## 一些细节总结：

Select count(\*) from user\_list; 查询出user\_list表中的记录条数

UTC时间与我们的时间相差8个小时

die($sql) 可以停止脚本执行，同时打印sql语句

十六进制下，数据前三位，是BOM信息

类是不可以重复载入的！！！\_\_autoload()按需载入

面向对象中的值传递和引用传递：

值传递：将对象的编号拷贝了一份，这两个变量的编号共同指向同一个对象

引用传递：两个变量共同指向同一个对象编号，该对象编号指向这一个对象

因此，这两种传值方式，只要修改其中的一个变量的值，另一个对象的值也会发生改变

Sql语句中日期的判断可以这样写：where dt < ‘2015-12-25 19:53:20’

面向对象中：

parent本身代表的是父类，

但是当在子类中调用父类的实例方法的时候，parent代表调用当前这个方法的这个对象

MVC:

基础模型类：

Db是上级类继承下来的实例属性，而parent是调用父类的静态属性，因此不能用parent调用。

$obj\_user在new的时候就可以使用$db这个属性了，因为它是父类BaseModel继承过来的属性，因此BaseModel中的这个$this实际上是指$obj\_user这个对象。

2月16日：

报错：BaseController未找到，如果直接将UserController这个类文件单独打开，这样被继承的基础控制器类没有被加载进来，当然会报错了！

在实现修改功能时，在对表单“兴趣项”做处理时，通过array\_sum对兴趣项的值进行求和相加，然后执行sql语句，将结果插入到数据库中，这样处理需要注意两个问题：

1. 表单中兴趣这一项的值必须是1，2，4，8 这种二进制的值；
2. 数据库字段设计时候必须是set类型的，这样通过array\_sum求得的结果才会被通过二进制来自动分配哪一项被选中了，因为set类型支持这种

当然，通常的做法是在控制器中将兴趣项的数据转换为字符串插入，在html页面的表单中通过in\_array()来判断兴趣的那一项是否在$\_POST提交的数组中，来判断选中状态。

2月23日：

在index.php页面头部加入空的html代码，在登录后台之后显示空白页，但未登录之前的页面都能正常显示？

造成空白的情况很多，程序错误，模板加载失败，本身就是空白页，或者只有空的html代码等等。以上情况是由于引入了frame框架导致的模版出现空白，在通常的项目开发中，类似index.php这样的php文件，头部是尽量不要出现空的html代码的，否则可能会导致验证码图片出不来，或者框架文件加载失败等异常！这种情形有时候也可能是被加载的文件使用了相对地址，可以修改为绝对路径地址尝试。

GD库扩展没有开启的问题：

在重装amp之后开启extension=php\_gd2.dll仍然无法显示GD库，那是因为extension\_dir出现了问题， "F:/php5.4.19/ext" 这个php扩展库（就是指向ext这个文件夹）的指向路径设置出现了问题，修改即好！

在验证码输入错误后，提示验证码错误，但是后续还会提示登录成功，并跳转至后台首页，这是为何？

在基础控制器中的跳转函数，是一定要在跳转之后写die()，停止脚本继续向下执行

24日又出现问题，在session存储好验证码之后 ,在跳转后判断时出现清空的现象，最终发现由于在BaseModel中使用的一个PDO的对象，那么外界在执行sql时必须要用PDO自己的方法才可以，或者将MYSQLDB类改造成一个PDO的类，然后在BaseModel中还是返回一个之前的$db对象，外界正常使用MYSQLDB这个类中的方法，exec只能执行增删改语句，查询语句不能使用它！！！

Day6：

1、常量使用"c2"和"c".2不都是字符串的意思吗？

正确的代码为：

const c2=12;

$a="c";

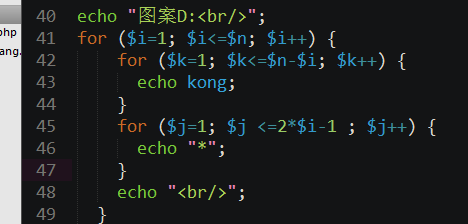
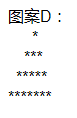
echo $a . 2;

echo constant($a . 2);

结果为：

在PHP中 . 是字符串连接，constant("c".2)是语法错误，因为字符串c没有被定义；

2、为什么输出的空格是斜的？



输出空格的那个循环应该写for($k=$n-$i;$k>=1;$k--)，空格要从多到少输出

3、define("DS", DIRECTORY\_SEPARATOR); 定义路径分隔符

4、在定义了一个变量之后，$GLOBALS里面为什么会有$key和$value?

因为GLOBALS存储了我们自定义的所有的全局变量和超全局变量

5、关于echo 输出整型的问题：

echo “你好$a”; 可以正常输出，

当然echo 你好.$a; 输出也是正常的，请在使用时加上引号。

echo 123不加引号的话，系统应该默认是个字符串，echo “123”也是作为字符串处理，

$a=1; echo $a; 作为整型数字变量来处理，

当然如果定义一个常量123，然后输出这个常量的话，也是作为数字一百二十三来处理的

在显示出来的结果上字符串123和数字一百二十三 是一样的。

Day7：

1、选择排序：

如何确定最后一项位置的下标？

当n等于0时候，即第一趟的最后一项位置就是n-1，下一趟的时候就少一个，就是减去$i所以最后一项位置的下标就是$n-$i-1

因为当i=0时，$j的下标为0~5共6个数，最后一个数的下标是5，因此也可以用$j-1表示

Day8：

1、有关字符编码的问题：

什么是Unicode。Unicode的基本目标是统一所有的编码，即它包含所有的字符集。这样只要一个系统支持Unicode，那么，就能处理这些字符集。一般Unicode有两个字节。现在的windows操作系统都是支持Unicode的。   
什么是UTF8呢？UTF8 == Unicode Transformation Format -- 8 bit   
是Unicode传送格式。即把Unicode文件转换成BYTE的传送流。UTF8是一种Unicode编码，即它的编码的字符集和Unicode是一致的。但编码的方式不一样。对于英文字符来讲，UTF8的编码和一般的一样，用到一个字节。但对于中文来讲，那么要用三个字节表示；

18、为什么表单会重复提交数据？

首先，页面第一次加载，在未进行任何操作，表单没有提交过的前提下，此时点击F5刷新，是没有任何问题的。  
F5页面刷新引起表单重复提交的前提条件：  
用户已通过  
 （1）submit按钮  
 （2）js的form.submit()方式提交过表单。  
根据提交表单的方式，查看提交表单后分两种情况，  
第一种：get：在地址栏，可看到页面地址栏变化：  
？后面是表单提交时传递的参数。毫无疑问，F5刷新时是会把这些参数重新提交的。  
第二种： post：chrome下打开开发者工具，可看到表单提交的数据。  
post方式提交的表单数据是存放在HTTP协议的消息体中以实体的方式传送到服务器的，  
当表单在提交时，浏览器的http请求头部会保存当前网页提交的表单数据，  
当表单再次提交时，如果数据未发生改变，则浏览器会默认将消息体中已经存在的表单数据提交给服务器。  
可见，F5页面刷新时，也会造成表单数据重新提交的。  
当表单以post方式提交重复数据时，默认Firefox浏览器会提示：  
“为了显示此页面，Firefox 必须发送将重复此前动作的数据（例如搜索或者下订单）”