟： setcookie (“username”,” bluewind”, 600000); //有效时间的单位是毫秒。

注意：setcookie 和 header 函数一样，需要放在任何能向客户端输出的语句之前。

b. 销毁一个变量

销毁 Cookie 变量只要将它的 value 设为空（””）就可以了，如想销毁上面那个变量只要再写一次： setcookie (“username” ,””);

就可以了。这常用作安全退出之用。

c. Cookie 的有效范围和生存期

Cookie 的有效范围（也就是说在这个范围的页面都能得到这个 Cookie 变量）默认的是该目录及其子目录，当然你可以用 setcookie 的 path 和 domain 参数进行修改。如果你不对 cookie 的 expire 进行设置（参见 1. 设置一个 Cookie 变量中的例子），那么当你离开网站的页面，cookie 也同时得到自动销毁。http://www.netscape.com/newsref/std/cookie_spec.html 是 cookie 原创者 Netscape 所提供的完整介绍信息。

26、PHP 的意思（英文全称、含义）

php 是 Hypertext Preprocessor 的缩写，php 是一种内嵌 HTML 的脚本语言。PHP 的独特语法混合了 c,java 和 perl 及 PHP 式的新语法。这门语言的的目标是让网页开发人员快速的写出动态的网页。

Personal HomePage tools

Hypertext Preprocessor Hypertext Preprocessor …

27、foo()和@foo()之间有什么区别?

foo() 会执行这个函式，任何解译错误、语法错误、执行错误都会在页面上显示出来。

@foo() 在执行这个函数时，会隐藏所有上述的错误讯息。

很多应用程式都使用 @mysql_connect() 和 @mysql_query 来隐藏 mysql 的错误讯息，我认为这是很严重的失误，因为错误不该被隐藏，你必须妥善处理它们，可能的话解决它们。

28、如何声明一个名为”myclass”的没有方法和属性的类?

```
class myclass
{
}
```

29、如何实例化一个名为”myclass”的对象?

```
$myclass = new myclass;
```

30、你如何访问和设置一个类的属性?

$myclass->username = ‘andy’;

31、GD 库是做什么用的?

GD 函式库用来做什么？

这 个可能是我最喜欢的函式库，自从 PHP 4.3.0 版本后 GD 便内建在 PHP 系统中。这个函式库让你处理和显示各式格式的图档，它的另一个常见用途是制作所图档。GD 以外的另一个选择是 ImageMagick，但这个函式库并不内建于 PHP 之中，必须由系统管理员安装在伺服器上。

MagickWand

32、指出一些在 PHP 输出一段 HTML 代码的办法。

嗯，你可以使用 PHP 中任何一种输出语句，包括 echo、print、printf，大部分人都使用如下例的 echo：

```
echo “My string $variable”;
```

你也可以使用这种方法：

Heredoc

```
echo <<<END
This text is written to the screen as output and this $variable is parsed too. If you wanted you can have <span> HTML tags in here as well.</span> The END; remarks must be on a line of its own, and can’t contain any extra white space.
END;
```

33、下面哪个函数可以打开一个文件，并对文件进行读和写操作? c

(a) fget() (b) file_open() (c) fopen() (d) open_file()

34、下面哪个选项没有将 john 添加到 users 数组中? bd

(a) $users[] = ‘john’;

(b) array_add($users, ‘john’);

(c) array_push($users, ‘john’, ‘andy’);

(d) $users ||= ‘john’;

35、

如何使用下面的类,并解释下面什么意思?

```
class Mymd5
{
    function get_md5($str)
    {
        $str=md5(md5($str).”xingmo”);
        return $str;
    }
}
```

参考 mymd5.php

36、用哪一个函数检测一个变量是否定义过?是否为空的函数是?是否为 NULL?
isset()、 empey、is_null()
要深刻理解这几个的含义。

37、$arr = array(‘james’, ‘tom’, ‘symfony’); 请打印出第一个元素的值 ac
a.echo $arr[0];
b.echo $arr{1};
c. $arr2 = array_shift($arr); echo $arr2;
array_pop

38、请将数组的值用’,’号分隔并合并成字符串输出。如何将一个以’,’隔开的字符串分割成数组?
参考 implode.php 把数组变成字符串
要掌握 implode 和 explode 的用法。

39、$a = ‘abcdef’; 请打印出$a 的第一个字母。
echo $a{0};
echo $a[0]; // 不建议用这种方式
substr($a, 0, 1);
最好是用{}。

40、PHP 可以和 sql server/oracle 等数据库连接吗?
可以。可以用 PDO 连接。

41、请写出 php5 的构造函数和析构函数
function __construct()
{
}
function __destruct()
{
}

42、写一个函数，尽可能高效的，从一个标准 url 里取出文件的扩展名
例如: http://www.sina.com.cn/abc/de/fg.php?id=1 需要取出 php 或 .php
参考 url1.php

43、求两个日期的差数，例如 2007-2-5 ~ 2007-3-6 的日期差数（天数）。
思路 1：先用 strtotime 转换成 unix 时间戳，然后相减，除以 86400.
思路 2：先用 mktime 转换成 unix 时间戳，然后相减，除以 86400.

参考 time1.php

44、请写一个函数，实现以下功能：
字符串”open_door” 转换成 “OpenDoor”、”make_by_id” 转换成 “MakeById”。
思路：
1)将’_’替换成’ ‘;
2)使用 ucwords()将各单词首字母大写；
3)去掉空格;str_replace()
rename_for_val.php

46、如果模板是用 smarty 模板。怎样用 section 语句来显示一个名为$data 的数组。比如：
$data = array(
0 => array(‘id’=>8, ‘name’=>’name1′),
1 => array(‘id’=>10, ‘name’=>’name2′),
2 => array([id]=15 [name]=’name3′)
)
写出在模板页的代码？ 若用 foreach 语句又要怎样显示呢？
section 和 foreach 循环二维数组的基本功。section 和 foreach 循环一维数组？section 和 foreach 循环三维数组？
Section 句式
{section name=customer loop=$data }
id: {$data[customer]}
{/section}

47、不用新变量直接交换现有两个变理的值。
考算法的基本功。
$a = ‘welcome’;
$b = ‘beijing’;
$a = ‘|’.$a.’|’.’|’.$b.’|’; //|welcome||beijing|
$b = str_replace(‘|’.$b.’|’, ”, $a);
$b = trim($b, ‘|’);

$a = str_replace(‘|’.$b.’|’, ”, $a);
$a = trim($a, ‘|’);
echo $a;
echo $b;

$a=5;$b=3;怎么交换两个数字？
$a=$a+$b; // 8
$b=$a-$b; //5
$a=$a-$b; //3

PHP 数字金额转中文大写格式，同时说明思路(考数组掌握)。

15001.25 壹万伍仟零壹元贰角伍分
$daixi = array (‘零’,’ 壹’,”);
$str{0}

写一个函数，能够遍历一个文件夹下的所有文件和子文件夹。
Readdir opendir scandir
参考 dir.php

50、表单中 get 与 post 提交方法的区别?
a、 Get 方法通过 URL 请求来传递用户的数据，将表单内各字段名称与其内容，以成对的字符串连接，置于 action 属性所指程序的 url 后，如 http://www.domain.com/test.asp?name=51js&password=51js，数据都会直接显示在 url 上，就像用户点击一个链接一样；Post 方法通过 HTTP post 机制，将表单内各字段名称与其内容放置在 HTML 表头(header)内一起传送给服务器端交由 action 属性能所指的程序处理，该程序会通过标准输入(stdin)方式，将表单的数据读出并加以处理

b、 Get 方式需要使用 $_GET 来取得变量的值；而 Post 方式通过 $_POST 来访问提交的内容

c、 Get 方式传输的数据量非常小，一般限制在 2 KB 左右，但是执行效率却比 Post 方法好；而 Post 方式传递的数据量相对较大，它是等待服务器来读取数据，不过也有字节限制，这是为了避免对服务器用大量数据进行恶意攻击。可在 php.in 中 对 post_max_size 进行设置。

建议：除非你肯定你提交的数据可以一次性提交，否则请尽量用 Post 方法

d、 Get 方式提交数据，会带来安全问题，比如一个登陆页面，通过 Get 方式提交数据时，用户名和密码将出现在 URL 上，如果页面可以被缓存或者其他人可以访问客户这台机器，就可以从历史记录获得该用户的帐号和密码，所以表单提交建议使用 Post 方法；Post 方法提交的表单页面常见的问题是，该页面如果刷新的时候，会弹出一个对话框。

建 议：出于安全性考虑，建议最好使用 Post 提交数据**********在 B/S 应用程序中，前台与后台的数据交互，都是通过 HTML 中 Form 表单完成的。Form 提供了两种数据传输的方式——get 和 post。虽然它们都是数据的提交方式，但是在实际传输时确有很大的不同，并且可能会 对数据产生严重的影响。虽然为了方便的得到变量值，Web 容器已经屏蔽了二者的一些差异，但是了解二者的差异在以后的编程也会很有帮助的。
Form 中的 get 和 post 方法，在数据传输过程中分别对应了 HTTP 协议中的 GET 和 POST 方法。二者主要区别如下：
a、Get 是用来从服务器上获得数据，而 Post 是用来向服务器上传递数据。
b、 Get 将表单中数据的按照 variable=value 的形式，添加到 action 所指向的 URL 后面，并且两者使用”?”连接，而各个变量之间使用”& amp;”连接；Post 是将表单中的数据放在 form 的数据体中，按照变量和值相对应的方式，传递到 action 所指向 URL。
c、 Get 是不安全的，因为在传输过程，数据被放在请求的 URL 中，而如今现有的很多服务

器、代理服务器或者用户代理都会将请求 URL 记录到日志文件中，然后 放在某个地方，这样就可能会有一些隐私的信息被第三方看到。另外，用户也可以在浏览器上直接看到提交的数据，一些系统内部消息将会一同显示在用户面前。 Post 的所有操作对用户来说都是不可见的。
d、Get 传输的数据量小，这主要是因为受 URL 长度限制；而 Post 可以传输大量的数据，所以在上传文件只能使用 Post（当然还有一个原因，将在后面的提到）。
e、Get 限制 Form 表单的数据集的值必须为 ASCII 字符；而 Post 支持整个 ISO10646 字符集。
f、Get 是 Form 的默认方法。
*.Post 传输数据时，不需要在 URL 中显示出来，而 Get 方法要在 URL 中显示。
*.Post 传输的数据量大，可以达到 2M，而 Get 方法由于受到 URL 长度的限制,只能传递大约 1024 字节.
*.Post 顾名思义,就是为了将数据传送到服务器段,Get 就是为了从服务器段取得数据.而 Get 之所以也能传送数据,只是用来设计告诉服务器,你到底需要什么样的数据.Post 的信息作为 http 请求的内容，而 Get 是在 Http 头部传输的。

51、session 与 cookie 的区别?
session 是服务器端缓存，cookie 是客户端缓存。

cookie 机制采用的是在客户端保持状态的方案，而 session 机制采用的是在服务器端保持状态的方案。

52、PHP 支持的数据类型有八种,以下被支持的有：
string
int
float

Bool

array

NULL
resource
object

A、array
B、floating-point numbers(double)
C、integer
D、date
E、string
[A B C E]
PHP 的变量属于松散数据类型，在计算时动态(dynamic)决定。如果要强制设置变量的数据类型的话，可以利用 settype()
函数。或利用 c 语言的强制转型方式(type casting)。

53、假定要使用 Apache+Php 的配置，并将 php3 编译成 Apache 的一个模块。那麼以下 httpd.conf 文件的语句是必须的：【C】

A、AddModule mod_php3.c

B、LoadModule php3_module libexec/libphp3.so

C、AddType application/x-httpd-php3 .php3

D、setup

E、make install

54、以下程序:

```
<HTML>
<HEAD>
<TITLE></TITLE>
<HEAD>
<BODY>
<?php
$num1 = 15;
$num2 = $num1;
echo “<p>$num2</p>”;
$num2 = &$num1;
$num2 = 20;
echo “<p>$num1</p>”;
?>
</BODY>
</HTML>
```

程序输出为:[ ]

A、15

B、35

C、20

D、5

AC

55、以下程序

```
<?php
$str1 = “01″;
$str1++;
$str1 += 1; //$str1 = $str1 + 1;
echo “<p>\$str1 => $str1</p>”;
?>
```

程序输出为：[ ]

A、$str1 => 01

B、$str1 => 2

C、$str1 => 03

D、$str1 => 3
E、$str1 => 1
D
参考 math1.php

56、全局变量与局部变量

```
$a=1;
sum()
{

echo $a;
}
sum();
```

程序输出为：[ ]
A、1
B、10
C、100
D、1000
E、空值
E

57、PHP 的控制语句

```
<?php
$a = 3;
$b = $a++;
if ($a > $b)
{
echo “a 比 b 大”;
}
elseif ($a == $b)
{
echo “a 等于 b”;
}
else
{
echo “a 比 b 小”;
}
?>
```

输出结果为：[ ]
A、a 比 b 大
B、a 等于 b

C、a 比 b 小
D、”a 比 b 小”
E、无输出
A

58、PHP 对字符串的处理程序
$name=”Jollen”;
echo ‘Name:$name’;
echo “Name:$name”;
输出结果为：[ ]
A、Name:Jollen
Name:Jollen
B、Name:Jollen
Name:$name
C、Name:$name
Name:Jollen
D、Name:$name
Name:$name
E、Name:”Jollen”
Name:Jollen
C
此题考单引号与双引号的基本功。

59、下面建立与 MySQL Server 的连接语法正确的是：[ ]
A、$link=connect(“host_name”,”user_name”,”password”);
B、$link=mysql_connect(“host_name”,”user_name”,”password”);
C、$link=mysqlconnect(“host_name”,”user_name”,”password”);
D、$link=mysql_pconnect(“host_name”,”user_name”,”password”);
E、$link=pconnect(“host_name”,”user_name”,”password”);
BD

60、rawurlencode()的作用是？
按照 RFC 1738 对 URL 进行编码
返 回字符串，此字符串中除了 -_. 之外的所有非字母数字字符都将被替换成百分号（%）后跟两位十六进制数。这是在 RFC 1738 中描述的编码，是为了保护原义字符以免其被解释为特殊的 URL 定界符，同时保护 URL 格式以免其被传输媒体（像一些邮件系统）使用字符转换时弄乱。

与 urlencode()的区别：
urlencode:
返回字符串，此字符串中除了 -_. 之外的所有非字母数字字符都将被替换成百分号（%）后跟两位十六进制数，空格则编码为加号（+）。此编码与 WWW 表单 POST 数据的编码方式是一样的，同时与 application/x-www-form-urlencoded 的媒体类型编码方式一样。由于历史原因，此编码在将空格编码为加号（+）方面与 RFC1738 编码（参见 rawurlencode()）不同。

61、请说明在 php.ini 中 safe_mode 开启之后对于 PHP 系统函数的影响?
safe_mode 是唯一 PHP_INI_SYSTEM 属性，必须通过 php.ini 或 httpd.conf 来设置。要启用 safe_mode，只需修改 php.ini： safe_mode = On 或者修改 httpd.conf，定义目录：

Options FollowSymLinks php_admin_value safe_mode 1

重启 apache 后 safe_mode 就生效了。启动 safe_mode，会对许多 PHP 函数进行限制，特别是和系统相关的文件打开、命令执行等函数。

默认情况下，所有操作文件的函数将只能操作与脚本 UID 相同的文件。

注意：如果在 linux 中启用了 safe_mode，那么如果要在一个目录中创建一个目录，比如要在/upload 中创建一个 20081202，那么/upload 目录所有者必须是 apache 的所有者。

62、PHP5 中魔术方法函数有哪几个，请举例说明各自的用法
__sleep
__wakeup
__toString
__set_state
__construct,
__destruct
__call,
__get,
__set,
__isset,
__unset
__clone
__autoload

What does <? echo count (“123″) ?> print out? D
A) 3
B) False
C) Null
D) 1
E) 0
65、What is the value of $a?
<?php
$a = in_array(’01′, array(’1′)) == var_dump(’01′ == 1);
?>
A) True
B) False
B

66、What is the value of $result in the following PHP code?

```
<?php
function timesTwo($int) {
$int = $int * 2;
}
$int = 2;
$result = timesTwo($int);
?>;
```

Answer: NULL

67、The code below _____ because _____.

```
<?php
class Foo {
?>
<?php
function bar() {
print “bar”;
}
}
?>
```

A) will work, class definitions can be split up into multiple PHP blocks.
B) will not work, class definitions must be in a single PHP block.
C) will not work, class definitions must be in a single file but can be in multiple PHP blocks.
D) will work, class definitions can be split up into multiple files and multiple PHP blocks.

68、When turned on, _____ will _____ your script with different variables from HTML forms and cookies. //D
A) show_errors, enable
B) show_errors, show
C) register_globals, enhance
D) register_globals, inject

69、What will be the output of the following PHP code:

```
<?php
echo count(strlen(“http://php.net“));
?>
```

Answer: 1

71、What is the difference between “print()” and “echo()”?
Answer: print is a function,echo is a language construct

72、写出以下程序的运行结果

```
$aa = null;
$bb = false;
```

```
If($aa= = = $bb)
{
    Echo ‘相同’;
}
Else
{
echo’不相同’;
}
```

zend optimizer 是什么
用优化代码的方法来提高 php 应用程序的执行速度，且可以解密 Zend Guard 加密过后代码，使之能够正常运行

74、有三个 php 文件位于同一目录下，内容为
a.php:——-

```
<?php function fa() { echo “in Function A\n”; }?>
```

b.php:——-

```
<?php include_once ‘a.php’; ?>
<?php function fb() { fa(); echo “in Function B\n”; } ?>
```

c.php:——-

```
<?php include ‘a.php’; ?>
<?php include ‘b.php’; ?>
<?php fa(); fb(); ?>
```

使用浏览器访问 c.php，请问是否存在问题。
如果存在问题，请指出修正方法并写出浏览器查看效果
如果不存在问题，请写出浏览器查看效果

将字符 09 转换成十进制数字。
Intval.php

77、What would the following code print to the browser? Why?
复制内容到剪贴板代码:

```
$num = 10;
function multiply(){
$num = $num * 10;
}
multiply();
echo $num;
```

79、以下哪一个函式可以把浏览器转向到另一个页面？ (B)

A) redir()

这不是一个 PHP 函式，会引致执行错误。

B) header()

这个是正确答案，header() 用来插入卷头资料，可以用来使浏览器转向到另一个页面，例如：header("Location: http://www.search-this.com/");

C) location()

这不是一个 PHP 函式，会引致执行错误。

D) redirect()

这不是一个 PHP 函式，会引致执行错误。

80、isset()和 empty()的区别

两者都是测试变量用的。但是 isset()是测试变量是否被赋值，而 empty()是测试一个已经被赋值的变量是否为空。 如果一个变量没被赋值就引用在 php 里是被允许的,但会有 notice 提示。如果一个变量被赋空值，$foo="" 或者$foo=0 或 者$foo=false,那么 empty($foo)返回真，isset($foo)也返回真，就是说赋空值不会注销一个变量。要注销一个变量，可以用 unset($foo)或者$foo=NULL。

答：session 的运行机制：

用户 A 访问站点 Y，如果站点 Y 指定了 session_start();(以下假设 session_start()总是存在)那么会产生一个 session_id,这个 session_id 一般会以 COOKIE 的形式保存到用户 A（我们可以通过在 php.ini 里设置 session.use_only_cookies 为 1，强制 SESSIONID 必须以 COOKIE 专递。）。这时候 SESSIONID 表现 为 $_COOKIE['PHPSESSID'];(PHPSESSID 可用 session_name()函数来下修改)

用户 A 接着访问，这个 session id($_COOKID['PHPSESSID'])就会在 A 每次访问 Y 的时候传送到站点 Y。

在站点 Y 上，会有这么一个目录，是用来保存 SESSION 的实际数据的。站点 Y 接收到 sessionid,然后通过 session id,来获得与 SESSION 数据的关联，并返回 SESSION 数据。

答：session 与 cookie 的区别：

SESSION 存储在服务器端，用户无法进行修改，相对比较安全，COOKIE 存储在客户端，用户通过手段可以进行修改，相对不安全。

Session 会在一定时间内保存在服务器上，当访问增多，会比较占用服务器资源。

单个 cookie 在客户端的限制是 3k，就是说一个站点在客户端存放的 COOKIE 不能超过 3k。

答：多台服务器如何共享 SESSION：

共享就是每台服务器公用一个，那显然要把这个 session 专门放到一个地方
比如存数据库，每台服务器都调这个数据库里的 session
存 memcache 一样的原理
答:可以，但是需要在传值的时候将 Session_id 写到 URL 中;因为禁用以后再传递参数的时候会以网址的形式参数接到后面传过去！
很多开发中涉及到用户的 Session 验证很保留的问题，这个问题比较有意思，总结了几种方案，只供参考。

[问题提出]

为了满足足够大的应用，满足更多的客户，于是我们架设了 N 台 Web 服务器（N>=2），在多台 Web 服务器的情况下，我们会涉及到一个问题：用户登陆一台服务器以后，如果在跨越到另一台服务器的时候能够继续使用客户的 Session？

（以下描述方案只是针对 Linux/Unix + Apache + MySQL +PHP 的开发架构，当然，也可以扩展到其他平台。）

[问题解决方案]

既然我们的问题已经摆在面前了，那么就要从技术角度去解决问题，给我们的客户更好的体验，总结了几个方案。

1. 写客户端 Cookie 的方式

当用户登陆成功以后，把网站域名、用户名、密码、token、session 有效时间全部采用 cookie 的形式写入到客户端的 cookie 里 面，如果用户从一台 Web 服务器跨越到另一台服务器的时候，我们的程序主动去检测客户端的 cookie 信息，进行判断，然后提供对应的服务，当然，如果 cookie 过期，或者无效，自然就不让用户继续服务了。当然，这种方法的弊端就不言而喻了，比如客户端禁用了 cookie 或者 cookie 被黑客窃取了 呢？

2. 服务器之间 Session 数据同步的方式

假设 Web 服务器 A 是所有用户登陆的服务器，那么当用户验证登陆一下，session 数据就会写到 A 服务器里，那么就可以自己写脚本或者守护进 程来自动把 session 数据同步到其他 Web 服务器，那么当用户跳转到其他服务器的时候，那么 session 数据是一致的，自然就能够直接进行服务无须 再次登陆了。缺点是，可能会速度慢，不稳定，如果是单向同步的话，登陆服务器出现问题，那么其他服务器也无法服务，当然也可以考虑双向同步的问题。

3. 利用 NFS 共享 Session 数据的方式

其实这个方案和下面的 Mysql 方案类似，只是存储方式不一样。大致就是有一台公共的 NFS 服务器（Network File Server）做共享服务器，所有的 Web 服务器

登陆的时候把 session 数据写到这台服务器上，那么所有的 session 数据其实都是保存在这台 NFS 服务器上的，不论用户访问那太 Web 服务器，都要来这台服务器获取 session 数据，那么就能够实现共享 session 数据了。缺点是依赖性太 强，如果 NFS 服务器 down 掉了，那么大家都无法工作了，当然，可以考虑多台 NFS 服务器同步的形式。

（关于 NFS 的经典文章：http://linux.vbird.org/linux_server/0330nfs.php）

4. 利用 Mysql 数据库共享 Session 数据的方式

这个方式与 NFS 的方式类似，也是采用一台 Mysql 服务器做共享服务器，把所有的 session 的数据保存到 Mysql 服务器上，所有 Web 服务器都来这台 Mysql 服务器来获取 Session 数据。缺点也是依赖性太强，Mysql 无法工作了影响所有的 Web 服务器，当然，可以考虑多太 Mysql 数据库来共享 session，使用同步 Mysql 数据的方式。

（Mysql 同步我写过文章：http://blog.csdn.net/heiyeshuwu/archive/2005/10/31/520007.aspx）

5. 使用硬件设备

这个算是比较成熟的解决方案了，使用类似 BIG-IP 的负载设备来实现资源共享，那么就能够又稳定又合理的的共享 Session 了。目前很多门户网站 采用这种方式。缺点很明显了，就是要收费了，硬件设备肯定需要购买成本的，不过对于专业或者大型应用来讲，是比较合理并且值得的。

（关于 BIG-IP 设备：http://www.f5.com.cn/channel.php?channel=product&type=BIG-IP-%D3%A6%D3%C3%C1%F7%C1%BF%B9%DC%C0%ED&id=36）

一、速普**公司

速普软件公司。公司规模 8 人左右，有自己的产品，做企业级 OA。做一套 php 面试题，试题不是很难。面试题和情况大致如下：

1：检测一个变量是否有设置的函数是否?是否为空的函数是?(2 分)

2：echo(),print(),print_r()的区别(3 分)

3：表单中 get 与 post 提交方法的区别?

4：session 与 cookie 的区别?

5：用 PHP 打印出前一天的时间格式是 2015-8-10 22:21:21(2 分)

6：能够使 HTML 和 PHP 分离开使用的模板引擎(1 分)

7：使用哪些工具进行版本控制?(1 分)

8：如何实现字符串翻转?(3 分)

9 : 有一个网页地址, 比如 PHP 开发资源网主页: http://www.baidu.com,如何得到它的内容?($1 分)

10：在 PHP 中 error_reporting 这个函数有什么作用? (1 分)

11：JS 表单弹出对话框函数是?获得输入焦点函数是? (2 分)

12：foo()和@foo()之间有什么区别?(1 分)

13：GD 库是做什么用的? (1 分)

14：写一个函数，能够遍历一个文件夹下的所有文件和子文件夹。

15：写出 SQL 语句的格式：插入 ，更新 ，删除 (4 分)

表名 User Name Tel Content Date

张三 13333663366 大专毕业 2006-10-11

张三 13612312331 本科毕业 2006-10-15

张四 021-55665566 中专毕业 2006-10-15

(a) 有一新记录(小王 13254748547 高中毕业 2007-05-06)请用 SQL 语句新增至表中

(b) 请用 sql 语句把张三的时间更新成为当前系统时间

(c) 请写出删除名为张四的全部记录

总结：大概就那么多，回来之后我发现这套题是那家公司从网上找的，无语啊。。。。大家可以好好做做，链接地址：http://dwz.cn/1xm0F5 。

二、同城帮

之前隶属于 360 公司，是 360 的一个分支，现在慢慢独立出去，属于创业性质的公司，主要是 O2O 方面的，面试内容如下：

1：说说项目（他比较感兴趣 360 老兵那个项目，其他项目没有看），问项目的具体内容，遇到的挑战，具体解决办法，缓存的使用，为什使用 redis，为什么使用独立文件服务器。

2 缓存 ,问了 memcache 与 redis 的区别 ,redis 的优势之处。怎样解决 memcache 命中率低的问题 ,问了在实际项目中 memcache 命中率。是否部署过 redis 服务器。

3 :svn 与 git 的区别 ,让说具体的工作流程 ,使用 git 的好处 ,以及怎样处理冲突，基本的命令写了两个。

4：数据库

一：数据库的存储引擎，myisam 与 innodb 的区别，说出除了这两种外的其他存储引擎。

二：int 与 bigint 的区别，实际使用的时候主键选择哪个？int（10）与 int（11）的区别，var_char 与 char 的区别

三：数据库设计，用户表与登录表分开的好处，

5：php 部分

一：session 与 cookie 的区别

二：分布式怎样解决 session 共享问题（可以从数据库，ccookie 存 session,nosql 方面解决）

三：get 与 post 的区别

四：php __autoload 机制

6：计算机网络

一：三次握手与四次挥手的过程（主要是画图），各个参数的含义

二：http 与 https 的区别，https 怎样保准安全（结合 ssl）

7：数据结构

一：链表的结构体（线性和链式），插入一个元素的操作代码（纸上写）

二：排序部分，时间复杂度，写出一个排序算法。

三：二分查找的思想，时间复杂度

8：开发环境，linux 常用命令，apache 与 ngiax 的区别，平常用的开发环境是 win 还是 linux 等

面试心得：技术面试大约 50 分钟，比较累，，，，还有是问问住宿、交通，毕业、实习时间，到岗时间等，hr 人不错， 面试整体不难，都是常用到的东西，算法要随手能写。

三、微博

一家网络公司，以服务大中华地区与海外华人为己任。比较大，2012 年 11 月新浪注册用户已突破 4 亿。面试内容做了一套卷子，大约 6 张（单面），试题大概如下：

1：为什么想加入新浪，对新浪的印象

2：未来的计划、目标、打算？

3：有没有微博号，昵称是什么，粉丝数多少？

4：填空：(主要写结果)

一、判断 null 的函数

二、判断变量是否存在的函数

三、判断是否为空的函数

5：php 转换 json 为数:组的函数

6：php 得到前一天的日期，格式如下（2015-08-24 10:20）然后写入到文件 /usr/test 中

7：提取 url，要求从"sina"中提取 url 部分（要求使用两种方法）

8：连续子数组的最大和例如输入数组为{1，-2，3，10，-4，7，2，5}最大的子数组{3，10，-4，7，2}和为 18

9：写出常见的 linux 命令功能：top、ps、mv、find、df、cat、chmod、chgrp、grep、wc

10：写出 linux 查看 80 端口的命令

11：有一个 ip 日志文件每行一个 ip，统计某一个 ip 出现的次数

12：数据库设计，有一个发布文章的数据字段：文章 id，文章标题，发表人，类别 id，子类别 id，所属地 id，创建时间，状态

问题：一：划出数据库设计图（表之间的关系），表可以自己添加，说明这样设计的目的，好处

二：写出创建创建文章表的语句，说说选择字段的依据。

三：写出查找最新发表的 10 篇文章的 sql 语句，说说优化的方法。

13：系统设计

场景 :现在要做一个用户日程提醒系统 ,需求如下 ,可以记录用户每天的日程，代办日程到了以后，以短信或者是邮箱的形式通知用户，用户可以查看历史日程，可以管理自己的日程（增删改查），用户有好友，可以查看好友开放出来的日程，可以进行好友聊天。

要求：一：考虑并发性。

二：用户量在千万级别，日程数量在亿，10 亿级别

三：保证安全、首页家在流畅

四：使用缓存服务器

问题：一：你感觉这个系统的难点在什么地方

二：这个系统各个模块之间的关系，

三：这个系统实施的具体细节（缓存，大数据方面）

四：系统的安全性怎样保证

一面（技术面）：

1：我这个系统的数据表设计

2：这个系统当用户比较多的时候可能反应较慢，怎样解决

回 答：利用缓存来解决频繁的数据库查找，用户的好友关系、好友的历史日程，我的历史日程可以缓存，他问当好友关系发生变化的时候怎样处理，一开始我说清除缓 存，下次查询直接查数据库，他说这样不好，这样数据库压力瞬间变大，让我再想想，我说可以清除缓存的后在查数据库恢复缓存，他说不是最好的解决方法，最后 他说可以先更新数据库，然后根据更新的好友关系在更新缓存（少查了一次数据库）。

3：这个系统的首页加载比较慢怎样办？

开始我说，采用页面静态化，他说首页是动态加载，页面静态化没有效果，反而会拖慢系统，文件 io 需要消耗性能，让我想想还有什么办法，我说缓存首页数据，但是要注意缓存更新时间

4：登陆问题，慢慢的系统的登陆比较慢，你知道为什么吗，怎样解决

我 说应该是用户量太大，登陆的 sql 语句优化不好，或者是没有加索引，他说加上索引并且 sql 优化到最好，还是比较慢是什么问题，我说是用户量实在太大了， 超出了 mysql 的单表容量了，他说是的，他又问该怎么解决。我说可以水平拆分，缓解数据增长的压力，他说什么是数据库拆分，水平拆分和垂直拆分有什么区 别我说垂直拆分解决表与表之间的 io 竞争，不解决单表中数据量增长出现的压力，把不向关的表放在其他的服务器上，水平拆分解决单表中数据量增长出现的压 力，不解决表与表之间的 io 争夺，感觉自己不是多懂，他说怎么拆分依据是什么（我对拆分只是了解）我说可以按照时间拆分，他说但是这样的拆分没有一点用， 他说你再想想，为什么这样不行，我说我感觉可以，拆分之后单表数据变小了，查询肯定快啊，他说你登录的时候怎样确定用户在哪个表里面呢？我说在 hash 把 用户与分表的关系 hash 到一个表里面，他说不是最好的解决办法，并且 hash 的麻烦，需要确定那张表才能进行查找。。。他让我再想想，我从存储过程，数 据库缓存来说，最后他说不好使，他说你可以采用用户名分表，用户名（a-z）分 26 个表，登陆的时候一下就可以找到他在的那张表了。这个问题大约谈了 30 分钟，回答的不是多好

5：第三方登录问题

他问我做过第三方登陆没有，我说我自己尝试过，利用 qq 做第三方登录，主要调用的腾讯提供的的接口，他又问我第三方登陆的原理，奥师认证方面的，原理，说说 https 的安全传输原理。

6：接口

他 问我做过接口没有，我说做过，他问接口怎样判断用户登录，我说是生

成 token，他问 token 的生成原理，token 的安全怎样解决，token 被劫持 了该怎样半，有什么好的办法。token 被劫持我不知道该怎么处理，这个可以查一查，反正当时我没想到好办法，我从 token 的生成规则，过期时间，添加 令牌方面回答的，他说还可以。

二面：（技术、项目面试）

1：问我为什么想加入新浪微博

2：说一下项目，他问 360 抗战老兵那个，他访问这个网站，问了系统的测试，开发细节，缓存那块问题，问了 mc 与 redis 的区别，内存管理的不同

3：访问了我的个人网站，说可以下网站的架构和部署

4：问我想找什么样的工作，我说想找网站后台，或者是 app 接口开发的工作

5：问我最有没有学习新的东西，想学什么新东西，我说可以看看 nodjs 想关注 php7

总 结：一面技术面大约 60 分钟，主要问了数据库方面的，感觉自己回答的不是很好，二面也是技术面，主要是问项目(360 老兵那个，大家对这个项目比较敏 感)，面试到 12:30 左右。整体感觉还行，问的都是数据库方面的优化，表拆分，大数据的处理方法等等。最后让我第二天等结果，第二天 hr 打电话说面试通 过，说了实习的事，周一上班，周二办理手续

四、美丽说

国内最大的女性快时尚电子商务平台，致力于为年轻时尚爱美的女性用户提供最流行的时尚购物体验，拥有超过 1 亿的女性注册，面试内容做了一套卷子，试题如下：

1：mysql 的索引是那种数据结构，为什么使用这种数据结构。

2：设计一个数据表，用来存储 url 信息，此表会经常发生插入删除操作，应用场景

是查找某个 url 是否存在，请写出表结构，并加说明。

3：比较单词 a 和 b 判断 b 单词的字母是否都在 a 中。

4：写一个 timer 类，用来统计应用程序的运行时间，并写出调用方法。

5：session 与 cookie 的区别，说明禁用 cookie 后 session 还能不能运行，可以参考（http://www.cnblogs.com/zyf-zhaoyafei/p/4477175.html）

6：新浪微博一年产生的数据量是多少，应该怎么存储这些数据。

7：状态吗：204，304，404 的含义，可以参考

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

8：介绍你了解的开源项目

一面（技术面）：

1：介绍 php 的魔术方法

2：索引的最左原则，可以参考

（http://blog.csdn.net/shangxiaoxue/article/details/7514187）

3：数据库 sql 的优化，哪些 sql 语句执行会忽略索引。

4：策略问题： 4 个人过桥，a 要 1 分钟， b 要 2 分钟，c 要 5 分钟，d 要 10 分钟，两人同时过桥，过桥速度以慢的为准，一人拿手电，过完桥后一个人吧手电送回来，问最快多长时间过完桥。

二面（技术面，问题）：

1：说说项目，360 老兵那个。

2：浏览器输入 url 到页面呈现，经过的过程，

3：mc 与 redis 的区别，内存管理方面。

三面（hr 面）：

hr 是个漂亮的妹子，聊得很开心，主要涉及工作打算，生活住宿方面的，大学学习经历，是否了解美丽说这个公司等等。最后让回去等通知。

总 结：美丽说，公司挺大的，一面感觉还是数据库方面有点吃力，主要是数据库底层方面的讨论不是多好。关于策略类型的问题，一般要不是你见过，基本上打不出 来，但是不能说自己不会，你应该给出自己的见解，即使不完全正确。最后回去等通知，说是还有一个 hr 今天不在，如果觉得可以 26 号会打电话通知我来复 试。。。最后还真给我打电话让我去复试，时间是 27 号。。。。

五、360**

笔试题：

1：写出 PHP 中以 array_ 开头的函数，并说明用法，至少写 6 个

2：mysql 常用的存储引擎，说出 3 个以上，并写出每个应用场景，优缺点

3：redis 与 MC 的区别，写出其中一个的内存管理机制

4：查看 Apache 80 端口的命令

5：用 C 语言实现 php 中的 base64 编码函数

6：用 PHP 写出快速排序

面试问：

1 :微博中 reids 的应用场景 ,redis 的内存分配机制 ,为什么使用 redis 不适用 MC，mc 与 redis 的并发哪个较高

2：array_map 的使用场景，array_merge(),合并数组之后的键名变换情况

3 :php 几个编码函数区别(json_encode, http_build_query, urlencode 等区别）

4：正则表达式中 .* + / 的含义

5：给定一个数字，2 进制转换成 16 进制

六、美*网

笔试题：

1：打印前一天的日期时间格式：2016.03.18 10:15:12，打印上个月的最后一天的日期

2：说出 empty()，isset() array_key_exits();的区别

3：浏览器地址栏输入 www.baidu.com 以后发生的事情，和用到的协议

4：从连接中（ <a href='www.meicai.com/index.html'>）提取出 url

5：PHP 发送强求的方法，有什么不同点

6：写出 10 个 mysql 的字段类型，说明使用场景

7：一个数据表，70 个字段，10G 的数据量，每天 50w 的数据增量，说出有哪些优化的方法

8：写一个定时任务，每天 2 点到 8 点，每隔 10 分钟调用一次 PHP 脚本 /data/www/test.php

9：写出查看 php-fpm 进程数的 linux 命令

10 :写一个函数 ,判断重一副扑克牌中随机抽出的 5 张牌是不是顺子 ,顺子的定义：5 张牌是连续的牌比如 23456

其中 2-10 是原数字，A 是 1，J 是 11，Q 是 12，k 是 13，小王大王是任意数。

七、瓜子二手车

一面（技术面）：

1、介绍一个写过的项目，画出架构和技术点

2、coding 杨辉三角，求 m 行 n 列的值

3、使用正则表达式判断 ip 的合法性，要考虑全面

4、redis 和 MC 的内存管理机制

5、redis 中数据存储方式，各种数据结构应用的场景，redis 做消息队列与专业的消息队列有什么区别和优势

5、最左原则的原理和 sql 的优化机制，什么时候不会用到索引。

6、分条件查询的优化和系统设计与架构

7、你最近学习的新技术，你对架构师的理解，你对 GO 语言的理解，速度快吗？快在哪？

二面（技术面）：

8、一条 sql 语句的优化机制，大数据分页处理

9、矩阵的规律遍历，写出遍历的代码

10、上司与你的关系，怎样保证工作与生活的协调等等

11、抓取 a.html 页面，把含有 a 便签中的 href 全部抓取，如果 url 含有 guazi.com 就把该条 url 标题上 guazi，然后再抓取该链接对应的页面，这样不断的遵化你抓取，你怎样解决死循环问题和页面抓取效率问题。

八、百度外卖

一面（技术面）：

1、说项目，介绍知识点，画结构图

2、数据库最左原则含义用法

3、一个日志文件，打印某一时刻的并发数（记录日志的条数）

4、一个数组按照固定的值排序(按照 id 排序)

```
$a = [   ['abc'] => ['id' => 1, 'value' => 233],
         ['bcd'] => ['id' => 2, 'value' => 463] ]
```

二面（技术面）：

1、说说项目

2、数据库最左原则

3、数据库索引建立的原则

4、数据库引擎，说出特点

5、redis 和 MC 的优缺点，都在什么场景下使用

6、打印日志文件中，出现最多的前 100 个 IP

7、说出 linux 中常用的命令

8、php 和其他语言不同的地方，PHP 中弱类型的实现

9、C 语言中结构体和共用体的区别，每个占用做少内存（二者已给出）

10、有没有自己写的一些作品，或者开源的东西

11、系统设计：百度外卖活动中的秒杀怎么设计，注意点是什么

三面（技术面）：

1、介绍做的项目，画出你做的一个模块的流程和开发关键点

2、有没有学习新的东西，比如新的技术，语言

3、GO 语言看过吗，有什么优点，用在什么地方合适

4、为什么选择百度外卖

5、为什么不愿意留在原公司，离职的原因在哪

四面（技术面）：

1、介绍一个最深刻的项目，说出技术点

2、coding：一个数组 0 - n-1，已经排好序，打印出满足 a[i]+a[j] = c 的所有 i 和 j，说出时间复杂度

3、以后有什么打算，学习的目标或者是自己的规划目标

4、收到的 offer 都有哪些，为什么没有选择这些公司

5、原来后台项目有多少个人，每一个人的技术实力在哪

九、小米科技

一面（技术面）：

1、写一个类，一个单例类，输入两个数组，$arr1, $arr2，每个数组中存有整数(0-9)，

这个类提供两个方法，一个是排序$arr1，一个是把两个数组当成两个大数然后进行相减，返回结果（第一个数组如果是[5, 2, 3, 4] ,相当于数字：5234）

2、讲解做的项目，讲述知识点和自己的贡献。

3、数据库中两个笛卡尔积在 join 中的体现，说出 left join 的中间过程

4、最左原则的意义，数据库索引的使用情况和优化

5、网站打开慢的原因和解决办法

二面（技术面）：

1、讲解项目的架构，redis 在项目中的使用情况

2、为什么不留在原来的平台

3、数据库事务原理，是否使用过

三面（技术面）：

1、数据库优化，索引建立原则

2、数据库事务的含义

3、redis 与 MC 的区别，redis 支持的数据结构有哪些

4、框架使用，优缺点

5、PHP 中的魔术方法

6、是否使用过前端框架，js，jquery 是否学习过

四面（技术面）：

1、为什么来小米网，对我们了解多少

2、给两个数组得到交集，各自的差集

3、给两张表，得到各自的交集和差集

4、数据库索引的字段，哪些会用到索引，该怎样查看。

5、PHP 短标签，怎样开启。结束标签有什么影响 (不能正常解析 PHP 文件)

6、session 和 cookie 的区别于联系，怎样保证集群下的 session 同步的问题

7、对架构师的理解，你认为怎样才能达到架构师

8、div 怎样实现三栏式布局

9、php 各个框架的应用，php 的安全相关

10、数据库的优化，怎样保证大数据量的数据库访问正常

11、数据库集群的特点和数据同步的机制

12、什么是响应式设计

----------一点题外话

面试技巧总结：

1：开始去的时候先面试小点的公司，长经验，熟悉面试的流程！

2：去面试一定要准备好简历、笔、纸！最好买个包带着，这样会比较方便。

3：面试的时候要掌握主动性，尽量把问题往自己会的方向上引导回答。

4：小公司面试一般都是一面，而且面试完毕给你发 offer，急于招人，你可以说回去考虑考虑。

5 :说话一定要客气 ,面试你的技术官都是老手 ,不要显摆自己又多牛逼 ,得不偿失。

6 :多收集面试的信息 ,我们这次出去的这几个人 ,每天都要聚一聚说说面试的情况，分享一下做的面试题，感觉很有收获

7： 对于自己不会的问题不要急于回答，面试官不是想让你一下就回答出来正确答案（这样显得出题的无能），而是想看思考问题分析问题的过程和你自己的想法，即使 自己想的不对也要把观点说出来，在新浪面试就是这样，数据库的拆分平时就没有接触过，只是在网上看过，他问的比较深，所以我就按照自己的理解去说的！

8：大公司面试一般流程比较复杂，三面左右，一面技术面，二面技术项目（简历上的）面，三面一般是 HR 面试（轻松得多）

面试技术总结（php 方向）：

新浪 PHP 面试题

1、cookie、session 的联系和区别，多台 web 服务器如何共享 session？
cookie 在客户端保存状态，session 在服务器端保存状态。但是由于在服务器端保存状态的时候，在客户端也需要一个标识，所以 session 也可能要借助 cookie 来实现保存标识位的作用。
cookie 包括名字，值，域，路径，过期时间。路径和域构成 cookie 的作用范围。cookie 如果不设置过期时间，则这个 cookie 在浏览器进程 存在时有效，关闭时销毁。如果设置了过期时间，则 cookie 存储在本地硬盘上，在各浏览器进程间可以共享。
session 存储在服务器端，服务器用一种散列表类型的结构存储信息。当一个连接建立的时候，服务器首先搜索有没有存储的 session id，如果没有，则建立一个新的 session，将 session id 返回给客户端，客户端可以选择使用 cookie 来存储 session

id。也可以用其他的方法，比如服务器端将 session id 附在 URL 上。

区别：

(1).cookie 在本地，session 在服务器端。

(2).cookie 不安全，容易被欺骗，session 相对安全。

(3).session 在服务器端，访问多了会影响服务器性能。

(4). cookie 有大小限制，为 3K

多服务器共享 session 可以尝试将 session 存储在 memcache 中。

2、http 协议中的 post 和 get 有何区别？

GET 用于获取信息，不应该用于修改信息，pOST 可用于更新修改信息。

GET 可传输数据大小和 URL 有关，而 pOST 没有限定大小，大小和服务器配置有关。

GET 放在 URL 中，因此不安全，而 pOST 传输数据对于用户来说是不可见的，所以相对安全。

在 ajax: post 不被缓存，get 被缓存所以一般在请求结尾加 Math.random();

SERVER 端接受:因为在 submit 提交的时候是按不同方式进行编码的，所以服务端在接受的时候会按照不同的方式进行接受!

编码方式:如果传递数据是非-ASCII,那么 GET 一般是不适应的，所以在传递的时候会做编码处理!

3、reqiure 的 include 都可包含文件，二者的区别何在？

require 在包含文件出错之后返回致命错误，跳出运行，而 include 在包含文件出错之后返回警告信息，继续运行。

4、php 中 web 上传文件的原理是什么，如何限制上传文件的大小？

pHp 上传文件默认大小为 2M，设置上传大小的配置项是 upload_max_filesize,post_max_size 设置一次 pOST 中 pHp 能接收的最大数据量，应该比 upload_max_filesize 大。

5、写一个函数，可以遍历文件夹下的所有文件和文件夹。

不知道我的理解是不是不对，如果是简单的列出文件和文件夹的话，下面两个应该可以。本人水平菜，没发现什么陷阱..考虑了一下，会不会题目是要 求做一个遍历，把子文件夹下的文件都要显示出来，这样的话需要使用递归对文件夹进行显示。对每次 scandir 出来的条目进行 is_dir 判断，是 dir 的话递归进行下一轮的 scandir。

总结：

人开发协同，分段测试，上线。

PHP 基础

1：变量的传值与引用。

2：变量的类型转换和判断类型方法。

3：php 运算符优先级，一般是写出运算符的运算结果。

4：PHP 中函数传参，闭包，判断输出的 echo，print 是不是函数等。

5：PHP 数组，数组函数，数组遍历，预定义数组（**面试必出**）。

6：PHP 面向对象，魔术方法，封装、继承、多态。设计模式，包括（单利、工厂、迭代器、装饰、命令、策略）。

7：正则表达式，每个标号含义，邮箱、网址、标签匹配，正则函数（**面试必出**）。

8：PHP 异常处理（级别，错误日志，控制错误输出）。

9：PHP 时间函数，日期计算函数。

10：文件系统，记录日志、目录、文件的遍历、上传、多方法得到文件扩展名、文件引用方式、引用函数区别。（**面试必出**）。

11：会话控制，主要说原理。session 与 cookie 在分布式应用中出现问题的解决方案。

12：PHP 模板引擎，常用模板引擎特点，MVC 好与不好的地方。

13：PHP 安全处理，过滤函数。

14：XML 的使用。

15：PHP 字符串的处理，包括转义（安全）、编码、截取、定位、与数组间的转换、处理函数等。（**面试必出**）。

16：Socket 编程，各种协议，head 头，curl 参数含义。

17：网络状态码含义，常用（204，304，404，504，502）。

18：Apache 配置文件，PHP 配置文件，各个含义字段的含义。

19：网络各种攻击的名词含义（SQL 攻击、XSS、CSRF、DDos），防止措施。

20：url 的处理函数，得到 url 指定的部分。

Mysql 基础

1：基础 sql 语句书写（一般让写关联和子查询语句）

2：索引的创建，优缺点，最左原则

3：存储引擎，常用的几个，优缺点，差别，原理（**面试必出**）

4：sql 注入的处理方法

5：mysql 处理函数（PHP 中封装的）

6：PDO 的使用方法，为什么使用

7：mysql 的优化，表拆分等

8：事务处理，sql 语句的处理效率等

9：数据表字段的类型，同类型间的区别，改如何选取，int(10)与 int(11)的区别等。

10：数据库索引使用的那种数据结构，画出数据结构

Linux

1：常用命令的使用，vim 编辑器的使用。

2：进程，cpu 等信息的查看命令。

3：文件内查看命令（主要涉及统计信息）。

4：Shell 的使用，命令操作。

NoSql

1：Redis 的应用场景，结合微博业务说出他的具体应用。

2：Redis 与 MC 支持数据的不同点，两者都支持哪些数据结构的存储，写越多越好。

3：Redis 持久化存储的原理，与 Mysql 的应用区别。怎样保持持久化数据与内存数据同步的关系（Redis 同步机制）

4：Redis 与 MC 在并发状态下的性能比较。

5：MC 的内存管理机制，当一个数据需要存储的时候怎样分配内存空间

6：Redis 的内存管理机制，与 MC 有哪些不同点。

开发环境

1：PHP7 中的新特性与废弃的特性

2：为什么要使用 PHP7，PHP7 快在哪里

版本控制

1：git 的使用命令，例如：写出版本回退命令。

2：git 与 svn 的区别。

3：如何进行多分支开发，包括多图，帮助大家记忆 **PHP 部分**

服务器部分

1、HTTP Keep-Alive 的作用

作用：Keep-Alive：使客户端到服务器端的连接持续有效，当出现对服务器的后继请求时，Keep-Alive 功能避免了建立或者重新建立连接。Web 服务器，基本上都支持 HTTP Keep-Alive。

缺点：对于提供静态内容的网站来说，这个功能通常很有用。但是，对于负担较重的网站来说，虽然为客户保留打开的连 接有一定的好处，但它同样影响了性能，因为在处理暂停期间，本来可以释放的资源仍旧被占用。当 Web 服务器和应用服务器在同一台机器上运行 时，Keep- Alive 功能对资源利用的影响尤其突出。

解决：Keep-Alive: timeout=5, max=100

timeout：过期时间 5 秒（对应 httpd.conf 里的参数是：KeepAliveTimeout），max 是最多一百次请求，强制断掉连接。就是在 timeout 时间内又有新的连接过来，同时 max 会自动减 1，直到为 0，强制断掉。

2、php 数组函数常见的那些? (array_merge、in_array 的作用)

PHP 中以 array_开头的数组函数有哪些，并说出使用方法（至少 6 个）

一、数组遍历函数

```
1 list();  //不是真正的函数，而是 PHP 的语言结构，用于给一组变量赋值，仅能用于索引数组
```

```
2 each();  //返回数组当前元素的键值对，并将指针移动到下一个元素位置

3 while(); //可配合 list 或 each 使用：

while(list($key, $value) = each($arr)){each $key, $value; }
```

二、数组内部指针控制

```
1 current();  //读取指针位置的内容

2 key();      //读取当前指针指向内容的索引值

3 next();     //将数组中的内部指针指向下一单元

4 prev();     //将数组内部指针倒回一位

5 end();      //将数组内部指针指向最后一个元素

6 reset();    //将目前指针指向第一个索引位置
```

三、数组键值操作函数

```
1 array_values($arr);       //获得数组的值

2 array_keys($arr);         //获得数组的键名

3 array_flip($arr);         //数组中的值与键名互换（如果有重复前面的会被后面的覆
盖）

4 array_search('PHP',$arr); //检索给定的值，加 true 则是严格类型检查

5 array_reverse($arr);      //将数组中的元素翻转(前后顺序)

6 in_array("apple", $arr);  //在数组中检索 apple

7 array_key_exists("apple", $arr); // 检索给定的键名是否存在数组中

8 array_count_values($arr);        // 统计数组中所有值出现的次数
9 array_unique($arr);              // 删除数组中重复的值
```

四、数组回调函数

```
1 array_filter(); //使用回调函数过滤数组中的元素，如果回调返回 true 则当前的元素被包含到返回数组中

2 array_walk();   //回调函数处理数组，自定义函数要有两个参数，本函数第三个参数可以作为回调第三个参数返回

3 array_map();    //可以处理多个数组，每个数组的长度应该相同，传入数组的个数和回调函数参数个数应该一致
```

二、数组的分段和填充

```
1 array_slice($arr, 0, 3);    //将数组中的一段取出，此函数忽略键名（数组的分段）

2 array_splice($arr, 0, 3, array("black","maroon"));    //将数组中的一段取出，返回的序列从原数组中删除

3 array_chunk($arr, 3, TRUE);   //将一个数组分割成多个，TRUE 为保留原数组的键名（分割多个数组）
```

四、数组与栈，列队

```
1 array_push($arr, "apple", "pear");    //将一个或多个元素压入数组栈的末尾(入栈)，返回入栈元素的个数

2 array_pop($arr);    // 将数组栈的最后一个元素弹出（出栈）

3 array_shift($arr);   //数组中第一个元素移出并返回（长度减 1，其他元素向前移动一位，数字键名改为从零计数，文字键名不变）

4 array_unshift($arr,"a",array(1,2));  //在数组的开头插入一个或多个元素
```

六、数组的排序

```
1 sort($arr);      //由小到大，忽略键名

2 rsort($arr);     //由大到小，忽略键名
3 asort($arr);     //由小到大，保留键名

4 arsort($arr);    //由大到小，保留键名
5 ksort($arr);     //按照键名正序排序

6 krsort($arr);    //按照键名逆序排序
```

七、数组的计算

```
1 array_sum($arr);   //对数组内部的所有元素做求和运算（数组元素的求和）

2 array_merge($arr1, $arr2); //合并两个或多个（相同字符串键名，后面覆盖前面，相同的数字键名，后面的附加到后面）

3

4 array_diff($arr1, $arr2);           //返回差集结果数组

array_diff_assoc($arr1, $arr2, $arr3);  //返回差集结果数组，键名也做比较

5 array_intersect($arr1, $arr2);  //返回交集结果数组

6 array_intersect_assoc($arr1, $arr2);   //返回交集结果数组，键名也做比较
```

八、其他的数组函数

```
1 array_unique($arr);   //移除数组中重复的值，新的数组中会保留原始的键名

2 shuffle($arr);              // 将数组的顺序打乱
```

3、PHP 中几个输出函数 echo，print()，print_r()，sprintf()，var_dump()的区别

1：echo：是语句不是函数，没有返回值，可输出多个变量值，不需要圆括号。不能输出数组和对象，只能打印简单类型(如 int,string)。

2：print：是语句不是函数，有返回值 1 ，只能输出一个变量，不需要圆括号。不能输出数组和对象，只能打印简单类型(如 int,string)。

3：print_r：是函数，可以打印复合类型，例如：stirng、int、float、array、object 等，输出 array 时会用结构表示，而且可以通过 print_r($str,true)来使 print_r 不输出而返回 print_r 处理后的值

4：printf：是函数，把文字格式化以后输出（参看 C 语言）

5：sprintf：是函数，跟 printf 相似，但不打印，而是返回格式化后的文字（该函数把格式化的字符串写写入一个变量中，而不是输出来），其　他的与 printf 一样。

例如：

```
1 $str = "Hello";
2 $number = 123;
3 $txt = sprintf("%s world. Day number %u",$str,$number);
4 //输出：  Hello world. Day number 123
```

6：var_dump()：函数，输出变量的内容、类型或字符串的内容、类型、长度。常用来调试。

可以通过 function_exists('函数名称')进行测试

```
1 var_dump(function_exists('print'));  //bool(false)
2
3 var_dump(function_exists('echo'));  //bool(false)
4
5 var_dump(function_exists('print_r')); //bool(true)
```

4、不用新变量直接交换现有两个变量的值

```
1 1:   3     list($a, $b) = array($b, $a);
4 2:
5     $a = $a . $b;
5     $b = strlen( $b );
6     $b = substr( $a, 0, (strlen($a) - $b ) );
7     $a = substr( $a, strlen($b) );
8 9 3:(必须用一个两个字符串都都不能出现的字符做为分隔符)
10     $a = $b.','.$a ;
```

```
11      $a = explode(',', $a);
12      $b1 = $a[1];
13      $a = $a[0];
1415 4：这个是当两个数都是数字的时候：
16      $a = $a + $b;
17      $b = $a - $b;
18      $a = $a - $b;
1920 5：借助数组
21      $a = array($a,$b);
22      $b = $a[0];
23      $a = $a[1];
```

5、heredoc

Heredoc 在正规的 PHP 文档中和技术书籍中一般没有详细讲述。他是一种 Perl 风格的字符串输出技术。使用 heredoc 技术可以实现界面与代码的准分离，比如 phpwind 模板。heredoc 的语法是用”<<<”加上自己定义成对的标签，在标签范围内的文字视为一个字符串

规则如下：

1、以**<<<End** 开始标记开始，以 End 结束标记结束，**结束标记必须顶头写，不能有缩进和空格，且在结束标记末尾要有分号**。开始标记和开始标记相同，比如常用大写的 EOT、EOD、EOF 来表示，也可以使用其他标记，只要保证**开始标记和结束标记不在正文中出现就行。**

2、位于开始标记和结束标记之间的变量可以被正常解析，**但是函数则不可以**。在 heredoc 中，变量不需要用连接符 . 或 , 来拼接，比如：

```php
$a=2;
$b= <<<EOF
"zyf"$a4 "zyf"
EOF;
echo $b; //结果连同双引号一起输出："zyf"2 "zyf"
```

3、heredoc 常用在输出包含大量 HTML 语法文档的时候。他要比传统的 echo 输出精炼很多，如下所示：

```php
function getHtml()
{
    echo "<html>";
    echo "<head><title>Title</title></head>";
    echo "<body>Content</body>";
    echo "</html>;
}

function getHtml()
{
echo <<<EOT
      <html>
      <head><title>Title</title></head>
      <body>Content</body>
      </html>
EOT;
}
```

6、写个函数来解决多线程同时读写一个文件的问题。

```php
<?php
    $fp = fopen("/tmp/lock.txt","w+");
    if(flock($fp, LOCK_EX)){// 进行排它型锁定
 fwrite($fp,"Write something here\n");
        flock($fp, LOCK_UN);// 释放锁定
    }else{
        echo "Couldn't lock the file !";
    }
    fclose($fp);
?>
```

7、禁掉 cookie 的 session 使用方案，设置 session 过期的方法，对应函数

通过 url 传值，把 session id 附加到 url 上（缺点：整个站点中不能有纯静态页面，因为纯静态页面 session id 将无法继续传到下一页面）

通过隐藏表单，把 session id 放到表单的隐藏文本框中同表单一块提交过去（缺点：不适用<a>标签这种直接跳转的非表单的情况）

直接配置 php.ini 文件,将 php.ini 文件里的 session.use_trans_sid= 0 设为 1,(好像在 win 上不支持）

用文件、数据库等形式保存 Session ID，在跨页过程中手动调用

```
1 第一种  setcookie() 直接用 setcookie 设置 session id 的生命周期。
```

```
3     $lifetime=60; //保存 1 分钟
4     session_start();
5     setcookie(session_name(), session_id(), time()+$lifetime, "/");
6 第二种  session_set_cookie_params()
7     $lifetime=60;//保存 1 分钟
8     session_set_cookie_params($lifetime);
9     session_start();
10    session_regenerate_id(true);
```

11 其中 session_regenerate_id();方法用于改变当前 session_id 的值，并保留 session 中数组的值。参数默认为 false,如果设置为 true 则改变 session_id 的值，并清空当前 session 数组。

8、json 格式数据有哪些特点

JSON 一种轻量级的数据交换格式。它基于 ECMAScript 的一个子集。 JSON 采用完全独立于语言的文本格式，但是也使用了类似于 C 语言家族的习惯（包括 C、C++、C#、Java、JavaScript、Perl、 Python 等）。这些特性使 JSON 成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成(网络传输速率)。

JSON 的结构基于下面两点

1. "名称/值"对的集合 不同语言中，它被理解为对象(object)，记录(record)，结构(struct)，字典(dictionary)，哈希表(hash table)，键列表(keyed list)等

2. 值的有序列表 多数语言中被理解为数组(array)

9、php 获取文件内容的方法，对应的函数

1：file_get_contents 得到文件的内容（可以以 get 和 post 的方式获取），整个文件读入一个字符串中

2：用 fopen 打开 url，以 get 方式获取内容（借助 fgets()函数）

3：用 fsockopen 函数打开 url（可以以 get 和 post 的方式获取），以 get 方式获取完整的数据，包括 header 和 body

4：使用 curl 库获取内容，使用 curl 库之前，需要查看 php.ini，查看是否已经打开了 curl 扩展

10、php 魔术方法与魔术常量

类方法：

1、__construct();

用法说明：具有构造函数的类会在每次创建新对象时先调用此方法，适合在使用对象之前做一些初始化工作。如果子类中定义了构造函数则不会隐式调用其父类的构造函数。要执行父类的构造函数，需要在子类的构造函数中调用 **parent::__construct()**。如果子类没有定义构造函数则会如同一个普通的类方法一样从父类继承。

2、__destruct();

用法说明：析构函数会在到某个对象的所有引用都被删除或者当对象被显式销毁时执行。

方法重载：

3、__call();

用法说明：在对象中调用一个不可访问方法时，__call(); 方法会被调用。

4、__callStatic();

用法说明：用静态方式中调用一个不可访问方法时，__callStatic(); 方法会被调用。

属性重载：(只对类中私有受保护的成员属性有效)

5、__get();

用法说明：读取不可访问属性的值时，__get() 会被调用。

6、__set();

用法说明：在给不可访问属性赋值时，__set() 会被调用。

7、__isset();

用法说明：当对不可访问属性调用 isset() 或 empty() 时，__isset() 会被调用。

8、__unset();

用法说明：当对不可访问属性调用 unset() 时，__unset() 会被调用。

序列化相关：

9、__sleep();

用法说明 :序列化时调用 ,serialize() 函数会检查类中是否存在该魔术方法。如果存在，该方法会先被调用，然后才执行序列化操作。

10、__wakeup();

用法说明：unserialize() 会检查是否存在一个 __wakeup() 方法。如果存在，则会先调用该方法，用在反序列化操作中，例如重新建立数据库连接，或执行其它初始化操作

操作类和对象方法：

11、__toString();

用法说明：方法用于一个类被当成字符串时调用，例如把一个类当做字符串进行输出

12、__invoke()；

用法说明：当尝试以调用函数的方式调用一个对象时，__invoke() 方法会被自动调用。

12、__set_state()；

用法说明：当调用 var_export() 导出类时，此静态 方法会被调用。 本方法的唯一参数是一个数组

13、__clone();

用法说明：当复制完成时，如果定义了 __clone() 方法，则新创建的对象（复制生成的对象）中的 __clone() 方法会被调用，可用于修改属性的值。

14、__autoload();

用法说明：该方法可以自动实例化需要的类。当程序要用一个类但没有被实例化时，改方法在指定路径下查找和该类名称相同的文件。否则报错。

说明：PHP 将所有以 __（两个下划线）开头的类方法保留为魔术方法。所以在定义类方法时，除了上述魔术方法，建议不要以 __ 为前缀。在命名自己的类方法时不能使用这些方法名，除非是想使用其魔术功能。

常量：

__LINK__ //文件中的当前行号。

__FILE__ //文件的完整路径和文件名。如果用在被包含文件中，则返回被包含的文件名。

__DIR__ //文件所在的目录。如果用在被包括文件中 则返回被包括的文件所在的目录，它等价于 dirname(__FILE__)。

__FUNCTION__ //函数名称。自 PHP 5 起本常量返回该函数被定义时的名字（区分大小写）。在 PHP 4 中该值总是小写字母的。

__CLASS__ //类的名称。自 PHP 5 起本常量返回该类被定义时的名字（区分大小写）。在 PHP 4 中该值总是小写字母的。

__METHOD__ //类的方法名（PHP 5.0.0 新加）。返回该方法被定义时的名字（区分大小写）。

__NAMESPACE__ //当前命名空间的名称（大小写敏感）。这个常量是在编译时定义的（PHP 5.3.0 新增）

11、PHP 如何获取客户端的 IP 地址

用$_SERVER 获取的 IP 地址有什么问题？

$_SERVER['REMOTE_ADDR']； 通过全局数组来获得

getenv('REMOTE_ADDR')；通过环境变量来获得

当客户机使用代理的时候获取不到真实的 IP 地址

12、写一个函数，可以遍历文件夹下的所有文件和文件夹。

```
1    function get_dir_info($path){
2          $handle = opendir($path);//打开目录返回句柄
3          while(($content = readdir($handle))!== false){
4                $new_dir = $path . DIRECTORY_SEPARATOR . $content;
5                if($content == '..' || $content == '.'){
6                      continue;
7                }
8                if(is_dir($new_dir)){
9                     echo "<br>目录：".$new_dir . '<br>';
```

```
10                      get_dir_info($new_dir);
11                  }else{
12                      echo "文件：".$path.':'.$content .'<br>';
13                  }
14              }
15          }
16          get_dir_info($dir);
```

13、有 mail.log 的一个文档，内容为若干邮件地址，用’\n’分隔换行。挑选 sina.com 的地址（包括从文件读取、过滤到列印出来）。

思路 1：用正则表达式（比较慢，效率地，不推荐用）

思路 2：cat mail.log | grep sina.com

14、PHP 缓存技术有哪些? tp 是局部还是完全缓存?

1. 全页面静态化缓存，也就是将页面全部生成 html 静态页面，用户访问时直接访问的静态页面，而不会去走 php 服务器解析的流程

2. 页面部分缓存，将一个页面中不经常变的部分进行静态缓存，而经常变化的块不缓存，最后组装在一起显示

3. 数据缓存，通过一个 id 进行请求的数据,将数据缓存到一个 php 文件中,id 和文件是对应的,下次通过这个 id 进行请求时 直接读 php 文件

4. 查询缓存，和数据缓存差不多,根据查询语句进行缓存;

5. 常用的缓存技术有：redis 和 memcache

个人认为 tp 应该是全局缓存 因为：tp 缓存实在本地生成一个 php 文件来存储数据库中读

取出来的数据

15、strlen()与 mb_strlen 的作用与区别

在 PHP 中，strlen 与 mb_strlen 是求字符串长度的函数

PHP 内置的字符串长度函数 strlen 无法正确处理中文字符串，它得到的只是字符串所占的字节数。对于 GB2312 的中文编码，strlen 得到的值是汉字个数的 2 倍，而对于 UTF-8 编码的中文，就是 3 倍（在 UTF-8 编码下，一个汉字占 3 个字节）。

采用 mb_strlen 函数可以较好地解决这个问题。mb_strlen 的用法和 strlen 类似，只不过它有第二个可选参数用于指定字符编码。例如得到 UTF-8 的字符串 *str* 长度，可以用 *mbstrlen(* str,'UTF-8')。如果省略第二个参数，则会使用 PHP 的内部编码。内部编码可以通过 mb_internal_encoding()函数得到。

需要注意的是，mb_strlen 并不是 PHP 核心函数，使用前需要确保在 php.ini 中加载了 php_mbstring.dll，即确保“extension=php_mbstring.dll”这一行存在并且没有被注释掉，否则会出现未定义函 数的问题。

16、写一个函数，尽可能高效的从一个标准 url 中取出扩展名

$arr = parse_url('http://www.sina.com.cn/abc/de/fg.php?id=1');

result=pathinfo(

arr['path']);var_dump($arr);

var_dump($result['extension']);

17、php.ini 中 safe mod 关闭 影响哪些函数和参数，至少写 6 个？

move_uploaded_file()　　exec()

system()　　passthru()

popen()　　fopen()

mkdir()　　rmdir()

rename()　　unlink()

copy()　　chgrp()

chown()　　chmod()

touch()　　symlink()

link()　　parse_ini_file()

set_time_limit()　　max_execution_time mail()

18、一群猴子排成一圈，按 1，2，…，n 依次编号。然后从第 1 只开始数，数到第 m 只,把它踢出圈，从它后面再开始数，再数到第 m 只，在把它踢出去…，如此不停 的进行下去，直到最后只剩下一只猴子为止，那只猴子就叫做大王。要求编程模拟此过程，输入 m、n，输出最后那个大王的编号。

```
1 <? php
2  function fuhuan($allnum, $ti){
```

```
3       $arr = array();
4       for($i = 0; $i < $allnum; $i++){
5           $arr[$i] = $i;
6       }
7 8       $nums = 1;
9       while(count($arr) > 1){
10          foreach ($arr as $key => $value) {
11              if($nums == $ti){
12                  unset($arr[$key]);
13                  $nums = 1;
14              }else{
15                  $nums++;
16              }
17          }
18      }
19      $new_arr = array_values($arr);
20      var_dump($new_arr[0] + 1);
21  }
22  fuhuan(10,10);
```

19、isset() 、empty()与 is_null 的区别

1、当变量未定义时，is_null() 和“参数本身”是不允许作为参数判断的，会报 Notice 警告错误；

2、empty , isset 首先都会检查变量是否存在，然后对变量值进行检测。而 is_null 和 “参数本身”只是直接检查变量值，是否为 null，因此如果变量未定义就会出现错误！

3、isset()：仅当 null 和未定义，返回 false；

4、empty()："、0、"0"、NULL、FALSE、array(),未定义，均返回 true；

5、is_null()：仅判断是否为 null，未定义报警告；

6、变量本身作为参数，与 empty()一致，但接受未定义变量时，报警告；

20、求两个文件的相对路径

```
 getpath('/a/b/c/d/e.php', '/a/d/12/34/c.php');

public function getpath($a, $b)
{
    $aarr = explode('/', $a);
    $barr = explode('/', $b);
    $count = count($barr) - 2;
    $pathinfo = '';
    for($i = 1; $i <= $count; $i++){
        if($aarr[$i] == $barr[$i]){
            $pathinfo .= '../';
        }else{
            $pathinfo .= $barr[$i] . '/';
        }
    }
    echo $pathinfo;
}
```

21、MVC 的优缺点

1、 MVC 的优点

（1）可以为一个模型在运行时同时建立和使用多个视图。变化-传播机制可以确保所有相关的视图及时得到模型数据变化，从而使所有关联的视图和控制器做到行为同步。

（2）视图与控制器的可接插性，允许更换视图和控制器对象，而且可以根据需求动态的打开或关闭、甚至在运行期间进行对象替换。

（3）模型的可移植性。因为模型是独立于视图的，所以可以把一个模型独立地移植到新的平台工作。需要做的只是在新平台上对视图和控制器进行新的修改。

（4）潜在的框架结构。可以基于此模型建立应用程序框架，不仅仅是用在设计界面的设计中。

2、 MVC 的不足之处

（1）增加了系统结构和实现的复杂性。对于简单的界面，严格遵循 MVC，使模型、视图与控制器分离，会增加结构的复杂性，并可能产生过多的更新操作，降低运行效率。

（2）视图与控制器间的过于紧密的连接。视图与控制器是相互分离，但确实联系紧密的部件，视图没有控制器的存在，其应用是很有限的，反之亦然，这样就妨碍了他们的独立重用。

（3）视图对模型数据的低效率访问。依据模型操作接口的不同，视图可能需要多次调用才能获得足够的显示数据。对未变化数据的不必要的频繁访问，也将损害操作性能。

（4） 目前，一般高级的界面工具或构造器不支持 MVC 模式。改造这些工具以适应 MVC 需要和建立分离的部件的代价是很高的，从而造成使用 MVC 的困难。

22、session 与 cookie 的联系和区别（运行机制），session 共享问题解决方案

区别与联系：

使用 session_start()调用 session ,服务器端在生成 session 文件的同时生成 session ID 哈希值和默认值为 PHPSESSID 的 session name，并向客户端发送变量为 PHPSESSID(session name)(默认)值为一个 128 位的哈希值。服务器端将通过该 cookie 与客户端进行交互，session 变量的值经 php 内部系列化后保存在服务器 机器上的文本文件中，和客户端的变量名默认情况下为 PHPSESSID 的 coolie 进行对应交互，即服务器自动发送了 http 头:header('Set-Cookie: session_name()=session_id(); path=/');即 setcookie(session_name(),session_id());当从该页跳转到的新页面并调用 session_start()后,PHP 将检查与给定 ID 相关联的服务器端存贮的 session 数据 ,如果没找到则新建一个数据集。

共享方案：

1：使用数据库保存 session， 使用数据库来保存 session，就算服务器宕机了也没事，session 照样在。

问题：程序需要定制；每次请求都进行数据库读写开销不小，另外数据库是一个单点，可以做数据库的 hash 来解 决这个问题。

2：使用 memcached 来保存 session， 这种方式跟数据库类似，内存存取性能比数据库好很多。

问题：程序需要定制，增加 了工作量；存入 memcached 中的数据都需要序列化，效率较低，断电或者重启电脑容易丢失数据；

3： 通过加密的 cookie，在 A 服务器上登录，在用户的浏览器上添加加密的 cookie，当用户访问 B 服务器时，检查有无 Session，如果没有，就检验 Cookie 是否有效，Cookie

有效的话就在 B 服务器上重建 session。简单，高效， 服务器的压力减小了，因为 session 数据不存在服务器磁盘上。根本就不会出现 session 读取不到的问题。

问题：网络请求占用很多。每次请求时，客户端都要通过 cookie 发送 session 数据给服务器，session 中数据不能太多，浏览器对 cookie 的大

小存在限制。不适合高访问量的情况，因为高访问量的情况下。

23、 正则表达式

匹配中文字符的正则表达式： [\u4e00-\u9fa5]
匹配双字节字符(包括汉字在内)：[^\x00-\xff]
匹配空行的正则表达式：\n[\sl]*\r
匹配 HTML 标记的正则表达式：/<(.*)>.*<\/\1>l<(.*) \/>/
匹配首尾空格的正则表达式：(^\s*)l(\s*$)
匹配 Email 地址的正则表达式：\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*
匹配网址 URL 的正则表达式：
^[a-zA-z]+://(\\w+(-\\w+)*)(\\.(\\w+(-\\w+)*))*(\\?\\S*)?$
匹配帐号是否合法(字母开头，允许 5-16 字节，允许字母数字下划线)：
^[a-zA-Z][a-zA-Z0-9_]{4,15}$
匹配国内电话号码：(\d{3}-l\d{4}-)?(\d{8}l\d{7})?
匹配腾讯 QQ 号：^[1-9]*[1-9][0-9]*$

24、写一个函数得到 header 头信息

```
function getHeader()
{
    $headers = [];
    if (function_exists('getallheaders')) {
        $headers = getallheaders();
    } elseif (function_exists('http_get_request_headers')) {
```

```
        $headers = http_get_request_headers();
    } else {
        foreach ($_SERVER as $key => $value) {
            if(strstr($key, 'HTTP_')) {
                $newk = ucwords(strtolower(str_replace('_', '-', substr($key,
5))));
                $headers[$newk] = $value;
            }
        }
    }

    var_dump($headers);
}
```

----------mysql 部分----------

1、select * from table where (ID = 10) or (ID = 32) or (ID = 22) 让结果按 10, 32, 22 的顺序检索出来？

```
Select *
from user_info
Where (ID IN (10, 32, 22))
order BY FIND_IN_SET(ID, '10, 32, 22')
```

----------linux 部分----------

1、core 文件是什么，有什么用？

core 是 unix 系统的内核。当你的程序出现内存越界的时候，操作系统会中止你的进程，并将当前内存状态倒出到 core 文件中，以便进一步分析。程序员可以通过 core 文件来找出问题所在。它记录了程序挂掉时详细的状态描述。

什么是 core dump Core 的意思是内存, Dump 的意思是扔出来，堆出来。开发和使用 Unix 程序时，有时程序莫名其妙的 down 了，却没有任何的提示(有时候会提示 core dumped). 这时候可以查看一下有没有形如 core.进程号的文件生成， 这个文件便是操作系统把程序 down 掉时的内存内容扔出来生成的，它可以做为调试程序的参考.

core dump 又叫核心转储，当程序运行过程中发生异常，程序异常退出时，由操作系统把程序当前的内存状况存储在一个 core 文件中，叫 core dump。如何使用 core 文件 gdb -c core 文件路径 [应用程序的路径] ,进去后输入 where 回车，就可以显示程序在哪一行当掉的，在哪个函数中.

为什么没有 core 文件生成呢? core 文件的生成跟你当前系统的环境设置有关系，可以用下面的语句设置一下，然后再运行程序便成生成 core 文件.

ulimit -c unlimited core 文件生成的位置一般于运行程序的路径相同，文件名一般为 core.进程号

不用 core 文件，程序出了问题产生信号是否知道？答：内核向进程发信号嘛。

2、共享内存除了文件映射还有什么方式？

共享内存对象映射。

二者有什么区别：

答：内存映射文件是由一个文件到一块内存的映射，使应用程序可以通过内存指针对磁盘上的文件进行访问，其过程就如同对加载了文件的内存的访问，因此内存文件映射非常适合于用来管理大文件。

3、请解释下列 10 个 shell 命令的用途

top、ps、mv、find、df、cat、chmod、chgrp、grep、wc

top 命令是 Linux 下常用的性能分析工具，能够实时显示系统中各个进程的资源占用状况，类似于 Windows 的任务管理器。

ps：查看进程

mv：移动或者更改文件

find：在子目录中搜索匹配的文件

df：linux 中 df 命令参数功能：检查文件系统的磁盘空间占用情况。

cat：把一个或多个文件内容显示到标准输出

chmod：改变文件属性

chgrp：改变用户分组

grep：在文件内进行搜索

wc：命令的功能为统计指定文件中的字节数、字数、行数，并将统计结果显示输出。

4、Linux 文件属性有哪些？（共十位）

-rw-r--r--那个是权限符号，总共是- --- --- ---这几个位。

第一个短横处是文件类型识别符：-表示普通文件；c 表示字符设备（character）；b 表示块设备（block）；d 表示目录 （directory）；l 表示链接文件（link）；后面第一个三个连续的短横是用户权限位（User），第二个三个连续短横是组权限位 （Group），第三个

三个连续短横是其他权限位(Other)。每个权限位有三个权限 ,r(读权限) ,w(写权限) ,x (执行权限) 。如果每个权限位都 有权限存在，那么满权限的情况就是：-rwxrwxrwx；权限为空的情况就是- --- --- ---。

权限的设定可以用 chmod 命令 ,其格式位 :chmod ugoa+/-/=rwx filename/directory。

例如：

一个文件 aaa 具有完全空的权限- --- --- ---。

chmod u+rw aaa (给用户权限位设置读写权限，其权限表示为：- rw- --- ---)

chmod g+r aaa (给组设置权限为可读，其权限表示为：- --- r-- ---)

chmod ugo+rw aaa (给用户 , 组 , 其它用户或组设置权限为读写 , 权限表示为 : - rw- rw- rw-)

如果 aaa 具有满权限- rwx rwx rwx。

chmod u-x aaa (去掉用户可执行权限，权限表示为：- rw- rwx rwx)

如果要给 aaa 赋予制定权限- rwx r-x r-x，命令为：

chmod u=rwx，go=rx aaa

5、linux 查询命令

1：find / -name "文件名"　在目录结构中搜索文件，并执行指定的操作。

2：grep

3：local 文件名 ---他是 'find -name' 的另一种写法，但要比后者快得多，原因在于它不搜索具体目录，而是搜索一个数据库（/var/lib/locatedb），这个数据库中含 有本地所有文件信息。Linux 系统自动创建这个数据库，并且每天自动更新一次，所以改命令查不到最新变动过的文件。为了避免这种情况，可以在使用 locate 之前，先使用 updatedb 命令，手动更新数据库。

4. whereis ---是定位可执行文件、源代码文件、帮助文件在文件系统中的位置。whereis 命令只能用于程序名的搜索，而且只搜索二进制文件（参数-b）、man 说明文件（参数-m）和源代码文件（参数-s）。如果省略参数，则返回所有信息。

5：which 作用是在 PATH 变量指定的路径中，搜索某个系统命令的位置，并且返回第一个搜索结果。也就是说，使用 which 命令，就可以看到某个系统命令是否存在，以及执行的到底是哪一个位置的命令。

----------服务器部分

1、Apache 与 Nginx 的优缺点比较

1、nginx 相对于 apache 的优点：

轻量级，比 apache 占用更少的内存及资源。高度模块化的设计，编写模块相对简单

抗并发，nginx 处理请求是异步非阻塞，多个连接（万级别）可以对应一个进程，而 apache 则是阻塞型的，是同步多进程模型，一个连接对应一个进程，在高并发下 nginx 能保持低资源低消耗高性能

nginx 处理静态文件好，Nginx 静态处理性能比 Apache 高 3 倍以上

2、apache 相对于 nginx 的优点：

apache 的 rewrite 比 nginx 的 rewrite 强大 ，模块非常多，基本想到的都可以找到 ，比较稳定，少 bug ，nginx 的 bug 相对较多

3：原因：这得益于 Nginx 使用了最新的 epoll（Linux 2.6 内核）和 kqueue（freebsd）网络 I/O 模型，而 Apache 则使用的是传统的 select 模型。目前 Linux 下能够承受高并发访问的 Squid、Memcached 都采用的是 epoll 网络 I/O 模型。 处理大量的连接的读写，

Apache 所采用的 select 网络 I/O 模型非常低效。

2、cgi 与 fastcgi 的区别

cgi 在 2000 年或更早的时候用得比较多， 以前 web 服务器一般只处理静态的请求，web 服务器会根据这次请求的内容 然后会 fork 一个新进程来运行外部 c 程序（或 perl 脚本...），这个进程会把处理完的数据返回给 web 服务器，最后 web 服务器把内容发送给用户，刚才 fork 的进程也随之退出。 如果下次用户还请求改动态脚本，那么 web 服务器又再次 fork 一个新进程，周而复始的进行。

后来出现了一种更高级的方式是， web 服务器可以内置 perl 解释器或 php 解释器。 也就是说这些解释器做成模块的方式，web 服务器会在启动的时候就启动这些解释器。 当有新的动态请求进来时，web 服务器就是自己解析这些 perl 或 php 脚本，省得重新 fork 一个进程，效率提高了。

fastcgi 的方式是，web 服务器收到一个请求时，他不会重新 fork 一个进程（因为这个进程在 web 服务器启动时就开启了，而且不会退 出），web 服务器直接把内容传递给这个进程（进程间通信，但 fastcgi 使用了别的方式，tcp 方式通信），这个进程收到请求后进行处理，把结果返回 给 web 服务器，最后自己接着等待下一个请求的到来，而不是退出。

fastcgi 跟 cgi 的区别是：

	在 web 服务器方面	**在对数据进行处理的进程方面**
cgi	fork 一个新的进程进行处理	读取参数，处理数据，然后就结束生命期

fastcgi　用 tcp 方式跟远程机子上的进程或本地进程建立连接　　要开启 tcp 端口，进入循环，等待数据的到来，处理数据

举个例子: 服务端现在有个 10 万个字单词， 客户每次会发来一个字符串，问以这个字符串为前缀的单词有多少个。 那么可以写一个程序，这个程序会建一棵 trie 树，然后每次用户请求过来时可以直接到这个 trie 去查找。 但是如果以 cgi 的方式的话，这次请求结束后这课 trie 也就没了，等下次再启动该进程时，又要新建一棵 trie 树，这样的效率就太低下了。　而用 fastcgi 的方式的话，这课 trie 树在进程启动时建立，以后就可以直接在 trie 树上查询指定的前缀了。

3、select, poll 和 epoll 的区别

select

select 最早于 1983 年出现在 4.2BSD 中，它通过一个 select()系统调用来监视多个文件描述符的数组，当 select()返回后，该数组中就绪的文件描述符便会被内核修改标志位，使得进程可以获得这些文件描述符从而进行后续的读写操作。

select 目前几乎在所有的平台上支持，其良好跨平台支持也是它的一个优点，事实上从现在看来，这也是它所剩不多的优点之一。

select 的一个缺点在于单个进程能够监视的文件描述符的数量存在最大限制，在 Linux 上一般为 1024，不过可以通过修改宏定义甚至重新编译内核的方式提升这一限制。

另外，select()所维护的存储大量文件描述符的数据结构，随着文件描述符数量的增大，其复制的开销也线性增长。同时，由于网络响应时间的延迟 使得大量 TCP 连接处于非活跃状态，但调用 select()会对所有 socket 进行一次线性扫描，所以这也浪费了一定的开销。

poll

poll 在 1986 年诞生于 System V Release 3 ,它和 select 在本质上没有多大差别 ,但是 poll 没有最大文件描述符数量的限制。

poll 和 select 同样存在一个缺点就是，包含大量文件描述符的数组被整体复制于用户态和内核的地址空间之间，而不论这些文件描述符是否就绪，它的开销随着文件描述符数量的增加而线性增大。

另外，select()和 poll()将就绪的文件描述符告诉进程后，如果进程没有对其进行 IO 操作，那么下次调用 select()和 poll() 的时候将再次报告这些文件描述符 ,所以它们一般不会丢失就绪的消息，这种方式称为水平触发（Level Triggered）。

epoll

直到 Linux2.6 才出现了由内核直接支持的实现方法，那就是 epoll，它几乎具备了之前所说的一切优点，被公认为 Linux2.6 下性能最好的多路 I/O 就绪通知方法。

epoll 可以同时支持水平触发和边缘触发（Edge Triggered，只告诉进程哪些文件描述符刚刚变为就绪状态，它只说一遍，如果我们没有采取行动，那么它将不会再次告知，这种方式称为边缘触发），理论上边缘触发的性能要更高一些，但是代码实现相当复杂。

epoll 同样只告知那些就绪的文件描述符，而且当我们调用 epoll_wait()获得就绪文件描述符时，返回的不是实际的描述符，而是一个代表 就绪描述符数量的值，你只需要去 epoll 指定的一个数组中依次取得相应数量的文件描述符即可，这里也使用了内存映射（mmap）技术，这样便彻底省掉了 这些文件描述符在系统调用时复制的开销。

另一个本质的改进在于 epoll 采用基于事件的就绪通知方式。在 select/poll 中，进程只有在调用一定的方法后，内核才对所有监视的文件描 述符进行扫描，而 epoll 事先通过

epoll_ctl()来注册一个文件描述符，一旦基于某个文件描述符就绪时，内核会采用类似 callback 的回调 机制 ,迅速激活这个文件描述符 ,当进程调用 epoll_wait()时便得到通知。

4、Memcache 和 Redis 区别

1. Redis 中，并不是所有的数据都一直存储在内存中的，这是和 Memcached 相比一个最大的区别。
2. Redis 在很多方面具备数据库的特征，或者说就是一个数据库系统，而 Memcached 只是简单的 K/V 缓存。
3. 他们的扩展都需要做集群；实现方式：master-slave、Hash。
4. 在 100k 以上的数据中，Memcached 性能要高于 Redis。
5. 如果要说内存使用效率，使用简单的 key-value 存储的话，Memcached 的内存利用率更高，而如果 Redis 采用 hash 结构来做 key-value 存储，由于其组合式的压缩，其内存利用率会高于 Memcached。当然，这和你的应用场景和数据特性有关。
6. 如果你对数据持久化和数据同步有所要求，那么推荐你选择 Redis，因为这两个特性 Memcached 都不具备。即使你只是希望在升级或者重启系统后缓存数据不会丢失，选择 Redis 也是明智的。
7. Redis 和 Memcache 在写入性能上面差别不大，读取性能上面尤其是批量读取性能上面 Memcache 更强
8. 1.nginx 使用哪种网络协议?
   nginx 是应用层 我觉得从下往上的话 传输层用的是 tcp/ip 应用层用的是 http
   fastcgi 负责调度进程

   2. <? echo 'hello tusheng' ; ?> 没有输出结果，可能是什么原因，简述的解决此问题的过程(提示：语法没有问题)

可能服务器上面没有开启短标签 short_open_tag =设置为 Off，,php.ini 开启短标签控制参数： short_open_tag = On

3. 简述下面程序的输出结果，简要说明为什么，如何解决这类问题?

```
<?php
$tmp = 0 == "a"? 1: 2;
echo $tmp;
?>
```

结果 1 int 和 string 类型强制转换造成的，0==="a"

0 == 0 肯定是 true 啊
PHP 是弱类型。。
$tmp = 0 === "a"? 1: 2;
echo $tmp; 这样就是 2

4. 已知一个字符串如下: $str = "1109063 milo 1";
用一行代码将该字符串里面的 1109063 赋值给$uid, milo 赋值给$user, 1 赋值给$type

空格如下

```
list($uid, $user, $type) = explode(" ", $str);
```

\t 如下

```
list($uid, $user, $type) = explode("\t", $str);
```

```
list($uid, $user, $type) = sscanf($str, "%d %s %d");
```

```
$n = sscanf($auth, "%d\t%s %s", $id, $first, $last);
```

5. 分别列出如下类型的有符号和无符号范围 TINYINT SMALLINT MEDIUMINT INT

TINYINT-2^7 - 2^7-10 ~ 2^8-1
SMALLINT-2^15 - 2^15-1 0 ~ 2^16-1
MEDIUMINT-2^23 - 2^23-1 0 ~ 2^24-1
INT-2^31 - 2^31-1 0 ~ 2^32-1

6. 将下面的数组用一行拼装成一个字符串 i am milo! day day up!

```
<?php
$arr = array(
'I', 'AM', 'MILO!', 'DAY', 'DAY', 'UP!'
);
?>


$str = strtolower(implode(" ",$arr));
```

7. 调用如下函数获取函数并获取 count 的值

```
<?php
function get_list($cnd = array(), &$count = false)
{
// 伪代码 处理$cnd 并赋值 datas
$datas = 'i am call back';
$count && $count = rand(1, 10000);
return $datas;
}
?>


$count=1;
$data = get_list($cnd,&$count);
echo $count;
```

8. 几种方式去取代 session 机制，简单描述各自的优劣

mysql、memcache、cookie 保持一种唯一状态标识码

9. 下列 HTTP 状态码出现的可能原因，如何处理
200, 301, 404, 502, 503

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，并且将来任何对此资源的引用都应该使用本响应返回的若干个 URI 之一。如果可能，拥有链接编辑功能的客户端应当自动把请求的地址修改为从服务器反馈回来的地址。除非额外指定，否则这个响应也是可缓存的。 新的永久性的 URI 应当在响应的 Location 域中返回。除非这是一个 HEAD 请求，否则响应的实体中应当包含指向新的 URI 的超链接及简短说明。 如果这不是一个 GET 或者 HEAD 请求，因此浏览器禁止自动进行重定向，除非得到用户的确认，因为请求的条件可能因此发生变化。 注意：对于某些使用 HTTP/1.0 协议的浏览器，当它们发送的 POST 请求得到了一个 301 响应的话，接下来的重定向请求将会变成 GET 方式。

404

请求失败，请求所希望得到的资源未被在服务器上发现。没有信息能够告诉用户这个状况到底是暂时的还是永久的。假如服务器知道情况的话，应当使用 410 状 态码来告知旧资源因为某些内部的配置机制问题，已经永久的不可用，而且没有任何可以跳转的地址。404 这个状态码被广泛应用于当服务器不想揭示到底为何请 求被拒绝或者没有其他适合的响应可用的情况下。

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。这个状况是临时的，并且将在一段时间以后恢复。如果能够预计延迟时间，那么响应中可以包含一个 Retry-After 头用以标明这个延迟时间。如果没有给出这个 Retry-After 信息，那么客户端应当以处理 500 响应的方式处理它。
注意：503 状态码的存在并不意味着服务器在过载的时候必须使用它。某些服务器只不过是希望拒绝客户端的连接。

200 OK 一切正常，对 GET 和 POST 请求的应答文档跟在后面。
301 Moved Permanently 客户请求的文档在其他地方，新的 URL 在 Location 头中给出，浏览器应该自动地访问新的 URL
404 Not Found 无法找到指定位置的资源。这也是一个常用的应答。
502 Bad Gateway 服务器作为网关或者代理时，为了完成请求访问下一个服务器，但该服务器返回了非法的应答。
503 Service Unavailable 服务器由于维护或者负载过重未能应答。例如，Servlet 可能在数据库连接池已满的情况下返回 503。服务器返回 503 时可以

提供一个 Retry-After 头。

10. 有如下数据库，用原生态 mysql 扩展去连接并查询 user 表的前十行
host: 192.168.0.254
port: 3306
user: one
pass: piece
database: db_user
table: user

```
$link = mysql_connect("192.168.0.254:3306","one","piece") or die('Could not connect: '.mysql_error());
mysql_select_db('db_user',$link);
$query = mysql_query("select * from user limit 10");
while($rs = mysql_fetch_array($query,MYSQL_ASSOC))
{}
```

11. 用 autoload($class) 实现 Lib 目录下的类的自动加载并可以兼容子目录

```
$request->action = lcfirst(implode(array_map(
'ucfirst',
explode('-', strtolower($request->action))
)));
-------------------------------------------------------------
function __autoload($class)
{
$cls = strtolower(str_replace("_","/",$class));

if(file_exsits(LIB.$cls.'.php'))
{
include_once(LIB.$cls.'.php');
}
else
{
die("not found {$class} class");
}
}
defined("LIB",'/data/wwwroot/www.xx.com/lib/');
$author = new Lib_Author();
-------------------------------------------------------------
function __authload($class)
```

```
{
$cls = explode("_",$class);
if(@is_dir($cls[1]))
{
if(@is_file($cls[2]))
{
include_once("CON_PATH".$cls[1].'/'.$cls[2].".php");
}
else
{
dir('error');
}
}
else if(@is_file($cls[1].".php"))
{
include_once("CON_PATH".$cls[1].".php");
}
else
{
dir('error');
}
}
---------------------------------------
function __autoload($class)
{
$cls = explode("_",$class);
$file = get_file($cls);
if($file=='error')
{
die('error');
}
include_once($file);
}
function get_file($dir)
{
if(is_array($dir))
{
foreach($dir as $k=>$v)
{
$tmpdir .= $v.'/';
if(is_dir('CON_PATH'.$tmpdir))
{
continue();
}
```

```
else if(is_file('CON_PATH'.$tmpdir.".php"))
{
return 'CON_PATH'.$tmpdir.".php";
}
else
{
return 'error';
}
}
return 'error';
}
return 'error';
}


defined("CON_PATH","/data/wwwroot/www.xx.com/app/cntroller/");
$sb = new controller_sb();
-----------------------------------
function __autoload_my_classes($classname)
{
# ... your logic to include classes here
}
spl_autoload_register('__autoload_my_classes');
------------------------------------------------------------
12. 用 set_error_handle 去捕获错误并输出，级别自己定
set_error_handle(callback,level)
function callback(int $errno , string $errstr [, string $errfile [, int $errline [,
array $errcontext ]]] ){
}


function dealErrorHandler($errno,$errstr,$errfile,$errline)
{
switch($errno){
case E_USER_ERROR:
echo "error [$errno] $errstr fatal error on line $errline in file $errfile";
break;
case E_USER_WARNING:
echo "my warning [$errno] $errstr":
break;
case E_USER_NOTICE:
echo "my notice[$errno] $errstr";
break;
default:
```

```
echo "unkonwn error type :[$errno] $errstr";
break;
}
}
set_erro_handler(dealErrorHandler);


trigger_error("notice", E_USER_NOTICE);
trigger_error("warning", E_USER_WARNING);
trigger_error("error", E_USER_ERROR);
```

13. 简述两种屏蔽 php 程序的 notice 警告的方法

初始化变量，文件开始设置错误级别或者修改 php.ini 设置 error_reporting set_error_handler 和 @抑制错误

1.在程序中添加：error_reporting (E_ALL & ~E_NOTICE);

2.或者修改 php.ini 中的：error_reporting = E_ALL

改为：error_reporting = E_ALL & ~E_NOTICE
3.error_reporting(0);或者修改 php.inidisplay_errors=Off

14. instanceof 的作用，经常在什么设计模式中使用

单例模式,但是其他的模式也会用到

15. 1023 用二进制表示，并简述计算过程

10-2
1023%2=1
511%2 =1
255%2 =1

127%2 =1
63%2 =1
31%2 =1
15%2 =1
7%2 =1
3%2 =1
1%2 =1
0 =0

1023
2^9=<N<2^10
511

k=9
10 9 8 7 6 5 4 3 2 1
1 1 1 1 1 1 1 1 1 1

1023 1
1023-1/2=511 1
511-1/2=255 1
255-1/2=127 1
127-1/2=63 1
63-1/2=31 1
31-1/2=15 1
15-1/2=7 1
7-1/2=3 1
3-1/2=1 1

2-10

只需用将二进制数的各个位上的数从最右边开始，最右边的第一个数乘以二的零次方，第二个数乘以二的一次方，第三个数乘以二的二次方，依次类推可得第 n 个数乘以二的（n-1）次方，然后把得到的结果相加即可
例如：110011=1*2^0+1*2^1+0*2^2+0*2^3+1*2^4+1*2^5=51
这也可以算是一个公式就是 An*2^(n-1) An 表示二进制数最右边开始的第 n 个数，

将第一项第二项第三项一直到第 n 项用式子 An*2^(n-1) 计算出来并加在一起即可

16. 下面 php 程序输出的内容是什么？为什么？

```
<?php
$str = "aa\tbb\tcc";
@list($a, $b, $c) = explode('\t', $str);
echo $a,$b,$c;
?>
```

aabbcc;//'\t'不会以\t 切割字符串,explode 之后申城一个 array(0=>"aa\tbb\tcc")所以。。。,'\t'换成"\t"就被切割

17. include 和 require 分别返回什么错误级别

include 会系统警告并继续执行,require 会发出系统警告但是会引致致命错误令脚本终止运行

18. 现有一个函数，有不确定多少个的参数(可能有 5 个也可能有 50 个)，如何去定义这个函数

方法一：不借助 php 内置函数

方法二：提示 func_num_args() func_get_arg() unc_get_args()

```
function param()
{
$numargs = func_num_args();
echo "Number of arguments: $numargs<br />\n";
if ($numargs >= 2) {
echo "Second argument is: " . func_get_arg(1) . "<br />\n";
}
$arg_list = func_get_args();
for ($i = 0; $i < $numargs; $i++) {
echo "Argument $i is: " . $arg_list[$i] . "<br />\n";
}
}

param(1,2,3,4,5);
```

```
/**
2 *  例子写完后，本来认为完事了，结果遇到有人问 call_user_func_array(), 看了一下手册
3 *  原来，我上面的那个 test 函数还可以精简成如下的例子，
4 */
5 function otest1 ($a)
6 {
7 echo( '一个参数' );
8 }
9
10 function otest2 ( $a,$b)
11 {
12 echo( '二个参数' );
13 }
14
15 function otest3 ( $a,$b,$c)
16 {
17 echo( '三个啦' );
18 }
19
20 function otest ()
21 {
22 $args=func_get_args();
23 $num=func_num_args();
24 call_user_func_array( 'otest'.$num,$args );
25 }
26
27 otest(1,2);
```

9.

19. 在一个函数(该函数没有 return 语句)里面去处理全局变量，并且改变他的值，用两种方法去实现(global 和引用&)

```
$var=1;
function get_pra()
{
global $var;
$var = 'xxx';
echo $var;
}
echo $var.'--';
get_pra();
```

```
echo $var;
---------------------------------
$test = 1;
$test1 = 2;
function get_yinyong()
{
global $test1;
$GLOBALS["test"] = &$test1;
}
echo $test."\n";
get_yinyong();
echo $test;
---------------------------
```

20. 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，不能使用 sql 函数以及 order by 等语句

表 user 字段 uid, username

估计一个 user 表中的区间,在此区间用 php 去一个随机数,sql 语句大于或者小于此 id 去 limit 几十条(保证 10 条数据),再不够散乱的话,取出来的数据 shuffle 函数打乱数组,array_rand 随即取出 10 个

21. 假设下面的 sql 语句里面的 uid 都能获取到具体值，经过下面语句查询后 uid 的顺序是什么，如何去按照 uid in 输入的顺序去排序

select uid from user where uid in(10, 1, 3, 8, 11, 4, 7);

可观的结果是 1,3,4,7,8,10,11 升序,有种情况特殊就是不确定因为中间的某些 id 人为直接修改可能不是升序了，如果按照 uid in 的顺序需要重新循环一次根据 id 获取查询结果数组中的值放进新数组中即可

22. 用 PHP 将一个字符串中的字母替换成**

preg_replace('/[a-zA-Z]*/','**',$str);
如果指定的字符就可以 str_replace('ooxx','**',$str);

23. 下面 2.php 中打印结果是什么？为什么？执行顺序 1.php->2.php

cookie,cookie 时间就有问题 time()+3600

24. 简述 php 常用的 json 编码函数，如何将 json 解码的时候返回数组

25. mysql 在 sql 语句中有' / 等词的时候，要对 sql 语句的每个具体值做些什么处理

mysql_real_escape_string

26. 如何在 php 中设置 header 头信息

header('');

27.有如下几个脚本，请问 2.php 的输出结果

1.php

```
<?php
setcookie('test', 'cookie_test', 3600);
?>
2.php
<?php
$cookie = isset($_COOKIE['test'])? $_COOKIE['test']: 'cookie';
echo $cookie;
?>
```

i am here

1

总结

a.如果 include 或 include_once 不是在函数或方法中被调用，则输出结果均一样。

b.如果 include 或 include_once 在函数或方法中被调用，则如果想让第二次及以后调用时有结果，则必须用 include，而不能用 include_once，这一点一定要注意。

28. 简述 call_user_func 的功能

调用函数或者类里面的函数，返回第一个参数的值。类似的功能 call_user_func_array

29. 访假设 nginx 已经配置 server_name www.120.net xxx.120.net 访问问 http://www.120.net/index.php 和 http://xxx.120.net/index.php 之后 $_SERVER["SERVER_NAME"] 和 $_SERVER["REQUEST_URI"]分别是什么

www.120.net xxx.120.net

/index.php /index.php

30. linux 下某文件的属性为 drwxr-xr-x 用数字表示其权限是

目录权限为 755 所有者 u 拥有读写修改权限所属组 g 拥有读、修改权限所属组之外 o 的拥有读和修改权限

31. 宽带的 1Mbps 理论上的下载速度是多少 KBps, 计算的方法

1*1024/8

1M=1024KB
1KB=1024B
1B=8bit

第二部分
1. 简单实现一个单例+工厂的设计模式 1 私有静态成员变量
2 __CLASS__获取当前类名
3 公共静态方法获取单例
4 覆盖__clone()方法

----十个字：私有静态量，公共静态法--------

2. 例举几个常用的魔术方法, 并说明作用? 如何在打印一个对象的时候展示我们自定义的内容?

魔术函数

1。__construct()
实例化对象时被调用，
当__construct 和以类名为函数名的函数同时存在时，__construct 将被调用，另一个不被调用。

2。__destruct()
当删除一个对象或对象操作终止时被调用。

3。__call()
对象调用某个方法，
若方法存在，则直接调用；
若不存在，则会去调用__call 函数。

4。__get()
读取一个对象的属性时，
若属性存在，则直接返回属性值；
若不存在，则会调用__get 函数。

5。__set()
设置一个对象的属性时，
若属性存在，则直接赋值；
若不存在，则会调用__set 函数。

6。__toString()
打印一个对象的时被调用。如 echo $obj;或 print $obj;

7。__clone()
克隆对象时被调用。如：$t=new Test();$t1=clone $t;

8。__sleep()
serialize 之前被调用。若对象比较大，想删减一点东东再序列化，可考虑一下此函数。

9。__wakeup()

unserialize 时被调用，做些对象的初始化工作。

10。__isset()
检测一个对象的属性是否存在时被调用。如：isset($c->name)。

11。__unset()
unset 一个对象的属性时被调用。如：unset($c->name)。

12。__set_state()
调用 var_export 时，被调用。用__set_state 的返回值做为 var_export 的返回值。

13。__autoload()
实例化一个对象时，如果对应的类不存在，则该方法被调用。

魔术常量

1。__LINE__
返回文件中的当前行号。

2。__FILE__
返回文件的完整路径和文件名。如果用在包含文件中，则返回包含文件名。自 PHP 4.0.2 起，__FILE__ 总是包含一个绝对路径，而在此之前的版本有时会包含一个相对路径。

3。__FUNCTION__
返回函数名称（PHP 4.3.0 新加）。自 PHP 5 起本常量返回该函数被定义时的名字（区分大小写）。在 PHP 4 中该值总是小写字母的。

4。__CLASS__
返回类的名称（PHP 4.3.0 新加）。自 PHP 5 起本常量返回该类被定义时的名字（区分大小写）。在 PHP 4 中该值总是小写字母的。

5。__METHOD__

返回类的方法名（PHP 5.0.0 新加）。返回该方法被定义时的名字（区分大小写）。

3. 类静态方法和实例化类方法比较及优缺点

4. 有一个论坛
threads 表记录主题以及标题等信息
posts 表记录主题内容以及回复内容等信息
threads 表主键为 tid
posts 表主键为 pid, 所属主题标记为 tid
通过 tid 将 threads 和 posts 一对多关联起来
现在数据量 posts 表达到了 1 亿, threads 表 2000 万, 大约一个主题有 5 篇回复
请你设计一下分表, 将 posts 表和 threads 表进行 mysql 分表

5. 现在有一个 mysql 主库/从库, 请问 php mysql 查询的时候怎么在 php 程序中实现主从分离? 主从分离有什么好处配置主从数组文件,自己封转几个 model 函数,查询的加载 slave 配置实例化,破坏数据的操作加载 master 进行实例化优点:并 发负载能力提高,利于数据维护和安全,提高可用性缺点:数据同步有些延迟

6. 简述 UCenter 的单点登录机制
所谓单点登录，无非就是几个站点共用一个用户中心，实现同步登陆，同步退出。

其实最终还是用户去登录，只是采用了 ajax （javascript 利用 src 异步跨域调用）用户不会发现。

而且利用了 p3p 头实现了，不同域名，单点登录(ucenter 用的 cookie)

缺点就是采用 ajax 客服端请求 ，如果有 10 个以上应用，登录速度就慢下来了。

7. linux 相关 有一个包 http://www.120.net/test-1.0.0.tar.gz

a. 将它下载到/usr/local/src

b. 将其源码编译安装到/usr/local/test 目录

c. 他依赖 mysql 包，位于/usr/local/mysql 目录

写出下载编译安装过程

wget - c http://www.120.net/test-1.0.0.tar.gz/usr/local/srctar zxvf /usr/local/src/test-1.0.0.tar.gzcd /usr/local/src/test-1.0.0./configure --prefix=/usr/local/test --exec--prefix=/usr/local/mysqlmake testmake install

8. 使用 php 的 memcache 扩展编写一个获取数据的函数(缓存即将过期超时加锁)

a. 数据超时之后去 mysql 获取，获取完后更新 memcache

b. 去 mysql 获取数据的时候加锁，让一个进程去 mysql 拉数据，其他人返回 memcache 中的数据

public function get_cache($key) { if($this->memcahe) { $var = $this->memcahe->get($this->pre.$key); $valid = $this->memcahe->get($this->pre.$key.'_valid'); if($var && !$valid) { $lock = $this->memcahe->get($this->pre.$key.'_lock'); if(!$lock) { $this->memcahe->set($this->pre.$key.'_lock', true, 0, 60); return false; } } return $var; } return false; }

public function set_cache($key, $var = null, $expire = 0) { if($this->memcahe) { $expire = (int)$expire; $expire = ($expire ? $expire : $this->expire); $this->memcahe->set($this->pre.$key, $var, 0, $expire+300); $this->memcahe->set($this->pre.$key.'_lock', false, 0, $expire); $this->memcahe->set($this->pre.$key.'_valid', true, 0, $expire); return true; } return false; }

9. 简述队列，堆栈的原理

都可以看做是一维数组来操作，队列先进先出，出列只能在列头，进列只能

在列尾，堆栈是后进先出，进栈和出栈都是从栈顶

堆栈的工作原理是什么?

堆 栈是一种抽象数据结构，其操作机理是后进先出。当你把新条目推进堆栈时，已经在堆栈内的任何条目都会压到堆栈的深处。同样的，把一个条目从堆栈移出则会让 堆栈内的其他条目都向堆栈的顶部移动。只有堆栈最顶端的条目能从堆栈中取出，条目离开堆栈的顺序和它们被推进堆栈的顺序一样。你不妨回想下自动售货机的装 货和取货过程就明白了。

10. arrayaccess 定义如下 用它实现一个数组

ArrayAccess {
/* Methods */
abstract public boolean offsetExists (string $offset)
abstract public mixed offsetGet (string $offset)
abstract public void offsetSet (string $offset , string $value)
abstract public void offsetUnset (string $offset)
}
class Single implements ArrayAccess{ private $name; private static $_Instance = null; private function __construct() { } static function load() { if(null == self::$_Instance) { self::$_Instance = new Single(); } return self::$_Instance; } public function setName($name) { $this->name = $name; } public function getName() { return $this->name; } /** * 实现四个方法 * offsetExists()，用于标识一个元素是否已定义 * offsetGet()，用于返回一个元素的值 * offsetSet()，用于为一个元素设置新值 * offsetUnset()，用于删除一个元素和相应的值 **/ public function offsetSet($offset, $value) { if (is_null($offset)) { $this->container[] = $value; } else { $this->container[$offset] = $value; } } public function offsetGet($offset) { return isset($this->container[$offset]) ? $this->container[$offset] : null; } public function offsetExists($offset) { return isset($this->container[$offset]); } public function offsetUnset($offset) { unset($this->container[$offset]); }}$s = Single::load();$s->setName("jack");$s["name"] = "mike";echo

$s->getName(); //jackecho $s["name"]; //mike
11. 假设 coreseek 安装目录为/usr/local/coreseek
配置文件为/usr/local/coreseek/etc/test.conf
索引名为 post
a. 创建索引
b. 启动服务
c. 重建索引(重建过程中保证搜索服务仍然可用)

indexer -c /usr/local/coreseek/etc/test.conf --allsearchd -c /usr/local/coreseek/etc/test.conf indexer -c /usr/local/coreseek/etc/test.conf --all --rotate12. 假设您有一张 posts 帖子表 对该表进行 sphinx 增量准实时索引，描述你的方案
使用“主索引+增量索引”方法有个简单的实现，在数据库中增加一个计数表，记录每次重新构建主索引时，被索引表的最后一个数据 id，这样在增量索引时只需要索引这个 id 以后的数据即可，每次重新构建主索引时都更新这个表。

13. php 代码：
$i = 97;$a = ($i++) + (++$i) + $i ;$b = (--$i) + ($i--) + $i + 6;

echo "$i, $a, $b";输出结果是什么

97, 295, 299
97
97+99+99
98+98+97+6

14. 以下代码，用于取得客户端 IP: if(getenv('HTTP_CLIENT_IP') && strcasecmp(getenv('HTTP_CLIENT_IP'), 'unknown')) { $onlineip = getenv('HTTP_CLIENT_IP');}
elseif(getenv('HTTP_X_FORWARDED_FOR') && strcasecmp(getenv('HTTP_X_FORWARDED_FOR'), 'unknown'))

{ $onlineip = getenv('HTTP_X_FORWARDED_FOR');}
elseif(getenv('REMOTE_ADDR') &&
strcasecmp(getenv('REMOTE_ADDR'), 'unknown')) { $onlineip =
getenv('REMOTE_ADDR');} elseif(isset($_SERVER['REMOTE_ADDR'])
&& $_SERVER['REMOTE_ADDR'] &&
strcasecmp($_SERVER['REMOTE_ADDR'], 'unknown')) { $onlineip =
$_SERVER['REMOTE_ADDR'];}但是以 HTTP_开始的请求 header 均属于客户端可以伪造的信息，在反向代理环境下，如何保证 PHP 不会接收到伪造的 HTTP_CLIENT_IP, HTTP_X_FORWARDED_FOR 值？

15. 例如 google,baidu 等大型网站，当使用不同客户端（如手机和 PC 机）访问同样的 URL 时，呈现的页面却不相同，这是何原理？ 如果能给出实际解决方案，可加分。

简单的可以用 user_agent 判断,但是及其初步

可以的话通过服务器或者手机终端特征或者 wap 网关 accept 信息等

16. 生产环境 php.ini 中 magic_quotes_gpc 及 magic_quotes_runtime 值应该设置为什么？onoff17. PHP 调用远程 http 接口时可使用 file_get_contents,但当远程主机不可达或响应过慢，会导致本地 PHP 进程被长时间挂起，从而影响本地服务器稳定性，如何避免超时时，PHP 进程长时间被挂起?

file_get_contents 可以设置下超时时间$ctx =
stream_context_create(array('http' => array('timeout' => 1
)
)
);
file_get_contents("http://www.want.com/", 0, $ctx);
curl 实现获取远程 http 接口也可以，同样需要设置超时时间
curl_setopt($s,CURLOPT_TIMEOUT,$timeout);
18. 同上题，如何避免 DNS 查询过慢导致超时？19. mysql 字符集 set names * 命令设置哪几个系统变量的值？（ACE） A、Character_set_client B、Character_set_system C、Character_set_results D、Character_set_server E、Character_set_connection F、Character_set_database20. 以下哪种校对规则不区分大小写？（A） A、utf8_general_ci B、utf8_general_cs C、utf8_general_bin21. 如何杜绝 XSS 攻击?
strip_tags 可以初步过滤,也可以自己写过滤函数针对特殊标签进行处理,用 ascii 码进行替换 23. 如何杜绝 CSRF 攻击?
在 Web 应用程序侧防御 CSRF 漏洞，一般都是利用 referer、token 或者验证码，tokenf 方式还是比较可信