**[PHP面试题目搜集](http://www.cnblogs.com/strick/p/4961765.html)**

搜集这些题目是想在学习PHP方面知识有更感性的认识，单纯看书的话会很容易看后就忘记。

曾经看过数据结构、设计模式、HTTP等方面的书籍，但是基本看完后就是看完了，没有然后了，随着时间的推移，也就渐渐忘记了。

所以通过这次搜集题目，可以复习复习一些知识点。而且自己动手写写，动脑思考，能更好的记住这些知识点。

留份记录也方便以后自己查看。题目基本都来自网络，如果有不正确的地方欢迎指出或补充。

**一、理论知识**

**1.1、PHP里面是区分大小写的吗？**

PHP对于系统函数、用户自定义函数、类名称等是不区分大小写的

PHP中的变量和常量是区分大小写的

对于文件名又因服务器操作系统而定，linux中区分，Win不区分

**1.2、$\_POST，$HTTP\_RAW\_POST\_DATA 和 php://input 的区别？**

**$\_POST：**

是获取表单POST过来数据，media type是“application/x-www-form-urlencoded”。

意思就是字段名和值都编码了，每个 key-value 对使用 ‘&’ 字符分隔开，key 和 value 使用 ‘=’ 分开，其他特殊字符都会被使用 urlencode 方式进行编码。

**$HTTP\_RAW\_POST\_DATA：**

可以获取原始的POST数据，但需要在 php.ini 中设置开启，并且不支持 enctype="multipart/form-data"方式传递的数据

**php://input：**

可以获取原始的 POST 数据，并且比$HTTP\_RAW\_POST\_DATA更少消耗内存，也不支持"multipart/form-data"，

可以使用 [file\_get\_contents()](http://php.net/manual/zh/function.file-get-contents.php)函数去获取它的内容

**1.3、没有通过static定义的方法，能否通过”对象名::方法名“这样的形式来调用？**

会产生一个strict错误，但在会继续执行代码。参考《[PHP静态调用非静态方法](http://www.cnblogs.com/yjf512/archive/2012/09/12/2682556.html)》。这篇文章里面还讲到了一个概念“[calling scope](http://www.laruence.com/2012/06/14/2628.html)”。

静态调用并不是说有::就是静态调用，而是看calling scope。$this指针指向的对象就是这个方法被调用时刻的calling scope。

**1.4、请简单阐述您最得意的开发之作**

**1.5、对于大流量的网站,您采用什么样的方法来解决各页面访问量统计问题**

a. 确认服务器是否能支撑当前访问量。

b. 优化数据库访问。(参考3.5)

c. 禁止外部访问链接（盗链）, 比如图片盗链。

d. 控制文件下载。

e. 使用不同主机分流。

f. 使用浏览统计软件，了解访问量，有针对性的进行优化。

**1.6、请介绍Session的原理**

因为HTTP是无状态的，所以一次请求完成后客户端和服务端就不再有任何关系了，谁也不认识谁。

但由于一些需要（如保持登录状态等），必须让服务端和客户端保持联系，session ID就成了这种联系的媒介了。

当用户第一次访问站点时，PHP会用session\_start()函数为用户创建一个session ID，这就是针对这个用户的唯一标识，

每一个访问的用户都会得到一个自己独有的session ID，这个session ID会存放在响应头里的cookie中，之后发送给客户端。

这样客户端就会拥有一个该站点给他的session ID。

当用户第二次访问该站点时，浏览器会带着本地存放的cookie(里面存有上次得到的session ID)随着请求一起发送到服务器，

服务端接到请求后会检测是否有session ID，如果有就会找到响应的session文件，把其中的信息读取出来；如果没有就跟第一次一样再创建个新的。

参考《[Session原理简述](http://www.pureweber.com/article/how-session-works/)》

**1.7、session共享问题解决方案**

a. 客户端Cookie保存，以cookie加密的方式保存在客户端，每次session信息被写在客服端，然后经浏览器再次提交到服务器。

b. 服务器间Session同步，使用主-从服务器的架构，当用户在主服务器上登录后，通过脚本或者守护进程的方式，将session信息传递到各个从服务器中

c. 使用集群统一管理Session，当应用系统需要session信息的时候直接到session群集服务器上读取，目前大多都是使用Memcache来对Session进行存储。

d. 把Session持久化到数据库，目前采用这种方案时所使用的数据库一般为mysql。

参考《[Session共享实现方案调研](http://chenzhou123520.iteye.com/blog/1647186)》

**1.8、测试php性能性能的工具，和找出瓶颈的方法。**

XHProf（[windows安装方法参照这里](http://www.cnblogs.com/strick/p/4699981.html)）是一个分层PHP性能分析工具。它报告函数级别的请求次数和各种指标，包括阻塞时间，CPU时间和内存使用情况。

它有一个简单的HTML的用户界面，基于浏览器的性能分析用户界面能更容易查看，也能绘制调用关系图。[参数值查看](http://koda.iteye.com/blog/463891" \t "_blank)。

**1.9、介绍一下常见的SSO(单点登陆)的原理。**

SSO是一种统一认证和授权机制，通过一个应用中的安全验证后，再访问其他应用中的受保护资源时，不再需要重新登录验证。

解决了用户只需要登录一次就可以访问所有相互信任的应用系统，而不用重复登录。

统一的认证系统是SSO的前提之一。认证系统的主要功能是将用户的登录信息和用户信息库相比较，对用户进行登录认证；

认证成功后，认证系统应该生成统一的认证标志（ticket），返还给用户。另外，认证系统还应该对ticket进行效验，判断其有效性。

参考《[单点登录SSO](http://www.cnblogs.com/yupeng/archive/2012/05/24/2517317.html)》

**1.10、您研究过的PHP框架的特点，主要解决什么问题，与其他框架的不同点。**

**1.11、禁掉cookie的session使用方案**

a. 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）

b. 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）

c. 直接配置php.ini文件,将php.ini文件里的session.[use\_trans\_sid](http://php.net/manual/zh/session.configuration.php#ini.session.use-trans-sid)= 0设为1（好像在win上不支持）

d. 用文件、数据库等形式保存Session ID，在跨页过程中手动调用

**1.12、PHP缓存技术有哪些?**

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. 内存式缓存，redis和memcache

参考《[PHP中9大缓存技术总结](http://www.imooc.com/article/1478)》

**1.13、JSON格式数据有哪些特点**

a. JSON一种轻量级的数据交换格式。它基于ECMAScript的一个子集。

b. 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）

c. 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)。

d. "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

e. 值的有序列表，多数语言中被理解为数组(array)

参考《[介绍JSON](http://www.json.org/json-zh.html)》

**1.14、isset() 、empty()与is\_null的区别**

[isset()](http://php.net/manual/zh/function.isset.php)：仅当null和未定义，返回false

[empty()](http://php.net/manual/zh/function.empty.php)：""、0、"0"、NULL、FALSE、array(),未定义，均返回false

[is\_null()](http://php.net/manual/zh/function.is-null.php)：仅判断是否为null，未定义 报警告

**1.15、MVC的优缺点**

**优点：**

分散关注、松散耦合、逻辑复用、标准定义

a. 开发人员可以只关注整个结构中的其中某一层，方便多开发人员间的分工

b. 可以很容易的用新的实现来替换原有层次的实现

c. 降低层与层之间的依赖

d. 利于各层逻辑的复用，有利于标准化

e. 对单元测试的支持更加出色

**缺点：**

a. 清晰的构架以代码的复杂性为代价， 对小项目优可能反而降低开发效率

b. 降低了系统的性能，例如业务造访数据库现在必须通过中间层来完成

c. 控制层和表现层有时会过于紧密，导致没有真正分离和重用

d. 有时会导致级联的修改，如果在表示层中增加一个功能，为保证符合分层式结构，可能需要在相应的控制层和模型层中都增加相应的代码

**1.16、PHP中单引号和双引号有什么区别？哪个速度更快？**

单引号更快

单引号内的数据不会被解析(任何变量和特殊转义字符)，所以速度更快

双引号要先查找语句中是否有变量，双引号内的数据会被解析，如变量($var)值会代入字符串中，特殊转义字符也会被解析成特定的单个字

这里顺便说个定界符（heredoc syntax）“<<<”，

其作用就是按照原样，包括换行格式什么的，输出在其内部的东西，任何特殊字符都不需要转义，变量会被正常的用其值来替换

**1.17、简述GBK，GBK2312，BIG5，GB18030**

GB2312支持的汉字较少，GBK是相比GB2312汉字更为丰富，包括全部的中日韩汉字

GB18030相比GBK增加了一些少数民族汉字汉字库更为多样，常人很少用

一般简体中文使用GBK而繁体中文使用BIG5

**1.18、接口与抽象类的区别是什么？**

**抽象类：**

抽象类是不能被实例化的类，只能作为其他类的父类来使用，抽象类是通过关键字abstract来声明

抽象类与普通类类似，都包含成员变量和成员方法，两者的区别在于，抽象类中至少包含一个抽象方法

抽象方法没有方法体，该方法天生就是要被子类重写的

抽象方法的格式为：abstract function abstractMethod()

子类继承抽象类使用extends

**接口：**

接口是通过interface关键字来声明，接口中的成员常量和方法都是public的，方法可以不写关键字public

接口中的方法也是没有方法体的，接口中的方法也是天生要被子类实现的

接口能实现多继承

子类实现接口使用implements

**1.19、谈谈对mvc的认识**

MVC(Model/View/Controller)模式包括三类对象。

Model是应用对象，View是它在屏幕上的表示，Controller定义用户界面对用户输入的响应方式

模型（Model）模型是应用程序的主体部分。模型表示业务数据，或者业务逻辑

视图（View） 视图是应用程序中用户界面相关的部分，是用户看到并与之交互的界面

控制器（controller) 控制器工作就是根据用户的输入，控制用户界面数据显示和更新model对象状态

**1.20、php中传值与传引用的区别**

按值传递：函数范围内对值的任何改变在函数外部都会被忽略

引用传递：函数范围内对值的任何改变在函数外部也能反映出这些修改

**1.21、php5的构造函数和析构函数**

\_\_construct：这个函数将被当成是一个构造函数并在建立一个对象实例时被执行

\_\_destruct：PHP将在对象被销毁前调用这个函数. 它称为析构函数

**1.22、PHP的垃圾收集机制是怎样的？**

PHP是一门托管型语言，在PHP编程中程序员不需要手工处理内存资源的分配与释放，这就意味着PHP本身实现了垃圾回收机制（Garbage Collection）

PHP使用的回收算法是[引用计数法](http://php.net/manual/zh/features.gc.refcounting-basics.php)，每个php变量存在一个叫"zval"的变量容器中。

一个zval变量容器，除了包含变量的类型和值，还包括两个字节的额外信息。第一个是"is\_ref"，是个bool值，用来标识这个变量是否是属于引用集合(reference set)。

第二个额外字节是"refcount"，用以表示指向这个zval变量容器的变量(也称符号即symbol)个数。

把一个变量赋值给另一变量将增加引用次数(refcount)

当任何关联到某个变量容器的变量离开它的作用域(比如：函数执行结束)，或者对变量调用了函数 unset()时，"refcount"就会减1

变量容器在"refcount"变成0时就被销毁

**1.23、说几个你知道的设计模式？**

[**单例模式：**](http://www.phpddt.com/php/getinstance.html)保证一个类仅有一个实例，并提供一个访问他的全局访问点例如框架中的数据库连接

[**简单工厂模式：**](http://www.phpddt.com/php/php-factory.html)它具有创建对象的某些方法，可以使用工厂类创建对象，而不直接使用 new。例如初始化数据库的时候会用到，比如MySQL，MSSQL

[**策略模式：**](http://www.phpddt.com/php/strategy.html)针对一组算法，将每一个算法封装到具有共同接口的独立的类中，例如进入个人主页时，根据浏览者的不同，给予不同的显示与操作

[**注册模式：**](http://www.phpddt.com/php/registry-pattern.html)提供了在程序中有条理的存放并管理一组全局对象 (object)，例如ZF框架中的Zend\_Registry::set

[**适配器模式：**](http://www.phpddt.com/php/design-adapter.html)将不同接口适配成统一的API接口，例如数据操作有mysql、mysqli、pdo等，可利用适配器模式统一接口

[**观察者模式：**](http://www.phpddt.com/php/observer.html)一个对象通过添加一个方法使本身变得可观察。当可观察的对象更改时，它会将消息发送到已注册的观察者。例如实现实现消息推送

[**装饰器模式：**](http://www.phpddt.com/php/design-decoration.html)不修改原类代码和继承的情况下动态扩展类的功能，例如框架的每个Controller文件会提供before和after方法

[**迭代器模式：**](http://www.runoob.com/design-pattern/iterator-pattern.html)提供一个方法顺序访问一个聚合对象中各个元素，在PHP中将继承[Iterator](http://php.net/manual/zh/class.iterator.php)类

[**原型模式：**](http://www.runoob.com/design-pattern/prototype-pattern.html)实现了一个原型接口，该接口用于创建当前对象的克隆。当直接创建对象的代价比较大时，则采用这种模式。例如，一个对象需要在一个高代价的数据库操作之后被创建。

**二、PHP编码**

**2.1、mysqli\_real\_connect()和mysqli\_connect()有啥区别？**

a. [mysqli\_real\_connect()](http://php.net/manual/zh/mysqli.real-connect.php) 要求一个由 [mysqli\_init()](http://php.net/manual/zh/mysqli.init.php) 创建的有效的对象

b. mysqli\_real\_connect() 可以与 [mysqli\_options()](http://php.net/manual/zh/mysqli.options.php) 一同使用来设置连接的不同选项

c. mysqli\_real\_connect() 有一个flag参数

**2.2、遍历目录及子目录中的文件**

这里可以用[scandir()](http://php.net/manual/zh/function.scandir.php)，或者[glob()](http://php.net/manual/zh/function.glob.php)函数，这里有篇文章介绍了《[四种方法](http://www.phppan.com/2009/10/use-php-traversal-file/)》，[在线代码中有两种方法](http://ideone.com/mq6N54)。

**2.3、 给任意一段URL，取出该URL中包含的扩展名。如“http://www.pwstrick.com/test.php?somevar”，返回.php或php**

使用了5种方法，[pathinfo()](http://php.net/manual/zh/function.pathinfo.php)，[explode()](http://php.net/manual/zh/function.explode.php)，[basename()](http://php.net/manual/zh/function.basename.php)、[strpos()](http://php.net/manual/zh/function.strpos.php)跟[substr()](http://php.net/manual/zh/function.substr.php)的组合，

使用正则（可以参考我以前的一篇文章《[JavaScript与PHP中正则](http://www.cnblogs.com/strick/p/4958606.html)》）以及[parse\_url()](http://php.net/manual/zh/function.parse-url.php)。[查看在线代码](http://ideone.com/kGnLQc" \t "_blank)。

**2.4、PHP怎样防止SQL注入？**

使用预处理语句和参数化查询。预处理语句和参数分别发送到数据库服务器进行解析，参数将会被当作普通字符处理。

这种方式使得攻击者无法注入恶意的SQL。 你有两种选择来实现该方法：[PDO](http://php.net/manual/zh/pdo.prepared-statements.php)和[MySQLI](http://php.net/manual/zh/class.mysqli-stmt.php)，[查看在线代码](http://ideone.com/BPCF1z)。参考《[PHP中该怎样防止SQL注入](http://blog.jobbole.com/67875/)》

**2.5、include、require、include\_once与require\_once的区别**

a. include和require都是引入指定的文件。\_once表示只引入一次，即之前已经引入过的不再引入。

b. 加载失败的处理方式不同：

include在引入不存文件时产生一个警告且脚本还会继续执行，包含进来。通俗点儿讲就是：带上她！

require则会导致一个致命性错误且脚本停止执行，有赖于这个文件。通俗点儿讲就是：我要她！

c. include是有条件包含函数，而 require则是无条件包含函数。

d. include()执行时需要引用的文件每次都要进行读取和评估，require()执行时需要引用的文件只处理一次(实际上执行时需要引用的文件内容替换了require()语句)

e. include有返回值，而require没有。参考《[PHP中include require include\_once require\_once 的区别](http://liuzhichao.com/p/1743.html)》

**2.6、写出一些php魔术方法**

　　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，建议不要以 \_\_ 为前缀，[查看PHP手册](http://php.net/manual/zh/language.oop5.magic.php)。

**2.7、shell命令方式向PHP传入参数的三种方式**

使用$argc $argv；使用[getopt](http://php.net/manual/zh/function.getopt.php)函数()；提示用户输入，然后获取输入的参数。参考《[PHP传入参数](http://www.qttc.net/201303288.html)》

**2.8、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 $a 相对于$b 的相对路径应该是 "../../12/34/c.php"**

 先将两个字符串用“/”分割成数组，再用[array\_diff\_assoc](http://php.net/manual/zh/function.array-diff-assoc.php)先检查$a与$b数组的差集。然后再$b与$a做差集。[查看在线代码](http://ideone.com/Slb84k" \t "_blank)。

**2.9、用php写出显示客户端IP、服务器端IP的代码和网页地址**

客户端IP：[$\_SERVER](http://php.net/manual/zh/reserved.variables.server.php)["SERVER\_ADDR"]

服务器端IP：$\_SERVER["REMOTE\_ADDR"]

网页地址：$\_SERVER["REQUEST\_URI"]

当前脚本的执行路径：$\_SERVER["SCRIPT\_FILENAME"]或者\_\_FILE\_\_

当前脚本的名称:$\_SERVER["PHP\_SELF"]或者$\_SERVER["SERIPT\_NAME"]

链接到前一页的URL地址：$\_SERVER["HTTP\_REFERER"]

**2.10、error\_reporting(2047)什么作用**

[error\_reporting](http://php.net/manual/zh/function.error-reporting.php)，设置应该报告何种 PHP 错误，这里是指显示所有的错误 E\_ALL

**2.11、echo，print()，print\_r()，printf()，sprintf()，var\_dump()有什么区别**

[echo](http://php.net/manual/zh/function.echo.php)：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)

[print](http://php.net/manual/zh/function.print.php)：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)。

[print\_r](http://php.net/manual/zh/function.print-r.php)：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

[printf](http://php.net/manual/zh/function.printf.php)：是函数，把文字格式化以后输出（参看C语言）

[sprintf](http://php.net/manual/zh/function.sprintf.php)：是函数，跟printf相似，但不打印，而是返回格式化后的文字，其他的与printf一样。

[var\_dump](http://php.net/manual/zh/function.var-dump.php)：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。

**2.12、$a = 1; $x =&$a; $b=$a++; 求$b和$x的值**

$b=1，$x=2

**2.13、php中对数组序列化和反序列化的函数，把utf-8转换成gbk的函数**

[serialize](http://php.net/manual/zh/function.serialize.php)，[unserialize](http://php.net/manual/zh/function.unserialize.php)，[iconv](http://php.net/manual/zh/function.iconv.php)("utf-8","gbk",$strs)

**2.14、strlen()与mb\_strlen的作用分别是什么？**

[strlen()](http://php.net/manual/zh/function.strlen.php)无法正确处理中文字符串的占位，对于gb2312得到的是汉字个数的2倍，utf8得到的是汉字个数的3倍

[mb\_strlen()](http://php.net/manual/zh/function.mb-strlen.php)就很好的解决了这个问题，它的第二个参数就是设置字符编码的

**2.15、用PHP能创建多级目录**

[mkdir](http://php.net/manual/zh/function.mkdir.php)($path, 0777, true);

**2.16、mysql\_num\_rows()和mysql\_affected\_rows()的作用和区别**

两者都是返回结果集中的行数，不同的是前者只对select操作有效，后者对update,insert,delete所影响的行数有效

**2.17、请列举、你能想到的所有的字符串查找算法，并加注释简单说明**

顺序查找，二分查找，分块查找，哈希表查找

**2.18、根据下面的代码写出结果**

$a = 2;$b = &$a;unset($a);

echo $b;

虽然使用了[unset](http://php.net/manual/zh/function.unset.php)，释放给定的变量，仅仅是断开了变量名和变量内容之间的绑定，并不意味着变量内容被销毁了。所以输出结果为“2”。

$a = 2;$b = 3;$c = &$a;$c = 2;

if(($a=5)>0 || ($b=$a)>0) {

$a++;$b++;

}

echo $a.'-'.$b.'-'.$c;

这里注意“||”这个符号，$a=5这里已经满足条件了，那么$b=$a就不用去执行了，那么$b还是3。

经过$a++和$b++后，两个变量都加1，变成6和4。而$c已经引用了$a，那么值也变成6。最后输出结果为“6-4-6”。参考《[PHP引用(&)使用详解](http://www.php-note.com/article/detail/245)》

**2.19、写个函数来解决多线程同时读写一个文件的问题**

首先用[fopen](http://php.net/manual/zh/function.fopen.php)打开某个文件，然后[flock](http://php.net/manual/zh/function.flock.php)加锁，再用[fwrite](http://php.net/manual/zh/function.fwrite.php)写内容，接着[flock](http://php.net/manual/zh/function.flock.php)释放锁，最后[fclose](http://php.net/manual/zh/function.fclose.php)关闭文档。[查看在线代码](http://ideone.com/KlbFIG" \t "_blank)。

**2.20、 设置session过期的方法，对应函数**

a. [setcookie()](http://php.net/manual/zh/function.setcookie.php)直接设置session\_id的生命周期。

b. 使用[session\_set\_cookie\_params()](http://php.net/manual/zh/function.session-set-cookie-params.php)，并且[session\_regenerate\_id(true)](http://php.net/manual/zh/function.session-regenerate-id.php)，当为true时则改变session\_id的值，并清空当前session数组。

[查看在线代码](http://ideone.com/R2l7FH)。

**2.21、php获取文件内容的方法，对应的函数**

a. [file\_get\_contents](http://php.net/manual/zh/function.file-get-contents.php)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

b. 用[fopen](http://php.net/manual/zh/function.fopen.php)打开url, 以get方式获取内容（借助fgets()函数）

c. 用[fsockopen](http://php.net/manual/zh/function.fsockopen.php)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

d. 使用[curl](http://php.net/manual/zh/book.curl.php)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

**2.22、php数组函数常见的哪些？**

a. 数组操作的基本函数（array\_values、array\_keys、array\_flip、array\_search、array\_reverse、in\_array、array\_key\_exists、array\_count\_values）

b. 数组的分段和填充（array\_slice、array\_splice、array\_chunk）

c. 数组与栈，列队（array\_push、array\_pop、array\_shift、array\_unshift）

d. 数组的排序（sort、asort、ksort）

e. 数组的计算（array\_sum、array\_merge、array\_diff、array\_intersect）

f. 其他的数组函数（array\_unique、shuffle）

**2.23、写一段上传文件的代码**

[POST方法上传](http://php.net/manual/zh/features.file-upload.post-method.php)，可以使用户上传文本和二进制文件。用 PHP 的认证和文件操作函数，可以完全控制允许哪些人上传以及文件上传后怎样处理。

通过[$\_FILES](http://php.net/manual/zh/reserved.variables.files.php)接收通过HTTP上传到服务器的文件，上传的内容存储在$\_FILES['xx']['tmp\_name']中，然后再通过[move\_uploaded\_file](http://php.net/manual/zh/function.move-uploaded-file.php)，将上传的文件移动到新位置。

在看些封装的上传类的时候，看到个函数[strrchr](http://php.net/manual/zh/function.strrchr.php)，查找指定字符在字符串中的最后一次出现，用这函数获取后缀。

**2.24、写出一个正则表达式，过虑网页上的所有JS/VBS脚本（即把标记及其内容都去掉）**

/<[^>].\*?>.\*?<\/>/si。[查看正则在线工具](http://tool.oschina.net/regex/)，有现成正则说明。

**2.25、用PHP打印出前一天的时间格式是2006-5-10 22:21:21**

echo [date](http://php.net/manual/zh/function.date.php)('Y-m-d H:i:s', strtotime('-1 day'));[查看更多Date/Time函数](http://php.net/manual/zh/ref.datetime.php)。

**2.26、foo()和@foo()之间有什么区别？**

foo()会执行这个函数，任何解释错误，语法错误，执行错误都会在页面上显示出来

[@](http://php.net/manual/zh/language.operators.errorcontrol.php)foo()在执行这个函数时，会隐藏上述的错误信息

**2.27、sort(),asort(),ksort(),arsort()，rsort()各有什么区别**

[sort](http://php.net/manual/zh/function.sort.php)：对数组进行排序，当函数结束时数组单元将被从最低到最高重新安排

[rsort](http://php.net/manual/zh/function.rsort.php)：对数组进行逆向排序

[asort](http://php.net/manual/zh/function.asort.php)：对数组进行排序并保持索引关系

[arsort](http://php.net/manual/zh/function.arsort.php)：对数组进行逆向排序并保持索引关系

[ksort](http://php.net/manual/zh/function.ksort.php)：对数组进行按照键名排序，保留键名到数据的关联，主要用于关联数组

**2.28、什么是可变变量？以下程序的输入值是什么？**

[可变变量](http://php.net/manual/zh/language.variables.variable.php)就是获取一个普通变量的值作为这个可变变量的变量名。输出值为“ok”。

$str = 'cd';

$$str = 'hotdog';

$$str = 'ok';

echo $cd;

**2.29、echo count("abc")将会输出什么？**

[count()](http://php.net/manual/zh/function.count.php)函数计算数组中的单元数目或者对象中的属性个数，通常是一个array()

对于对象，如果安装了 SPL，可以通过实现 [Countable](http://php.net/manual/zh/class.countable.php) 接口来调用 count()。该接口只有一个方法count(),此方法返回count()函数的返回值。

如果参数不是数组型或者实现了countable接口的对象，将返回 1，只有一个例外，如果参数是NULL，则结果是 0。

**2.30、GD库有什么作用？**

[GD库](http://php.net/manual/zh/ref.image.php)提供了一系列用来处理图片的API，使用GD库可以处理图片，或者生成图片。

在网站上GD库通常用来生成缩略图或者用来对图片加水印或者对网站数据生成报表。

**三、数据库**

**3.1、设一数据量为千万级的数据库，通过监控发现，有90%的查询请求都是指向同一部分数据的，而这部分数据只占整个数据库容量的1%。**

a.这种1%数据支撑了90%的查询工作，建义独立起来，这个方便数据在数据的buffer里，提高查询效率。

b.分库：后性能快速提升，如果效率还是上不去，可以考虑在那１%的数据上做一些nosql缓存。

c.分表：能够在数据库缺少集群解决方案的情况下获得一定的横向扩展能力，同时负载也能分担到多台物理设备，从这个角度上讲也能解决一些性能瓶颈。

d.如果有条件的话，此应用也可以通过缓存方案来解决，比如通过memcached,redis缓存1%结果集。

参考《[ChinaUnix问答的讨论](http://ask.chinaunix.net/question/42)》与《[德问的讨论](http://www.dewen.io/q/3071/%E8%A7%A3%E5%86%B390%25%E6%9F%A5%E8%AF%A21%25%E6%95%B0%E6%8D%AE%E7%9A%84%E5%8D%83%E4%B8%87%E7%BA%A7%E7%9A%84%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1)》

**3.2、MySQL如何分库分表？**

参考《[MySQL分库分表环境下全局ID生成方案](http://blog.sae.sina.com.cn/archives/1284)》，《[第一次操刀数据库分表的教训与经验](http://www.cnblogs.com/qqloving/p/3414999.html)》，《[mysql分库分表的一些技巧](http://my.oschina.net/cmcm/blog/175104)》

**3.3、MySQL数据库存储引擎有哪些？**

MyISAM、InnoDB、HEAP、BOB、ARCHIVE、CSV等

**3.4、 MyISAM与InnoDB的区别**

a. **存储结构：**MyISAM在磁盘上存储成三个文件。而InnoDB所有的表都保存在同一个数据文件中，一般为2GB

b. **存储空间：**MyISAM可被压缩，存储空间较小。InnoDB需要更多的内存和存储，它会在主内存中建立其专用的缓冲池用于高速缓冲数据和索引。

c. **事务支持：**MyISAM不提供事务支持。InnoDB提供事务支持事务，外部键等高级数据库功能。

d. **AUTO\_INCREMENT：** MyISAM可以和其他字段一起建立联合索引。InnoDB中必须包含只有该字段的索引。

e. **表锁差异：**MyISAM只支持表级锁。InnoDB支持事务和行级锁。

f. **全文索引：**MyISAM支持 FULLTEXT类型的全文索引。InnoDB不支持。

g. **表主键：**MyISAM允许没有任何索引和主键的表存在，索引都是保存行的地址。InnoDB如果没有设定主键或者非空唯一索引，就会自动生成一个6字节的主键(用户不可见)，数据是主索引的一部分，附加索引保存的是主索引的值。

h. **表的具体行数：**MyISAM保存有表的总行数。InnoDB没有保存表的总行数。但是在加了wehre条件后，myisam和innodb处理的方式都一样。

i. **外键：**MyISAM不支持。InnoDB支持

j. **CURD操作：**MyISAM如果执行大量的SELECT，MyISAM是更好的选择。如果你的数据执行大量的INSERT或UPDATE，出于性能方面的考虑，应该使用InnoDB表。

参考《[MySQL存储引擎中的MyISAM和InnoDB](http://www.biaodianfu.com/mysql-myisam-innodb.html)》

**3.5、MySQL数据类型有哪些？**

参考《[SQL 数据类型](http://www.w3school.com.cn/sql/sql_datatypes.asp)》

**3.6、MySQL数据库作发布系统的存储，一天五万条以上的增量，预计运维三年,怎么优化？**

a.设计良好的数据库结构，允许部分数据冗余，尽量避免join查询，提高效率。

b.选择合适的表字段数据类型和存储引擎，适当的添加索引。

c. mysql库主从读写分离。

d. 找规律分表，减少单表中的数据量提高查询速度。

e。添加缓存机制，比如memcached，apc等。

f. 不经常改动的页面，生成静态页面。

g. 书写高效率的SQL

**3.7、sql语句应该考虑哪些安全性？**

主要是防止注入，做好 ' " / \ 等特殊字符的转义，数据提交时注意html过滤，注意--,#这样的注释，注意子查询和一些mysql函数sleep,load\_file等等

**3.8、MYSQL取得当前时间的函数是?格式化日期的函数？**

now(); date\_format();[查看更多MySQL内置函数](http://www.51ebk.com/html/1267.html)。

**3.9、关系数据库的瓶颈**

**Memcached+MySQL**

随着访问量的上升，几乎大部分使用MySQL架构的网站在数据库上都开始出现了性能问题，程序员们开始大量的使用缓存技术来缓解数据库的压力，优化数据库的结构和索引。Memcached作为一个独立的分布式的缓存服务器，为多个web服务器提供了一个共享的高性能缓存服务，在Memcached服务器上，又发展了根据hash算法来进行多台Memcached缓存服务的扩展，然后又出现了一致性hash来解决增加或减少缓存服务器导致重新hash带来的大量缓存失效的弊端。

**MySQL主从读写分离**

Memcached只能缓解数据库的读取压力。读写集中在一个数据库上让数据库不堪重负，大部分网站开始使用主从复制技术来达到读写分离，以提高读写性能和读库的可扩展性。

**分表分库**

随着web2.0的继续高速发展，MySQL主库的写压力开始出现瓶颈，而数据量的持续猛增，由于MyISAM使用表锁，在高并发下会出现严重的锁问题，大量的高并发MySQL应用开始使用InnoDB引擎代替MyISAM。同时，开始流行使用分表分库来缓解写压力和数据增长的扩展问题。

**MySQL的扩展性瓶颈**

大数据量高并发环境下的MySQL应用开发越来越复杂，也越来越具有技术挑战性。分表分库的规则把握都是需要经验的。分库分表的子库到一定阶段又面临扩展问题。还有就是需求的变更，可能又需要一种新的分库方式。MySQL数据库也经常存储一些大文本字段，导致数据库表非常的大，在做数据库恢复的时候就导致非常的慢，不容易快速恢复数据库。大数据下IO压力大，表结构更改困难，正是当前使用MySQL的开发人员面临的问题。

**3.10、什么叫关系型数据库？**

关系型数据库是支持采用了关系模型的数据库，简单来说，关系模型就是指二维表模型。

关系型数据库以行和列的形式存储数据，以便于用户理解。这一系列的行和列被称为表，一组表组成了数据库。

**四、linux基础**

**4.1、linux下查看当前系统负载信息的一些方法**

参考《[linux 下查看系统资源和负载，以及性能监控](http://www.oschina.net/question/54100_45110)》

**4.2、vim的基本快捷键**

　　参考《[史上最全Vim快捷键键位图 -- 入门到进阶](http://cenalulu.github.io/linux/all-vim-cheatsheat/)》

**4.3、请解释下列10个shell命令的用途 top、ps、mv、find、df、cat、chmod、chgrp、grep、wc**

top 查看系统进程占用资源的情况。ps 显示详细的进程信息。

mv是为文件或目录改名，或将文件从一个目录中移植到另一个目录中。

find 查找文件或目录。

df 检查文件系统的磁盘空间占用情况。

cat 显示文件内容，创建文件一个新的文件，合并文件内容。

chmod 改变文件或目录的访问权限。

chgrp 改变文件或目录所属的组。

grep是一个强大的文本搜索工具。

wc 统计指定文件的数据信息，例如行数，字节数

**4.4、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。

**五、网络基础**

**5.1、写出下列服务的用途和默认端口ftp、ssh、http、telnet、https**

ftp文件传输协议是一种常见的文件拷贝方式，默认的是20用于数据连接，21用于控制连接端口。

ssh连接服务器执行操作，默认的是22端口号。

HTTP超文本传输协议，提供一种发布和接收HTML页面的方法，端口号为80

Telnet协议是TCP/IP协议族中的一员，是Internet远程登陆服务的标准协议和主要方式，默认端口为23

HTTPS是以安全为目标的HTTP通道，简单讲是HTTP的安全版，默认端口为443

**5.2、写出你能想到的所有HTTP返回状态值，并说明用途（比如：返回404表示找不到页面）**

200 OK 请求成功（其后是对GET和POST请求的应答文档）

301 Moved Permanently 所请求的页面已经转移至新的url

302 Found 所请求的页面已经临时转移至新的url

304 Not Modified，服务器告诉客户，原来缓冲的文档还可以继续使用

401 Unauthorized 被请求的页面需要用户名和密码

403 Forbidden 对被请求页面的访问被禁止

500 Internal Server Error，请求未完成。服务器遇到不可预知的情况

502 Bad Gateway，请求未完成。服务器从上游服务器收到一个无效的响应

503 Service Unavailable 请求未完成。服务器临时过载或当机

更多状态码参考《[HTTP 状态消息](http://www.w3school.com.cn/tags/html_ref_httpmessages.asp)》

**5.3、POST和GET有什么区别？**

a. GET是从服务器上获取数据，POST是向服务器传送数据

b. GET是通过发送HTTP协议通过URl参数传递进行接收，而POST是实体数据，通过表单提交

c. GET传送的数据量较小，不能大于2KB。POST传送的数据量较大，一般被默认为不受限制。但理论上，IIS4中最大量为80KB，IIS5中为100KB

d. GET安全性非常低，POST安全性较高

**5.4、请写出HTTP头，并符合以下要求：**

**１）这是一个post请求**

**２）目标：http://www.example.com:8080/test**

**３）POST变量：username: test  pwd: test2  intro: Hello world!**

**４）包含以下COOKIE信息：cur\_query: you&me**

POST  http://www.example.com:8080/test  HTTP/1.1

Cookie:cur\_query=you&me

username=test&pwd:=test2&intro=Hello world!

更多属性查看《[HTTP Header 详解](http://kb.cnblogs.com/page/92320/)》

**六、服务器基础**

**6.1、Apache与Nginx的优缺点比较**

**a. 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

**b. 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

**c. 原因：**

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。

参考《[Apache与Nginx的优缺点比较](http://www.itlipeng.cn/?p=617)》

**6.2、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。

**web内置模块**：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**：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。

参考《[fastcgi与cgi有什么不同](http://www.ttlsa.com/web/fastcgi-cgi-diff/)》

**6.3、Memcached和Redis区别**

a. 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。

b. 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。

c. 在100k以上的数据中，Memcached性能要高于Redis。

d. 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。

e. 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。

f. 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。

参考《[Redis和Memcached的区别](http://www.biaodianfu.com/redis-vs-memcached.html)》

**6.4、PHP 有哪些优缺点？**

**优点：**

a. 语法简单的，上手很快，而且还有很多很便捷的开发工具，比如说Zend Studio ,EclipsePHP Studio 等

b. 跨平台，而且还都是免费的 ，可以快速的在搭建LAMP(Linux Apache MYSQL, PHP), 支持很多主流的数据库系统，比如MYSQL， Oracle, PostgreSQL等等

c. 支持目前主流的技术，比如 WebService ,XML, AJAX 等等

d. PHP已经有很成熟的面向对象体系，可以支持面向对象的开发 （PHP5）

e. 有很多很好现有的框架，开源的论坛，以及博客等

f. 持续的更新和维护，以及社区的支撑，很多技术爱好者的共同努力，使PHP得到很广泛的应用，很多知名的网站也在使用PHP作为开发语言

**缺点：**

a. 对多线程支持不是很好，只能做一些简单的模拟线程

b. 语法不够严谨，如果以前做C++，Java 的就会很有感觉了，比如变量还没有定义，就可能直接使用

c.  也许有经验的PHP程序员最感到痛苦的地方是PHP的解释运行机制。这种运行机制使得每个PHP页面被解释执行后，所有的相关资源都会被回收。也就是说，PHP在语言级别上没有办法让某个对象常驻内存。在PHP中，所有的变量都是页面级的，无论是全局变量，还是类的静态成员，都会在页面执行完毕后被清空。以JSP为例，在JSP中，Java Bean的scope有四种有效值：Page、Application、Session、Request，分别对应页面、程序、会话、请求四种生存期。但在PHP中，只有Page一种生存期

参考资料：

<http://www.cnblogs.com/Excellent/archive/2011/12/15/2288733.html>   PHP常见面试问题

<http://www.cnblogs.com/xuechao/archive/2011/07/14/2106828.html>    应聘新浪PHP开发笔试答题总结

<http://www.cnblogs.com/bluespot/p/4223056.html>   高级php面试题

<http://www.jiancool.com/article/10391979797/;jsessionid=300759BA4B790D0C0640B2B37EE338BD>   试卷2

<http://www.cnblogs.com/jacko/p/4836731.html>   PHP面试出场率较高的题目

<http://www.cnblogs.com/tuozi001/p/3328216.html>   面试题目: PHP 有哪些优缺点？

<http://www.cnblogs.com/echohao/p/4722398.html>    php常见的面试题目

<http://www.cnblogs.com/coderchuanyu/p/3993820.html>    整理的一些PHP面试题目

<http://www.cnblogs.com/davidhhuan/archive/2010/04/21/1716960.html>   php面试题目

<http://www.cnblogs.com/511mr/articles/2269517.html>   经典php面试题目

<http://www.cnblogs.com/ccw5510/archive/2010/06/03/1751051.html>    腾讯PHP程序员面试题目

<https://www.ibm.com/developerworks/cn/opensource/os-php-designptrns/>    五种常见的 PHP 设计模式

<http://www.phpddt.com/php/registry-pattern.html>    PHP设计模式之注册模式