1.echo、print、print\_r、printf、var\_dump、var\_export区别

答：echo： echo是语言结构，无返回值，可以输出多个变量，变量之间用逗号分开，支持输出简单类型的变量（string 、int）。

print（）：是函数，有返回值，只能打印一个参数，也支持输出简单类型的变量（string 、int）。

print\_r():是函数，有返回值，可以打印复杂类型的变量（数组，对象），不仅可以打印变量的值，还能显示变量的类型。不能打印布尔型，如果true打印1，false和null打印为空。

printf():来源于C预言的printf，打印格式化的字符串。

var\_dump():不仅能打印（int，string，数组，对象，布尔值），而且还能输出变量的类型，长度、值。可以打印多个值，中间用逗号隔开。

var\_export()：在第二个值设置ture,不会打印变量，而是将其字符串形式返回，返回值可以赋值给PHP变量。

2.get和post区别

答：①get从服务器获取数据，post向服务器发送数据。

②get将表单字段和值附加到url上，可以看到，安全性非常低；post使用http post机制，将字段和值放置在HTML 的header中发送到url地址，用户看不到这个过程，安全性较高。

③get传送量较小，不能大于2kb;post传送量较大，默认不受限制。

3.cookie和session区别，禁用cookie，session是否仍可运行？

答：①存储位置：cookie存储在浏览器端，session存储在服务器端。

②安全性： session安全性比cookie要高，别人可以分析存储在本地的cookie，进行cookie欺骗，考虑到安全应使用session。

③单个cookie保存数据不能超过4KB，一个站点最多保存20个cookie ;

session会在服务器保存一段时间，当访问增多，占用服务器资源影响性能。

④将登陆等重要信息存放为session，其他信息如保存，可以放在cookie。

禁用cookie后：

①设置PHP.ini配置文件中“session.use\_trans\_sid=1”，或者在编译时打开“--enable-trans-sid”选项，让PHP自动跨页传递session ID;

②通过url传值、隐藏表单传递session ID；

③用文件或数据库形式保存session ID，在跨页过程中手动调用

4.PHP中PEAR是什么，PECL是什么

答：PEAR是PHP扩展与应用库（PHP Extension and Application Repository）。PEAR将PHP程序开发过程中常用的功能编写成类库，涵盖了页面呈现、数据库访问、数据结构、文件操作、缓存操作、网络协议等多方面，用户通过下载类库并做适当的订制以实现自己需要的功能，避免重复造“车轮”，PEAR的出现大大提高PHP程序的开发效率和开发质量。

pecl：可以看做是PEAR的一个组成部分，提供了与PEAR类似的功能，不同的是，PEAR是使用纯PHP代码实现的，用户在下载了类库后加以调试便能使用，而PECL是使用C语言编写的，补充了PHP难以实现的底层功能。往往需要用户重新编译或者在配置文件中设置才能使用在自己的代码中。

总结：PEAR是PHP的上层扩展，pecl是PHP的底层扩展，他们都是为特定的应用提供现成的函数或类。

5.PHP中常用的排序算法有哪些？

答：冒泡排序、快速排序、选择排序、插入排序

**function** maopao($arr){  
 **for** ($i = 1; $i < count($arr); $i++) {//需要冒泡的轮数  
 **for** ($j = 0; $j < count($arr) - $i; $j++) {//比较的次数  
 **if** ($arr[$j] > $arr[$j + 1]) {  
 $temp = $arr[$j + 1];  
 $arr[$j + 1] = $arr[$j];  
 $arr[$j] = $temp;  
 }  
 }  
 }

**return** $arr;  
}

//思路分析：在要排序的一组数中，选出最小的一个数与第一个位置的数交换。  
//然后在剩下的数当中再找最小的与第二个位置的数交换，  
//如此循环到倒数第二个数和最后一个数比较为止。

**function** selectFun($arr)//选择排序  
{  
 $len = count($arr);  
 **for** ($i = 0; $i < $len - 1; $i++) {//轮数  
 $min = $i; //假设最小值位置  
 **for** ($j = $i + 1; $j < $len; $j++) {//比较次数  
 **if** ($arr[$min] > $arr[$j]) {  
 $min = $j; //发现更小的，记录最小值的位置  
 }  
 }  
 **if** ($min != $i) { //最小值位置与假设最小值位置不同，交换位置  
 $tmp = $arr[$min];  
 $arr[$min] = $arr[$i];  
 $arr[$i] = $tmp;  
 }  
 }  
 **return** $arr;  
}

//思路分析：在要排序的一组数中，假设前面的数已经是排好顺序的，  
//现在要把第n个数插到前面的有序数中，  
//使得这n个数也是排好顺序的。如此反复循环，直到全部排好顺序。  
**function** insertFun($arr){  
 $len=count($arr);  
 **for** ($i=1;$i<$len;$i++){  
 $tmp=$arr[$i];  
 **for** ($j=$i-1;$j>=0;$j--){  
 **if**($tmp<$arr[$j]){  
 $arr[$j+1]=$arr[$j];  
 $arr[$j]=$tmp;  
 }**else**{  
 **break**;  
 }  
 }  
 }  
 **return** $arr;  
}

//思路分析：选择一个基准元素，通常选择第一个元素或者最后一个元素。  
//通过一趟扫描，将待排序列分成两部分，一部分比基准元素小，一部分大于等于基准元素。  
//此时基准元素在其排好序后的正确位置，然后再用同样的方法递归地排序划分的两部分。  
**function** quickFun($arr){  
 $len=count($arr);  
 **if**($len<=1){  
 **return** $arr;  
 }  
 $base\_num=$arr[0];  
 $left[]=**array**();  
 $right[]=**array**();  
 **for** ($i=1;$i<$len;$i++){  
 **if**($base\_num>$arr[$i]){  
 $left[]=$arr[$i];  
 }**else**{  
 $right[]=$arr[$i];  
 }  
 }  
 $left=quickFun($left);  
 $right=quickFun($right);  
**return** array\_merge($left,**array**($base\_num),$right);  
}

6.正则表达式验证邮件

$email='123456@qq.com';  
$patern='/\w+@(\w|\d)+\.\w{2,3}/i';  
**if**( preg\_match($patern,$email,$matches)){  
 **echo** 'success';  
}**else**{  
 **echo** 'no';  
}  
var\_dump($matches);

因为 preg\_match() 在第一次匹配后 将会停止搜索。preg\_match\_all() 不同于此，它会一直搜索subject 直到到达结尾。 如果发生错误preg\_match()返回 FALSE。

7.程序输出结果  
$str1=**null**;  
$str2=**false**;  
**echo** $str1==$str2?'相等':'不相等';//相等  
$str3='';  
$str4=0;  
**echo** $str3==$str4?'相等':'不相等';//相等  
$str5=0;  
$str6='0';  
**echo** $str5===$str6?'相等':'不相等';//不相等

$str7=0;

$str8=false;

echo $str7==$str8?'相等':'不相等';//相等

8 数据库中字段类型varchar 和char区别，那种字段查询效率高？

答：①varchar长度是可变的， char长度不可变

②存储：char比varchar类型的数据，存储与查找速度快。

③存储空间：因为char插入数据，类型长度固定，需要用空格进行占位，所以存储数据时占用更大空间，而varchar却不会。char以空间换时间效率，而varchar是以空间效率为首位的。

④存储方式：char对英文字符（Ascll）占用1个字符，1个汉字占用2个字符；varchar对英文字符和汉字都是占用2个字符。两者存储数据都是非unicodez字符数据。

9、MySQL数据库优化方法

答 ①选择合适的字段属性，尽量把字段设置为Not Null

②主键：每张表最好都有主键，主键类型最好都是数值类型

外键：为表合理的添加外键，外键类型最好也是数值类型

③优化查询语句，

用select 字段form表名，代替select \*from 表名

使用内连接（join）代替子查询，

使用联合Union代替手动创建临时表

④给经常充当where条件的字段添加索引，在建有索引的字段上尽量不要使用函数进行操作

⑤使用缓存

⑥拆分表：拆分字段，拆分记录

⑦主从数据库，负载均衡

10、Apache和NGINX区别和优劣

答：相同点：①都是http服务器软件

②功能上都采用模块化结构设计

③都支持通用的语言接口：如PHP、Python、perl

④支持正向、反向代理、虚拟主机、url重写、压缩传输、ssl加密传输

不同点：①Apache处理速度很慢，占用很多内存资源

②功能上：Apache支持所有模块动静态编译，NGINX都是静态编译

③对fcgi的支持：Apache支持的很不好，NGINX支持的很好

④处理链接方式：NGINX支持 epoll,Apache不支持

⑤空间使用上：NGINX安装包仅几百k

⑥nginx抗并发，NGINX处理请求是异步非阻塞的，而apache是阻塞型的，在高并发下NGINX能保持低资源低消耗高性能

⑦nginx作为负载均衡服务器，支持7层负载均衡

apache更成熟，稳定，比nginx的bug少

11.几种搭建负载均衡的方法，如何在不同的web服务器端保持用户的登录状态？

答：DNS轮询：是最简单的负载均衡，以域名作为访问入口，通过配置多条dns A记录使得请求可以分配到不同的服务器。 直接将服务器地址暴露给客户不利于服务器安全。

IP负载均衡：基于特定的Tcp/IP技术实现的。可以使用硬件设备或软件实现。硬件产品主要是F5，软件产品主要由LVS、HAproxy、NGINX,

CDN:( Content Delivery Network 内容分发网络 )，通过发布机制将内容同步到大量的缓存节点，并在dns服务器上进行扩展，找到用户最近的缓存节点作为服务提供点。

session:分布式或者集群时，session不能共享。将session存储到redis，减少服务器压力。先通过redis来判断用户状态也可以实现session共享。

cookie：登录验证后，创建登录凭证（用户id+登录时间+过期时间），将凭证加密后存储在cookie，以后每次请求都发送cookie，服务器根据对应的解密算法对其进行验证。

token: 将token添加到http header或者作为参数添加到url

12.MySQL数据库的引擎及区别

Innodb:支持事务，支持外键，支持行级锁，速度较慢，不支持存储过程，支持自动增长列，不支持全文索引，用于执行大量insert和update操作。

Myisam:不支持事务，不支持外键，支持表级锁，支持全文索引，速度较快，保存成文件形式，跨平台使用更加方便。适用于在应用中执行大量select操作。

memory：使用存在于内存中的内容来创建表。

13. 简述memcached、redis、mogodb

memcached:

redis:

mongodb: 文档型的非关系型数据库，优点在查询功能强大，能存储海量数据。

14.快速排序实现一维数组排序（第5题已给出）

15.遍历文件夹下的所有文件和子文件夹

$dir1='../874';  
**function** scanFile($dir){  
 $arr=scandir($dir);  
 **foreach** ($arr **as** $val){  
 **if**($val!='.'&&$val!='..'){  
 $path=$dir.'/'.$val;  
 **if**(is\_dir($path)){  
 scanFile($path);  
 }**else**{  
 **echo** "<p>$path</p>";  
 }  
 }  
 }  
}  
scanFile($dir1);

**function** my\_dir($dir) {  
 $files = **array**();  
 **if**(@$handle = opendir($dir)) { //注意这里要加一个@，不然会有warning错误提示：）  
 **while**(($file = readdir($handle)) !== **false**) {  
 **if**($file != ".." && $file != ".") { //排除根目录；  
 **if**(is\_dir($dir."/".$file)) { //如果是子文件夹，就进行递归  
 $files[$file] = my\_dir($dir."/".$file);  
 } **else** { //不然就将文件的名字存入数组；  
 $files[] = $file;  
 }  
  
 }  
 }  
 closedir($handle);  
 **return** $files;  
 }  
}  
**echo** "<pre>";  
print\_r(my\_dir($dir1));  
**echo** "</pre>";