* **layouts**

1. 当不使用指定的页面设置时，系统默认使用与该表对应的页面设置。

例：当不在form\_controller.rb中加layout “public\_layout”时，系统使用的是layouts/form.rhtml中指定的样式，即：<%= stylesheet\_link\_tag 'scaffold' %>中定义的样式。

1. 不套用任何layout

法一：

def ct\_image\_send\_img

render :layout => false

end

法二：

layout "doc\_layout", :except => [:download\_list, :ct\_image\_send\_img]

1. 指定页面设置

当用指定的页面设置时，不仅要在form\_controller.rb中加layout “public\_layout”，还要在public\_layout.rhtml中加上<%= stylesheet\_link\_tag 'scaffold' %>（假设sacffold为自定义的样式）。

1. 统一归纳layout的调用(vinea项目)

相同的layout归纳在一个文件中(user\_base\_controller.rb)，不同的直接写在对应的controller文件中即可覆盖公用的文件中设置的layout.如果对应的controller文件中没有设置layout就说明使用的是公用文件中设置的layout.

User\_base\_controller.rb设置公用layout的文件

|  |
| --- |
| **class** UserBaseController < ApplicationController  layout 'user'  **end** |

需要单独设置layout的文件purchase\_controller.rb

|  |
| --- |
| **class** PurchaseController < **UserBaseController**  layout **:select\_layout**  **def** select\_layout  **case** params[**:action**]  **when** 'confirm', 'edit\_address'  'buy\_list'  **when** 'list\_products', 'wmp\_confirm'  **false**  **else**  'user'  **end**  **end**  **end** |

1. 按进入页面的不同调用CSS文件

|  |
| --- |
| <% **if** params[**:controller**] == 'info' %>  <% **unless** params[**:action**] == 'settlement' %>  <link href="/stylesheets/other1.css" media="screen" rel="stylesheet" type="text/css" />  <% **else** %>  <link href="/stylesheets/news.css" media="screen" rel="stylesheet" type="text/css" />  <% **end** %>  <% **end** %> |

1. 在页面中插入CSS文件

|  |
| --- |
| <% content\_for **:header do** %>  <%= stylesheet\_link\_tag "user/live" %>  <% **end** %>  这样写会将样式文件插入到<head>中,如果直接在页面中写<%= stylesheet\_link\_tag "user/live" %>就会把样式文件插入到<body>中 |

1. 按登录用户的身份进行控制（wiki:Ruby/关于layout的动态引用）

|  |
| --- |
| **def** get\_layout  **if** session[**:user**] && session[**:user**].login != 'admin'  **return** 'blog'  **elsif** session[**:user**] && session[**:user**].login == 'admin'  **return** 'admin'  **else**  **return** 'anonymous'  **end**  **end** |

通过调用这个函数，实现动态引用（blog.rhtml，admin.rhtml,anonymous.rhtml分别为普通用户、管理员、匿名用户的**layout**）

1. 在用户界面和管理员界面显示不同的全局菜单

|  |
| --- |
| layout **:get\_layout**  **def** get\_public\_actions  **return** ["show","requested","question","reservation",  "confirm\_create","show\_content","show\_error","thanks"]  **end**  **def** get\_layout  public\_actions = get\_public\_actions  **if** public\_actions.include?(params[**:action**])  **return** 'public\_layout'  **else**  **return** 'admin\_layout'  **end**  **end** |

函数get\_public\_actions中的"show""requested""question"等页面是属于用户界面部分的。

1. 在user.controller中设置：

|  |
| --- |
| layout "public\_layout",**:only**[**‘**add\_user**’**,**’**edit\_user**’**,]  只在user下这几个页面显示  layout "public\_layout",**:except**[**‘**add\_user**’**,**’**edit\_user**’**,]  在user下这几个页面不显示 |

1. 不同的页面使用不同的布局

|  |
| --- |
| <%**if** params[**:controller**] == "cart" %>  <div id="top\_main\_body2">  <%**else**%>  <div id="main\_body">  <%**end**%>  <%= **yield** %>  </div> |

1. 使用cache

|  |
| --- |
| <%= stylesheet\_link\_tag 'global', 'header', 'footer', 'left', 'right', 'other', **:cache** => **false** %> |

* Table样式

1. 为某一列添加样式

|  |
| --- |
| <% **if** column.name=="id"%>  <th class="t\_first\_cell">  <%= human\_attribute\_name(Project, column.name) %>  </th>  <%**else**%>  <th>  <%= human\_attribute\_name(Project, column.name) %>  </th>  <%**end**%> |

1. 为某一行添加样式

|  |
| --- |
| <%**if** record\_id=="associated-record1"%>  <tr bgcolor=”red”>  <%= render **:partial**=>"ajax\_list\_record" %>  </tr>  <%**else**%>  <tr id="<%= record\_id %>" >  <%= render **:partial**=>"ajax\_list\_record" %>  </tr>  <%**end**%> |

1. 单双行样式

|  |
| --- |
| <% i=**0**  **for** add **in @adds**%> (此处要换行)  <% **if** i%**2**==**0**%>  <tr style="background-color**:**#ffffff">  <%**else**%>  <tr style="background-color**:**#f7f7f7">  <%**end**%>  <%i+=**1**%>  <td>asdfdsf</td>  <td>asdfasdf</td>  </tr>  <%**end**%> |

|  |
| --- |
| 长的写法：  <% **if** odd\_or\_even == **0** %>  <tr id="product\_<%= item\_product.id %>">  <% **else** %>  <tr id="product\_<%= item\_product.id %>" class="gray">  <% **end** %>  比较好的写法一：  <tr id="product\_<%= item\_product.id %>" <%= 'class="gray"' **unless** odd\_or\_even == **0** %>>  写法二：  <tr id="product\_<%= item\_product.id %>" class="<%= 'gray' unless odd\_or\_even == 0 %>"> |

* 菜单样式

1. 设置当前页的菜单样式：(riskmap)

在application\_helper.rb中定义两个函数

其中“on”为当前页面的样式；

|  |
| --- |
| **def** show\_admin\_menu\_link\_class(mark)  current\_page = "off"  **if** mark\_admin\_menu\_link\_class == mark  current\_page = "on"  **end**  **return** current\_page  **end** |

|  |
| --- |
| **def** mark\_admin\_menu\_link\_class  link\_ary1 = ['create\_template','risk\_new','success\_create','risk\_edit']  link\_ary2 = ['create\_user\_risk','success\_generate']  link\_ary3 = ['edit\_del\_template']  link\_ary4 = ['edit\_del\_risk','risk\_generate\_new','risk\_generate','risk\_generate\_edit']  link\_ary5 = ['sys\_user\_list','sys\_user\_edit','sys\_user\_editconfirm']  **case** params[**:controller**]  **when** 'admin/risk'  **if** link\_ary1.include?(params[**:action**])  **return 1**  **end**  **if** link\_ary2.include?(params[**:action**])  **return 2**  **end**  **if** link\_ary3.include?(params[**:action**])  **return 3**  **end**  **if** link\_ary4.include?(params[**:action**])  **return 4**  **end**  **when** 'admin/user'  **if** link\_ary6.include?(params[**:action**])  **return 5**  **end**  **end**  **end** |

在菜单中按如下方式调用样式

|  |
| --- |
| <%= link\_to localize(**:admin\_menu**, **:create\_template**),{**:controller**=>'/admin/risk',**:action**=>'create\_template'},{**:class**=>show\_admin\_menu\_link\_class(**1**)} %> |

1. 菜单JS条件判断

|  |
| --- |
| onmouseout="**this**.className = <%= (params[:subNavid] == "subNavi\_7" && params[:aid] == "edit\_7") ? "'navi\_7 navi\_7\_on'" : "'navi\_7'" %> "  ----------------------------------------------------------------  class='<%= (params[:subNavid] == "subNavi\_7" && params[:aid] == "edit\_7") ? "navi\_7 navi\_7\_on" : "navi\_7" %>' |

注意书写格式:onmouseout中双引号里的样式还要加单引号,而且class中双引号里的样式不再需要加单引号.

* 动态样式设置

1. **宽度**

|  |
| --- |
| <%**if** display\_state%>  <%**@width**= "66%"%>  <%**@div\_w**="33%"%>  <%**else**%>  <%**@width**= "100%"%>  <%**end**%> |

调用方法：

|  |
| --- |
| <div style="float**:**left**;**width**:**<%=@width%>;"></div> |

1. **<%%>中根据条件加样式**

<%%>中不要再用<%%>,要用”#”.

|  |
| --- |
| <%= link\_to "標準", {**:controller** => 'harika', **:action** => 'change\_fontsize', **:size** => 'default'}, {**:class**=>"#{(session[**:fontsize**]=='default' || session[**:fontsize**]==**nil**) ? 'fontsize\_default' **:** 'fontsize\_default\_off'}"} %> |

<%%>中可以加标签

1. **在link\_to中加标签**

|  |
| --- |
| <%= link\_to "本日来店予定", {**:presetname** => 'today\_visit'}, **:title** =>"本日来店予定" %>  <%= Karte.today\_visited\_count(session[**:shop\_id**]) %>  上面两个可以合写成下面的形式.  <%= link\_to "<label class='info\_title'>本日来店予定</label><label class='info\_con'>#{Karte.today\_visited\_count(session[**:shop\_id**])}</label>", {**:presetname** => 'today\_visit'}, **:title** =>"本日来店予定" %> |

1. **空参情况**

|  |
| --- |
| <%= text\_field\_tag 'login', "", **:style**=>"width:160px;" %> |

如果不先传一个空参，它就会把style作为value值传入。

* 数据库

1. 根据数据库中读取的数据设置样式

|  |
| --- |
| add为表名,view\_status为字段  <!--当读取的字符串为”DONE”时-->  <%**if** add.view\_status=="DONE" %>  <img src="/images/done.gif">  <!--当读取的字符串为”ACTIVE”时-->  <%**else** add.view\_status=="ACTIVE"%>  <img src="/images/active.gif">  <%**end**%> |

* ruby常识
* <%-………………-%>第一个减号表示不显示前面的换行，第二个减号表示不显示后面的换行。
* 在Rake Tasks右边的文本框中输入--trace可以显示详细的错误信息。
* 在application.rb中添加require 'pp'，表示输出。

1. 导入已存在项目

* 方法一

在project区单击右键，选择import，选择General/Existing Porjects into Workspace,选择select root directory后，选择要导入的项目，点击finish即可。(要有.project才可以导入)

此方法可以只导入src

* 方法二

file -> new -> project,选择Rails/rails project,点击next,在project name输入框中写上项目名，并让Options/Generate rails application skeleton不选中，然后点击finish即可。

此方法会连doc src一起导入

1. Rake Tasks

* db:create创建数据库
* db:migrate创建数据库中的表

1. helper中函数调用

form\_controller.rb中的函数都是实现一个页面的功能，将实现其它功能的小函数写在helpers/目录下对应该页面的文件中即form\_helper.rb.这些函数只能被views/form/目录下的页面使用，要想让所有的页面使用，需要将该函数写在helpers/application\_helper.rb中。

如果要调用helpers/目录下其它文件中的函数，可在顶行加：include“文件名”。

1. 必须登录才能进入网页

在Application.rb中的class ApplicationController < ActionController::Base函数中添加before\_filter :login\_required可实现必须登录才能进入页面的效果。

1. 文字路径

<%= Inquiry.human\_attribute\_name(:product\_name) %>

models>>Inquiry.rb------------------:product\_name => "お問い合わせのサービス名、商品名な"

1. 寻找没有明显标志的代码

<div id="target\_id"></div>与之对应的在页面源码中显示的是：这个DIV中包含一个table,，像这种情况寻找代码的方法是：在.rjs(ajax文件)和controller文件中寻找，要引用其它文件中的代码一定要有render，因此在check\_user.rjs文件中就有：

|  |
| --- |
| **if @user and @licence and @prohibition** text=render **:partial**=>'dis\_authority\_user',**:locals**=>{**:licence**=>**@licence**,**:prohibition**=>**@prohibition**}  page.replace\_html 'target\_id', text  **else**  page.replace\_html 'target\_id', ''  **end** |

从而可以找到<div id="target\_id"></div>中的引用的代码是来自dis\_authority\_user文件。

1. <%=\_ ( “文字信息” ) %>：

这样写gettext可以获取并翻译其中的文字。

* 快捷键

1. Ctrl+d删除整行；
2. Ctrl+w关闭当前文件
3. tab右缩进；tab+shift左缩进
4. ctrl+/注释，用在.rb文件中。
5. 当某个端口的项目不能启动时，可在任务管理器中结束进程ruby.exe,即可启动该项目;

* DOS环境：

1. 输入irb可进行Ruby编辑；
2. 输入ping ip可检查是否能链接到服务器；
3. 输入msconfig可打开“系统配置实用程序”对话框；

* 样式设置

1. **button背景图片替换**

将按钮图片作为背景图片，通过移动背景图片来实现鼠标放上，移出，点击时等不同的效果。用onmousedown会在鼠标按下时就替换图片，如果用onclick在鼠标按下时不会替换图片，只在鼠标放起时才会替换图片。

ftr-btn-editsave.gif

|  |
| --- |
| input.submit\_btn**{**  border**:**0px**;**  width**:**150px**;**  height**:**50px**;**  font-size**:**0px**;**  **}**  input#sb\_editsave**{**  background**:**url(/sys\_images/ftr-btn-editsave.gif) no-repeat center top**;**  **}** |

|  |
| --- |
| <%=submit\_tag localize(**:label**, **:confirm**),"class"=>"submit\_btn","id"=>"sb\_editsave","onmouseover"=>"MOver('editsave')","onmouseout"=>"MOut('editsave')","onmousedown"=>"MDown('editsave')" %> |

|  |
| --- |
| **function** MOver(id\_after){  **var** sb\_id="sb\_";  sb\_id+=id\_after;  **var** sb\_btn=document.getElementById(sb\_id);  sb\_btn.style.backgroundPosition="right top";  }  **function** MOut(id\_after){  **var** sb\_id="sb\_";  sb\_id+=id\_after;  **var** sb\_btn=document.getElementById(sb\_id);  sb\_btn.style.backgroundPosition="center top";  }  **function** MDown(id\_after){  **var** sb\_id="sb\_";  sb\_id+=id\_after;  **var** sb\_btn=document.getElementById(sb\_id);  sb\_btn.style.backgroundPosition="left top";  } |

1. **Img button**

为a设置ID，a:hover, a:active在IE和FF下都有效，如果直接为button设置hover和active，则只在FF下有效果，IE并不支持。

|  |
| --- |
| <%=link\_to(image\_tag("/sys\_images/btn\_change.gif"), {**:action** => "change\_plan\_type", **:id** => **@site**.id}, {**:id**=>"ftrChangeplan"}) %> |

|  |
| --- |
| <%= link\_to image\_tag('/images/common/fontsiza\_default.jpg', **:alt** => '標準', **:border** => '0', **:id** => '01'),{**:controller** => 'harika', **:action** => 'change\_fontsize', **:size** => 'default'} %> |

1. **Form**

|  |
| --- |
| <% form\_tag **:action** => 'update' **do** %>  <% **end** %>  <% form\_tag ({**:action** => 'update'},**:id**=>"id1") **do** %>  <% **end** %>  ―――――――――――――――――――――――――――――――――――  <% form\_remote\_tag **:url** => "/card/search/#{params[**:id**]}"  **do** -%>  <% form\_remote\_tag **:url** => "/card/search/#{params[**:id**]}", **:html**=>{**:class**=>"mt20"} **do** -%> |

1. **Img**

|  |
| --- |
| <img src="<%= url\_for(:controller => 'friend\_link', :action => 'show\_image', :id =>image, :size => 'flash') %>"> |

1. **Text**

* style1

|  |
| --- |
| <%=link\_to localize(**:label**,**:new**), **:action**=>"#{**@action\_prefix**}new", **:id**=>**@belongs\_to** %>  无法加载样式  <%= link\_to "選択画像削除",{**:id**=>"btn\_del\_select\_img"}%>  如果这样写，会将ID作为参数传递，应该像下面这种方式写：  <%= link\_to "選択画像削除",{},{**:id**=>"btn\_del\_select\_img"}%> |

<%= text\_field\_tag "completion\_keyword", '', **:class**=>'keyword' %>

* style2

|  |
| --- |
| <%=link\_to "logout",**:controller**=>"user",**:action**=>"logout"%>  -------------------------------------------------------  <%=link\_to "logout",{**:controller**=>"user",**:action**=>"logout"},{**:id**=>"g\_logout"}%> |

* <%= ajax\_link\_to\_new %>

ajax\_link\_to\_new 即“新规作成”是在associated\_helper.rb中定义的一个函数，可以通过修改此方法为此文字链接加样式。如：

<%= ajax\_link\_to\_new nil, "btn"%>

“btn”即所加载的样式，nil为此函数传递的另一个参数。

* style4

|  |
| --- |
| <%=link\_to\_remote as\_('新建'), **:url**=>{**:controller**=>'admin/daliy\_task',**:action**=>'ajax\_new' } %>  --------------------------------------------------------  <%=link\_to\_remote as\_('新建'),{ **:url**=>{**:controller**=>'admin/daliy\_task',**:action**=>'ajax\_new' }},**:id**=>"btn-new" %> |

1. **Submit button**

|  |
| --- |
| <%= submit\_tag localize(**:label**, **:confirm**) %>  -----------------------------------------------------------  <%= submit\_tag localize(**:label**, **:confirm**),**:id**=>"ftrEditsave" %> |

1. **button\_to**

button\_to方法是自定义的，可以通过修改此方法为其加样式，如：

|  |
| --- |
| <%= button\_to localize(**:label**, **:new**), **:action**=>"risk\_new",**:risk\_type**=>**@risk\_type** %>  -----------------------------------------------------------  <%= button\_to localize(**:label**, **:new**), "btn" , **:action**=>"risk\_new",**:risk\_type**=>**@risk\_type** %> |

“btn”即所加载的样式。

* 按钮写法转换

1. **将button转换成img\_button**

* 普通button

|  |
| --- |
| <%= button\_to "料金プランの変更", **:action** => "change\_plan\_type", **:id** => **@site**.id %>  ―――――――――――――――――――――――――――――――――  <%=link\_to(image\_tag("/images/btn\_change.gif"), **:action** => "change\_plan\_type", **:id** => **@site**.id) %> |

* Submit button

|  |
| --- |
| <%=submit\_tag localize(**:label**, **:confirm**) %>  ――――――――――――――――――――――――――――――――  <%=image\_submit\_tag("/images/edit\_save.gif",**:class**=>  "submit\_btn"*，*  **:style**=>*”*background-color**:**#cccccc”,:valign=>”top”) %> |

1. **将Link text 转换成button**

* link\_to

|  |
| --- |
| <%= link\_to \_("编辑"), **:action** => "#{**@action\_prefix**}risk\_edit",**:id**=>**@risk**.id%>  ―――――――――――――――――――――――――――――――――  <%= button\_to\_("编辑"),"btn",**:action**=> "#{**@action\_prefix**}risk\_edit",**:id**=>**@risk**.id%> |

其中”btn”即是定义的样式

button\_to 是在application\_helper.rb 中定义的方法:

|  |
| --- |
| **def** button\_to(name,cls,options = {}, html\_options = **nil**, \*parameters\_for\_method\_reference)  url=options.is\_a?(String)?options**:self**.url\_for(options,\*parameters\_for\_method\_reference)  res="<input type='button' value='#{name}' class='#{cls}' onClick=\"window.open('#{url}','\_self');\">"  **return** res  **end** |

* link\_to ocalize

|  |
| --- |
| <%= link\_to ocalize(**:label**, **:new**), **:action** => "risk\_new",**:risk\_type**=>**@risk\_type** %>  ―――――――――――――――――――――――――――――――――  <%= button\_to \_ (" 新規作成 "), "btn" , **:action** => "risk\_new",**:risk\_type**=>**@risk\_type** %> |

* 通过设计样式仿成button按钮

a{ display:block; }

1. **将input button转换成link text**

|  |
| --- |
| <input id="upload\_answer" onclick='save\_answer(**this**,"/risk/update\_answer","edit");' value="<%= \_("　アンケート提出")%>" type="button"/> |

* 标签

1. **Img**

|  |
| --- |
| <%=image\_tag( "/images/dave.gif", :class=>"beve", **:**size=>"80x120" ) %> |

* 常用方法

1. **put\_out\_div ( viena project )**

将此函数定义在application\_helper.rb中；主要用于循环函数，即当循环输出N个DIV后就输出一个<div class = “ clear\_space “ > </div>

|  |
| --- |
| **def** put\_out\_div(count,n)  str = ''  **if** count % n == **0**  str += <<-EOF  <div class="clear\_space"></div>  EOF  **end**  str  **end** |

|  |
| --- |
| <%**@content\_list**.each **do** |item|%>  <div class="search\_result\_box">  <h4>  <%=link\_to\_content(item)%>  </h4>  <%=link\_to\_content(item,image)%>  <p class="comment">  <%=item.comment\_short %>  </p>  <p class="tn\_copyright">  <%=item.copyright %>  </p>  </div>  <%**end**%> |

|  |
| --- |
| <%**@content\_list**.each\_with\_index **do** |item,index|%>  <%=put\_out\_div(index,3)%>  <div class="search\_result\_box">  <h4>  <%=link\_to\_content(item)%>  </h4>  <%=link\_to\_content(item,image)%>  <p class="comment">  <%=item.comment\_short %>  </p>  <p class="tn\_copyright">  <%=item.copyright %>  </p>  </div>  <%**end**%> |

* **奇怪问题**

1. **输入框与图片的间距**

auto\_complete插件中，输入框与其后面的元素之间在IE6/IE7中没有间距，原因不明；但在IE8/FF/Chrome/Safari中是有间距；

平常写法的输入框（即不用此插件的话）与其后面的元素无论在哪个浏览器中，都是有间距的；

|  |
| --- |
| <%= text\_field\_with\_auto\_complete **:work**, **:customer\_name**, **:class** => "w01" %>  <%= link\_to\_function "#{image\_tag('btn/select\_btn.gif', **:alt** => '選択')}", "openWithSBox('/customer/select\_customer?search\_work=', '500', '800');" %> |

应用了auto\_complete插件后的显示效果：

IE6/IE7中的效果:

2011-08-30_160603.png

IE8/FF/Chrome/Safari中的效果：

2011-08-30_160635.png