1. 常用公共函数：
   1. redirect($url);跳转到指定的url。
   2. alert($msg);使用js弹出框，用法和js一致。
   3. js\_include\_tag();引用js文件。可以传入多个参数。Js文件必须放入javascript文件夹中。参数可以是不带后缀名的。例如：js\_include\_tag(‘index’,’admin.js’,’http://www.jwooo.com/js.js’);
   4. js\_include\_once\_tag();与js\_include\_tag类似，只是相同的文件仅引用一次。
   5. css\_include\_tag();引用css文件。与js\_include\_tag类似，css文件需要放入css目录。
   6. css\_include\_once\_tag();
   7. rand\_str($len=10);参数随机字符串，仅包含数字和大小写字母。参数为字符串长度，默认为10.
   8. is\_ajax();判断当前请求是否为ajax方式。返回为true或者false。
   9. get\_config($var);读取系统配置参数。参数为参数名称。参数配置在config/config.php中。
   10. debug\_info($msg,$type=’php’);输出调试信息，受debug\_tag开关控制。有两种输出方式：php和js方式。
   11. get\_current\_url();返回当前页面的url.
   12. use\_jquery();在当前页面使用jsquery.函数内部使用的js\_include\_once\_tag函数,因此在一个页面中,可多次调用,但尽量只调用一次.
   13. use\_jquery\_ui();使用jqueryui库.调用此函数后,自动加载jquery库.
   14. validate\_form($form\_id);对表单项进行js验证,表单项的class需要设定,具体设定参照第三节.
   15. show\_fckeditor($name=’editor’,$toolbarset=’Admin’,$expand\_toolbar=true);输出一个fckeditor.第一个参数为post的name,第二个参数为使用的工具栏名称,第三个参数为是否自动展开工具栏.
2. 数据库操作函数:(需要include frame.php文件)
   1. get\_db();获得数据库连接,数据库连接参数配置在config/config.php文件中.可配置备用数据库连接.返回值数据库连接操作实例或者false.
   2. close\_db();关闭数据库连接.
   3. 查询类数据库操作: $db = get\_db(); $db->query($sql);包含select show 等语句.(注:以下文档中,未说明情况下,$db均为通过get\_db();获得.)
   4. $db->query($sql)执行失败,则返回false.如果成功,将返回结果集,可直接通过字段名获取值.例如: $ret= $db->query(‘select \* from users’); echo “id=” . $ret[0]->id;
   5. 直接使用$db获取查询结果:
      1. $db->record\_count 查询结果行数.
      2. $db->field\_count.每行记录包含的字段数.
      3. $db->get\_field\_name($index) 返回第index个字段的名称.index从0开始.
      4. $db->move\_first(); 移动到查询结果第一行.如果找不到则返回为false.
      5. $var = $db->field\_by\_name(‘title’);获取当前行的title字段值并存入$var中.
      6. $var = $db->field\_by\_index(0); 获取当前行的第一个字段值并存入$var中.
      7. $db->move\_next();移动到查询结果下一行.如果找不到则返回为false.
      8. $db->move\_to($index);移动到指定的行数.
   6. 修改类数据库操作:
      1. $db->execute($sql);包含insert delete update等语句.返回true or false
      2. $db->affect\_count,返回上次执行影响的行数,类似query中的recordcount.
      3. $db->last\_insert\_id,返回上次insert的自动增加的id号.
   7. 数据库查询分页操作:
      1. $db->paginate($sql,$per\_page=10,$page\_var=’page’);用于获得查询数据库,并自动安装分页查询,返回为结果集.$per\_page用于指定每页包含几条内容.$page\_var为传递参数,一般使用默认值即可,除非在同一个页面中有多个地方需要使用到翻页页面时,需要使用不同的值.eg: $news = $db->paginate(‘select \* from news where cate\_id=1 order by created\_at desc’,5);
      2. paginate($url=’’,$page\_var=’page’,$ajax\_dom);翻页显示:和$db->paginate()配合使用.$url为需要连接到页面.$page\_var未传递参数,与$db->paginate()保持一致.$ajax\_dom:如果翻页需要使用ajax,则此参数用于指定返回的dom元素id.
3. form field验证:

在需要验证页面里,调用函数valide\_form($form\_name);在form里需要验证的field里,按照以下方式验证:

***required:***字段不能为空,eg:<input type=”text” name=”name” class=”required”>

***email:***字段为邮箱: eg:<input type=”text” name=”name” class=”required email”>

***url:***字段为url, eg:<input type=”text” name=”name” class=”required url”>

***date:***字段为日志类型,eg: eg:<input type=”text” name=”name” class=”required date”>

***number:***数字类型, eg:<input type=”text” name=”name” class=”required number”>

***creditcard:***信用卡号, eg:<input type=”text” name=”name” class=”required creditcard”>

***equalTo:***和另外一个input值相同, eg:<input type=”text” name=”password” class=”required” id=”password”>

<input type=”text” name=”password\_validta” class=”required” equalTo=”#password”>