目录

[前端开发工具 3](#_Toc419817502)

[正则表达式工具：regexpal 4](#_Toc419817503)

[数据库开发工具使用NaviCat 5](#_Toc419817504)

[关于Apache虚拟主机的配置 6](#_Toc419817505)

[yii2学习笔记 6](#_Toc419817506)

[关于yii2框架新安装 6](#_Toc419817507)

[关于上传文件总结 6](#_Toc419817508)

[关于图像处理的函数 7](#_Toc419817509)

[数据模型Model在系统设计中的应用 7](#_Toc419817510)

[数据模型Model的属性验证 7](#_Toc419817511)

[核心验证器 8](#_Toc419817512)

[1、布尔型验证器'boolean' 8](#_Toc419817513)

[2、验证码'captcha' 8](#_Toc419817514)

[3、比较'compare' 8](#_Toc419817515)

[4、日期'date' 9](#_Toc419817516)

[5、默认值'default' 9](#_Toc419817517)

[6、双精度浮点型'double' 9](#_Toc419817518)

[7、电子邮件'email' 9](#_Toc419817519)

[8、存在性'exist' 9](#_Toc419817520)

[9、文件'file' 9](#_Toc419817521)

[10、滤镜'filter' 10](#_Toc419817522)

[11、图片'image' 10](#_Toc419817523)

[12、范围'in' 10](#_Toc419817524)

[13、整数'integer' 10](#_Toc419817525)

[14、数字'number' 11](#_Toc419817526)

[15、必填'required' 11](#_Toc419817527)

[16、安全'safe' 11](#_Toc419817528)

[17、字符串'string' 11](#_Toc419817529)

[18、正则表达式'match' 11](#_Toc419817530)

[19、裁剪'trim' 11](#_Toc419817531)

[20、唯一性'unique' 12](#_Toc419817532)

[21、网址'url' 12](#_Toc419817533)

[视图View 12](#_Toc419817534)

[在视图中渲染另一个视图 13](#_Toc419817535)

[布局文件嵌套 13](#_Toc419817536)

[关于视图文件View的嵌套总结 14](#_Toc419817537)

[Yii2代码注册CSS信息、Javascript代码、meta信息 14](#_Toc419817538)

[View中的Model若继承自数据库ActiveRecord 则可通过属性判断是否为新纪录isNewRecord 14](#_Toc419817539)

[view中使用的widget,第三方开发 15](#_Toc419817540)

[控制器Controller中可以使用redirect来重定向 15](#_Toc419817541)

[搜索的数据模型ModelSearch 15](#_Toc419817542)

[数据分页功能 15](#_Toc419817543)

[ActiveFied使用方法 16](#_Toc419817544)

[通过behavior编程自动更新时间戳 16](#_Toc419817545)

[浏览计数器更新 17](#_Toc419817546)

[使用composer命令安装yii2的extension 17](#_Toc419817547)

[Dynamic forms使用方法 17](#_Toc419817548)

[Bootstrap前端框架使用 17](#_Toc419817549)

[模态框在网页加载后自动执行（用于注册及登陆框的弹出） 17](#_Toc419817550)

[模态框背景禁止点击设置代码 17](#_Toc419817551)

[使用文字型按钮激活模态框 17](#_Toc419817552)

[Bootstrap在yii框架中的使用可以打开vendor目录下yii框架下的yii-bootstrap观看用法 18](#_Toc419817553)

[列大小表示方法col-lg-x（desktop） col-md-x（desktop）col-sm-x（tablet）col-xs-x（phones） 18](#_Toc419817554)

[Bootstrap示例代码网址 18](#_Toc419817555)

[登录示例代码 18](#_Toc419817556)

[CSS的应用 19](#_Toc419817557)

[增加自定义css样式文件 19](#_Toc419817558)

[Mysql数据库 19](#_Toc419817559)

[数据库表创建失败处理 19](#_Toc419817560)

[基于Yii2的web系统设计 20](#_Toc419817561)

[公用系统模块应该包括 20](#_Toc419817562)

[用户管理 20](#_Toc419817563)

[评论管理 可以实现对任意数据类型的评论及管理 20](#_Toc419817564)

[Tag标签系统 标签的生成 更新 检索 20](#_Toc419817565)

[文件的上传与下载 21](#_Toc419817566)

[log日志管理与系统统计分析 21](#_Toc419817567)

[系统时间设置 21](#_Toc419817568)

[系统中必须有缓存机制 21](#_Toc419817569)

[系统支持主题功能theme 21](#_Toc419817570)

[数据库增删改与查询操作 21](#_Toc419817571)

[使用BaseActiveRecord中的updateAllCounters页面浏览计数功能 22](#_Toc419817572)

# 前端开发工具

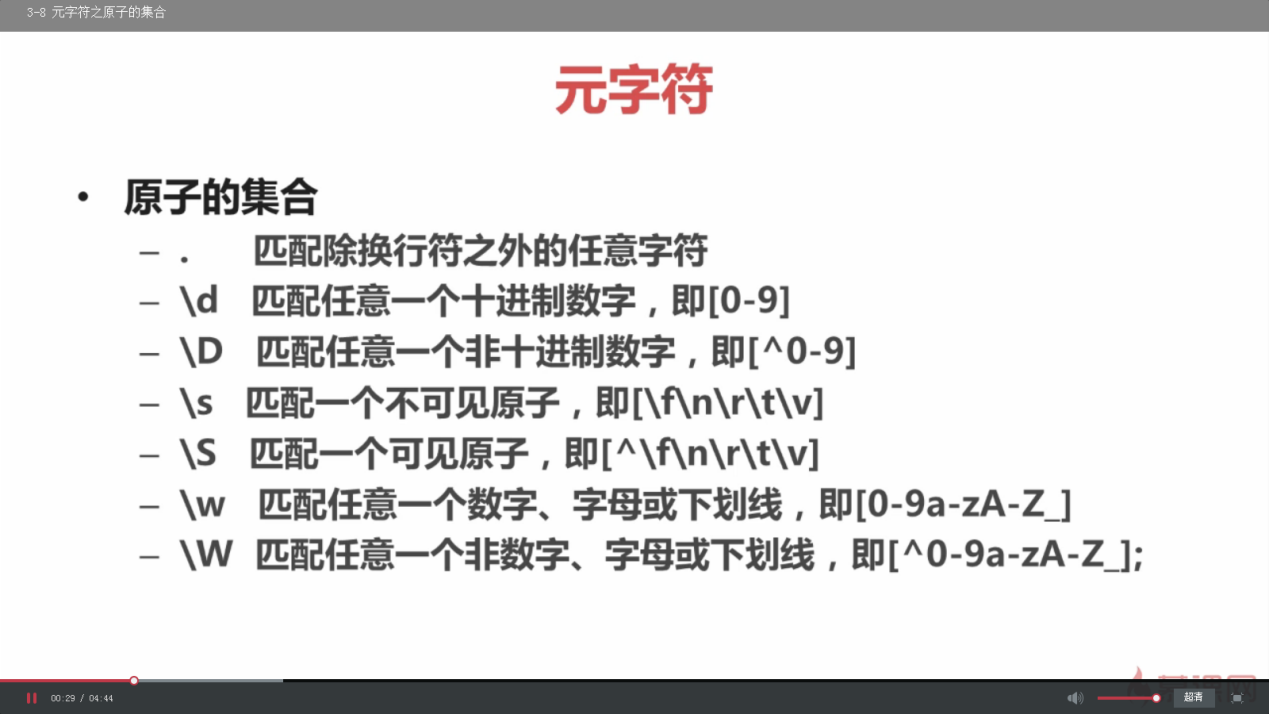


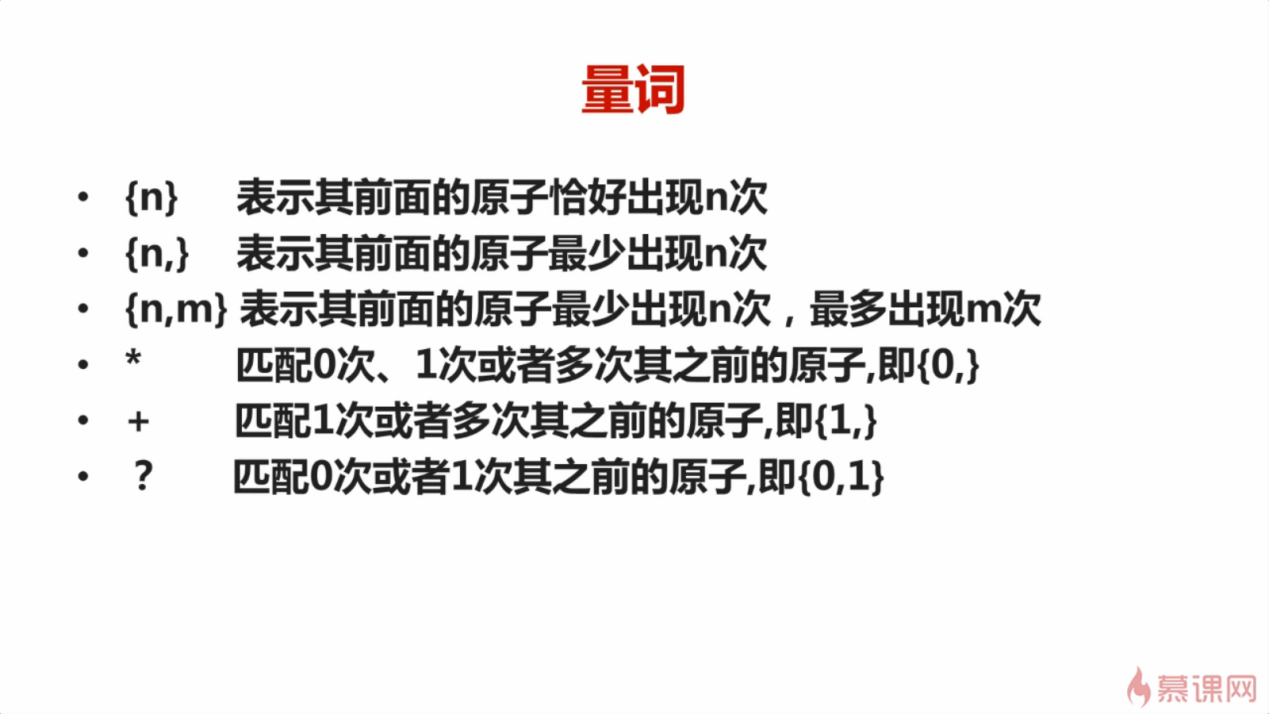
~~终端工具PowerCMD不好用，composer在其下不能运行，但是在cmd下运行正常~~

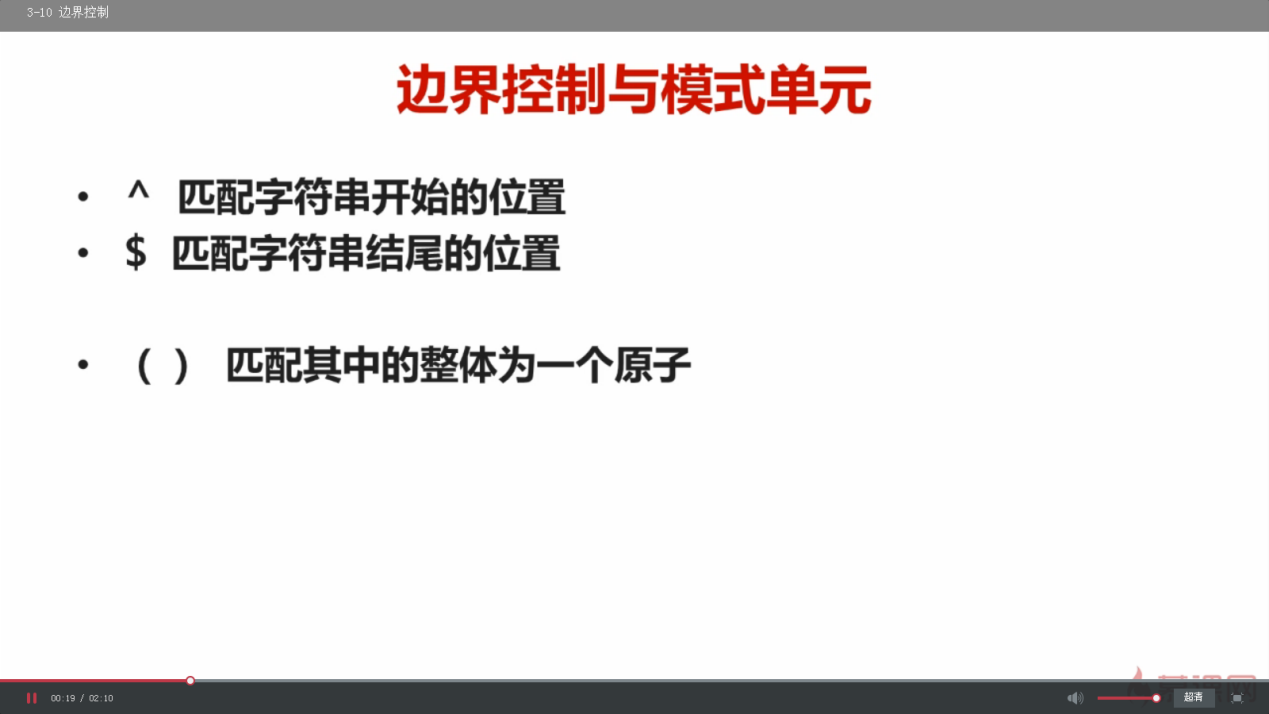
注册码：  
用户名：大眼仔~旭（Anan）  
注册码：PCMDA-P0S0R-TGHBE-E8KP8  
用户名：大眼仔~旭（Anan）  
注册码：PCMDA-7C2Y1-TH12E-9DATE  
用户名：大眼仔~旭（Anan）  
注册码：PCMDA-50NJM-3ACZE-2MS6Q

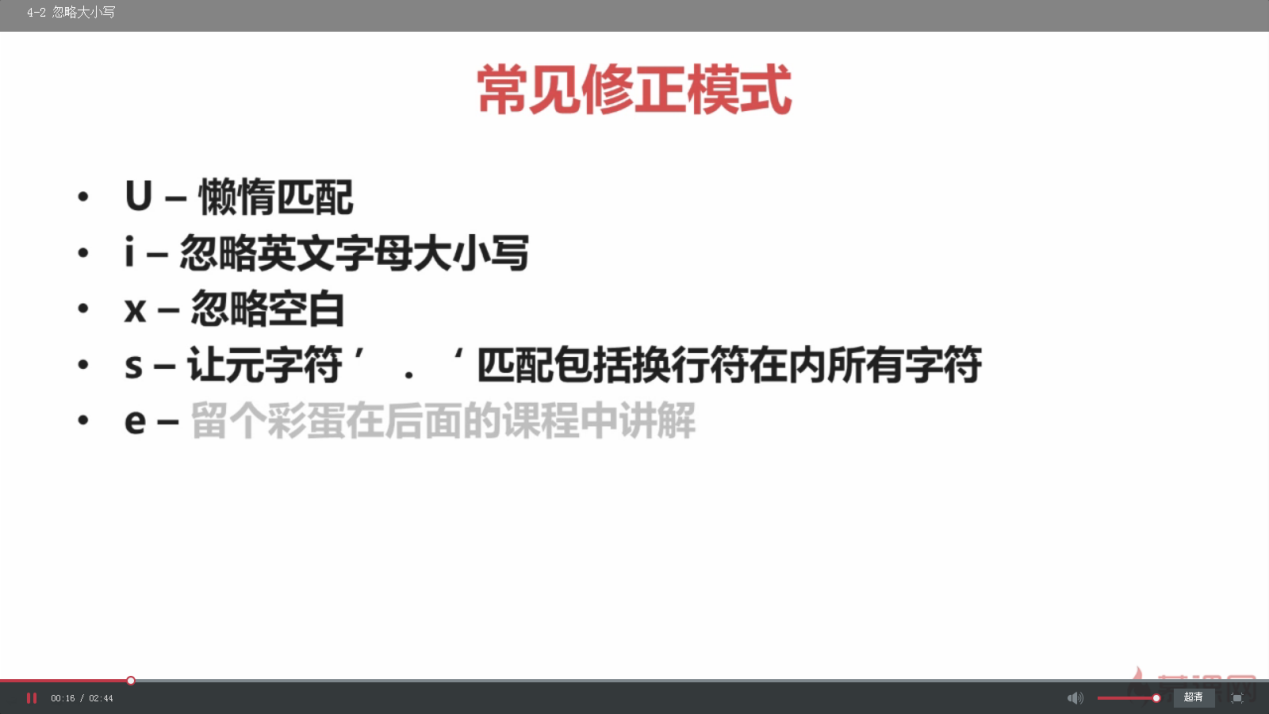
前端WEB编辑器：Sublime text 3、webstorm、Dreamwaver、MarkdownPad(编辑git的MD文档)

# 正则表达式工具：regexpal









# 数据库开发工具使用NaviCat

1. 创建外键时 必须保证字段的类型完全一致 int 和 unsigned int 是不同的
2. 数据库表名不分大小写 统一使用小写
3. unique属性在索引设置的tab页上 属于约束的一种

# 关于Apache虚拟主机的配置

1、打开主配置文件中httpd.conf 下关于虚拟主机的配置选项

2、配置extra 目录下的 httpd-vhosts.conf 这时主配置文件中的webroot设置失效 必须在httpd-vhosts.conf中重新全部配置 意思是httpd-vhosts.conf中必须包含所有的虚拟主机配置

# yii2学习笔记

## 关于yii2框架新安装

1. 参考资料 深入了解Yii2 www.digpage.com
2. 当出现网页不存在404 错误时 考虑 common\config\main.php 中的urlManager设置错误 niceurl url路径的表示方法设置
3. 新安装composer后务必安装 composer global require "fxp/composer-asset-plugin:1.0.0"否则运行composer update 或新增加扩展时会出错
4. 下载了新模板 要运行 yii.bat

## 关于上传文件总结

<http://www.yiichina.com/doc/guide/2.0/input-file-upload>

* controller和model中加上use yii\web\UploadedFile;
* 必须在view中增加<?php $form = ActiveForm::begin(['id' => 'form-signup','options' => ['enctype' => 'multipart/form-data']]); ?>否则不能上传
* uploads目录必须手工先建好，在web目录下
* 不支持中文名字的文件上传
* 验证规则

public function rules()

{

return [

[['file'], 'file', 'extensions' => 'jpg, png', 'mimeTypes' => 'image/jpeg, image/png',],

];

}

## 关于图像处理的函数

在shopImooc的ImageFunction.php中 有缩略图生成函数可以使用

## 数据模型Model在系统设计中的应用

1. Model类对应数据模型，数据模型可以是数据库，文件或其他数据
2. 在web应用中，若有与用户输入交互，则一般先生成一个FormModel继承自Model类，验证用户输入正确后，再生成一个ActiveRecord与数据库交互，最后返回数据类，而不是直接使用继承自数据库的类
3. 查询数据库某一数据表时，首先生成一个继承自该表ActiveRecord类的类，该类主要就是用于查询数据，通过该类得到ActiveQuery类，再通过ActiveDataProvider类使用该query获得数据

## 数据模型Model的属性验证

数据模型中的验证规则rules （）,数据验证函数validate（），出错后errors属性以及场景设置scenarios()

首先调用validate（），validate()根据rules（）中的规则进行验证，若不正确将会将错误信息保存到Model：：errors属性中

public function rules()

{

return [

// 在"register" 场景下 username, email 和 password 必须有值

[['username', 'email', 'password'], 'required', 'on' => 'register'],

// 在 "login" 场景下 username 和 password 必须有值

[['username', 'password'], 'required', 'on' => 'login'],

];

}

## 核心验证器

<http://www.yiichina.com/doc/guide/2.0/tutorial-core-validators>

### 1、布尔型验证器'boolean'

// 检查 "selected" 是否为 0 或 1，无视数据类型

['selected', 'boolean'],

### 2、验证码'captcha'

**view中增加代码**

use yii\captcha\Captcha;

<?= $form->field($model, 'verifyCode')->widget(Captcha::className(), [

'template' => '<div class="row"><div class="col-lg-3">{image}</div><div class="col-lg-6">{input}</div></div>',

]) ?>

**model中增加代码**

public $verifyCode;

['verificationCode', 'captcha'],

### 3、比较'compare'

// 检查 "password" 特性的值是否与 "password\_repeat" 的值相同

['password', 'compare'],

// 检查年龄是否大于等于 30

['age', 'compare', 'compareValue' => 30, 'operator' => '>='],

注意比较属性的输入顺序，\_repeat的属性放在前面让用户先输入，比较的属性放在后面

<?= $form->field($model, 'password\_repeat')->label('password')->passwordInput() ?>

<?= $form->field($model, 'password')->label('repeat\_password')->passwordInput() ?>

？？？

### 4、日期'date'

[['from', 'to'], 'date'],

### 5、默认值'default'

// 若 "country" 为空，则将其设为 "USA"

['country', 'default', 'value' => 'USA'],

### 6、双精度浮点型'double'

// 检查 "salary" 是否为浮点数

['salary', 'double'],

### 7、电子邮件'email'

// 检查 "email" 是否为有效的邮箱地址

['email', 'email'],

### 8、存在性'exist'

// a1 必需存在，但检验的是 a1 的值在字段 a2 中的存在性

['a1', 'exist', 'targetAttribute' => 'a2'],

### 9、文件'file'

// 检查 "primaryImage" 是否为 PNG, JPG 或 GIF 格式的上传图片。

// 文件大小必须小于 1MB

['primaryImage', 'file', 'extensions' => ['png', 'jpg', 'gif'], 'maxSize' => 1024\*1024\*1024],

### 10、滤镜'filter'

// trim 掉 "username" 和 "email" 输入

[['username', 'email'], 'filter', 'filter' => 'trim', 'skipOnArray' => true],

// 标准化 "phone" 输入

['phone', 'filter', 'filter' => function ($value) {

// 在此处标准化输入的电话号码

return $value;

}],

### 11、图片'image'

// 检查 "primaryImage" 是否为适当尺寸的有效图片

['primaryImage', 'image', 'extensions' => 'png, jpg',

'minWidth' => 100, 'maxWidth' => 1000,

'minHeight' => 100, 'maxHeight' => 1000,

],

### 12、范围'in'

// 检查 "level" 是否为 1、2 或 3 中的一个

['level', 'in', 'range' => [1, 2, 3]],

### 13、整数'integer'

// 检查 "age" 是否为整数

['age', 'integer'],

### 14、数字'number'

// 检查 "salary" 是否为数字

['salary', 'number'],

### 15、必填'required'

// 检查 "username" 与 "password" 是否为空

[['username', 'password'], 'required'],

### 16、安全'safe'

// 标记 "description" 为安全特性

['description', 'safe'],

### 17、字符串'string'

// 检查 "username" 是否为长度 4 到 24 之间的字符串

['username', 'string', 'length' => [4, 24]],

### 18、正则表达式'match'

// 检查 "username" 是否由字母开头，且只包含单词字符

['username', 'match', 'pattern' => '/^[a-z]\w\*$/i']

### 19、裁剪'trim'

// trim 掉 "username" 和 "email" 两侧的多余空格

[['username', 'email'], 'trim'],

### 20、唯一性'unique'

// a1 需要唯一，但检验的是 a1 的值在字段 a2 中的唯一性

['a1', 'unique', 'targetAttribute' => 'a2'],

### 21、网址'url'

// 检查 "website" 是否为有效的 URL。若没有 URI 方案，则给 "website" 特性加 "http://" 前缀

['website', 'url', 'defaultScheme' => 'http'],

## 视图View

<http://www.yiichina.com/doc/guide/2.0/structure-views>

 将预定义变量列到视图文件头部注释处，例如：error.php

/\* @var $this yii\web\View \*/

/\* @var $name string \*/

/\* @var $message string \*/

/\* @var $exception Exception \*/

PostCommentController对应的目录为 @app/views/post-comment注意横杠

布局

属于一种特殊的视图，公共视图，接受变量$content,$content代表render返回到布局的局部视图的渲染结果

在布局中可访问两个预定义变量：$this 和 $content

可以通过数据块实现内容视图和布局视图间的数据传递

首先，在内容视图中定一个或多个数据块：

...

<?php $this->beginBlock('block1'); ?>

...content of block1...

<?php $this->endBlock(); ?>

...

<?php $this->beginBlock('block3'); ?>

...content of block3...

<?php $this->endBlock(); ?>

然后，在布局视图中，数据块可用的话会渲染数据块，如果数据未定义则显示一些默认内容。

...

<?php if (isset($this->blocks['block1'])): ?>

<?= $this->blocks['block1'] ?>

<?php else: ?>

... default content for block1 ...

<?php endif; ?>

...

<?php if (isset($this->blocks['block2'])): ?>

<?= $this->blocks['block2'] ?>

<?php else: ?>

... default content for block2 ...

<?php endif; ?>

...

<?php if (isset($this->blocks['block3'])): ?>

<?= $this->blocks['block3'] ?>

<?php else: ?>

... default content for block3 ...

<?php endif; ?>

...

### 在视图中渲染另一个视图

<?= $this->render('\_overview') ?>

### 布局文件嵌套

在子布局文件中使用beginContent嵌套上层布局

<?php $this->beginContent('',[]); ?>

...

**<?php echo $content; ?>**

<?php $this-endContent(); ?>

### 关于视图文件View的嵌套总结

视图可以向上嵌套，即向上嵌套布局文件，使用beginContent函数

视图可以向下嵌套，即在视图中渲染另外一个视图，使用render函数

两者都可以传递参数给其他视图

### Yii2代码注册CSS信息、Javascript代码、meta信息

老汉视频呢十 01:01:52秒处

$this->registerMetaTag(...);

$this->registerLinkTag(...);

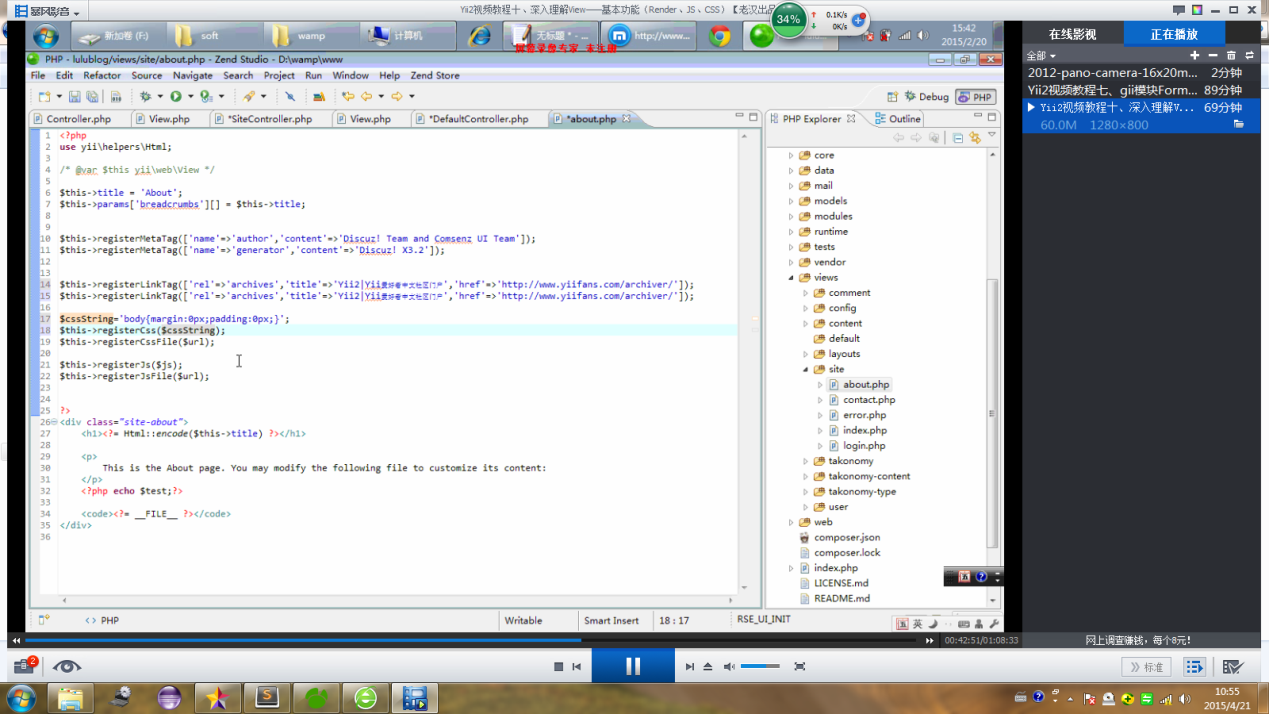
$this->registerCss(...);

$this->registerCssFile(...);

$this->registerJs(...);

$this->registerJsFile(...);

**当使用registerJsFile时js文件间有依赖关系使用'depends'参数设置，同时可以将该js文件放在页面的头部或尾部，'position'使用View：：POS\_HEAD**



|  |
| --- |
|  |
|  |  |

### View中的Model若继承自数据库ActiveRecord 则可通过属性判断是否为新纪录isNewRecord

<div class="form-group">

<?= Html::submitButton($model->isNewRecord ? 'Create' : 'Update', ['class' => $model->isNewRecord ? 'btn btn-success' : 'btn btn-primary']) ?>

</div>

### view中使用的widget,第三方开发

<http://Demos.krajee.com>

## 控制器Controller中可以使用redirect来重定向

带参数重定向到actionView

$this->redirect(['view', 'id' => $model->code]);

不带参数重定向到actionIndex

$this->redirect(['index']);

## 搜索的数据模型ModelSearch

1. 类名为数据模型名称+Search，例如：GoodsSearch表示搜索商品类
2. 继承自对应的ActiveRecord类 例如：GoodsSearch继承Goods类，Goods类继承自ActiveRecord类
3. 在GoodsSearch类中完成search($params)方法,一般由Controller中的actionSearch（）方法中使用，查询参数通过Yii::$app->request->queryParams获得
4. 在search方法中，首先通过ActiveRecord的find方法得到ActiveRecord使用的ActiveQuery对象
5. 将该ActiveQuery对象赋值给一个ActiveDataProvider对象
6. load（$params）???
7. 验证输入的查询条件，即字段是否符合rules（）中的规定
8. 执行相关字段过滤查询
9. 返回结果$dataProvider

在查询Controller中使用Search（）

一般在function actionIndex（）{

$searchModel = new GoodsSearch();

$dataProvider = $searchModel->search(Yii::$app->request->queryParams);

...

}

## 数据分页功能

<http://www.yiifans.com/forum.php?mod=viewthread&tid=69&fromuid=2>

Yii中的分页功能主要由yii\web: Linkable接口、yii\widgets: LinkPager类和yii\data: Pagination类三个组成。

* yii\data: Pagination 主要功能是对分页中的参数进行设置，如当前页、每页大小、总页数，总记录数等。
* yii\widgets: LinkPager 主要是根据yii\data: Pagination类所提供的参数生成前台页面的分页html代码。

**yii2.0 通过Activedataprovider 和 Pagination来实现分页功能**

<http://yiifans.com/forum.php?mod=viewthread&tid=3794&extra=>

## ActiveFied使用方法

<?= $form->field($model, 'keyWords',['inputOptions'=>['placeholder'=>'输入关键字','class' => 'form-control',],])->label(false) ?>

## 通过behavior编程自动更新时间戳

**use** yii\behaviors\TimestampBehavior;

**use** yii\db\Expression;

**public function** behaviors()

{

**return** [

[

'class' => TimestampBehavior::*className*(),

'createdAtAttribute' => 'created\_date',

'updatedAtAttribute' => 'updated\_date',

'value' => **new** Expression('NOW()'),

],

];

}

若使用的字段名为created\_at 和 updated\_at则可以简单使用下列代码，与上述代码功能相同

**public function** behaviors()

{

**return** [

TimestampBehavior::*className*(),

];

}

## 浏览计数器更新

$model= $this->findModel($id);

$model->updateCounters(['view\_count'=>1]);

## 使用composer命令安装yii2的extension

使用命令composer 代替php composer.phar ,在php的目录下已经生成了composer.bat的批处理文件

## Dynamic forms使用方法

在github上搜索yii2 dynamic form

## 单文件上传与多文件上传

<http://www.yiichina.com/tutorial/328>

# Bootstrap前端框架使用

## 模态框在网页加载后自动执行（用于注册及登陆框的弹出）

其中myModal这个是Modal的id

<script>

function popup()

{

$('#myModal').modal('show');

}

window.onload=popup;

</script>

## 模态框背景禁止点击设置代码

<div class="modal fade" id='myModal' data-backdrop="static">

### 使用文字型按钮激活模态框

<http://www.yiifans.com/forum.php?mod=viewthread&tid=11764&extra=>

<a data-toggle="modal" href="#mymodal" class=" btn btn-primary" >点击我弹出模态窗</a>

## Bootstrap在yii框架中的使用可以打开vendor目录下yii框架下的yii-bootstrap观看用法

## 列大小表示方法col-lg-x（desktop） col-md-x（desktop）col-sm-x（tablet）col-xs-x（phones）

<http://v3.bootcss.com/examples/grid/>

## Bootstrap示例代码网址

<http://v3.bootcss.com/getting-started/>

### 登录示例代码

<http://v3.bootcss.com/examples/signin/>

<div class="container">

<form class="form-signin">

<h2 class="form-signin-heading">Please sign in</h2>

<label for="inputEmail" class="sr-only">Email address</label>

<input type="email" id="inputEmail" class="form-control" placeholder="Email address" required autofocus>

<label for="inputPassword" class="sr-only">Password</label>

<input type="password" id="inputPassword" class="form-control" placeholder="Password" required>

<div class="checkbox">

<label>

<input type="checkbox" value="remember-me"> Remember me

</label>

</div>

<button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>

</form>

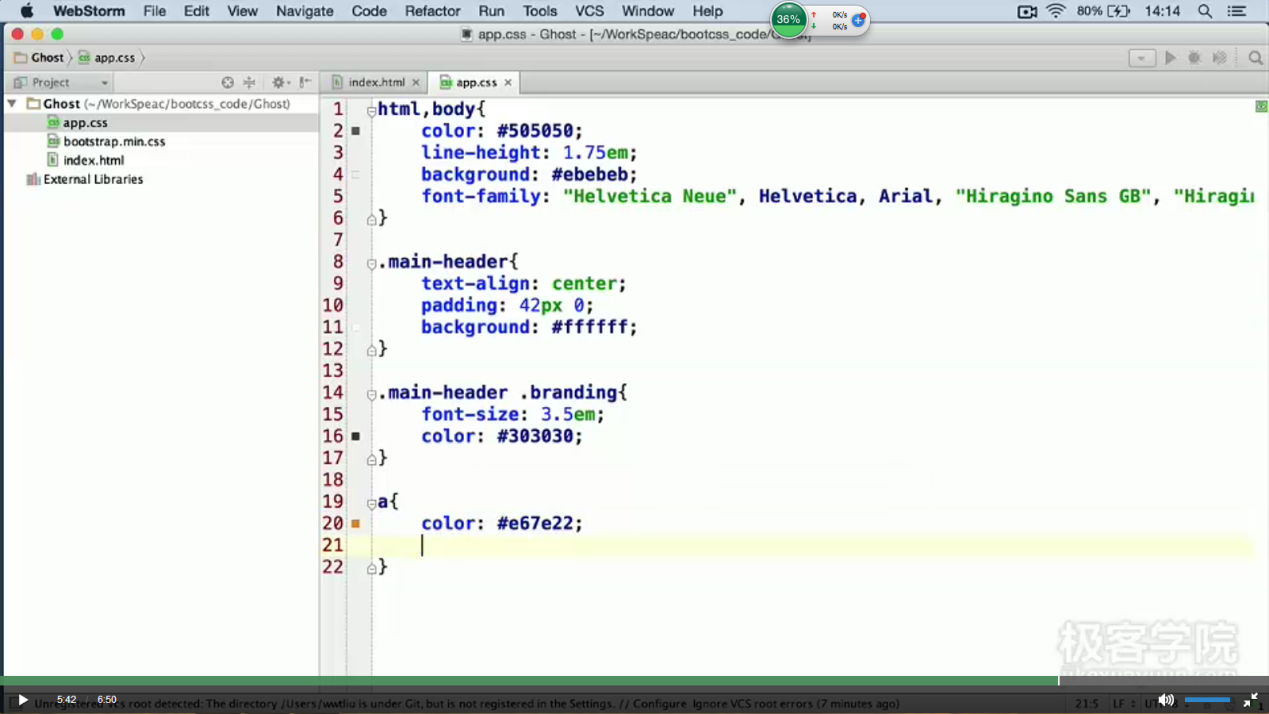
</div> <!-- /container -->

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# CSS的应用

## 增加自定义css样式文件

通过link标签引入自定义样式<link rel="stylesheet" href="" >



# Mysql数据库

## 数据库表创建失败处理

http://tieba.baidu.com/p/1261547370?pid=15055702256&cid=0#15055702256

停止数据库服务

删除mysql目录下data目录下数据库名字下的相关表文件一般两个\*.frm \*.ibd

重启数据库

若建表还存在问题，则

在mysql的bin目录下运行

**mysqlcheck -r g2gw -u root**

**mysqlcheck -o g2gw -u root**

# 基于Yii2的web系统设计

## 公用系统模块应该包括

## 用户管理

1. 登录
2. 注册
3. 修改
4. 忘记密码 找回密码可通过邮件 短信
5. 授权（rbac）

## 评论管理 可以实现对任意数据类型的评论及管理

实现对任意数据的评论，通过model\_type （string类型）来区别数据

无限回复，通过parent\_id实现

## Tag标签系统 标签的生成 更新 检索

1. 可以设计标签表tag

tagid //int 标签唯一标识

tag\_name //string 标签名称

count //int 标签被引用的次数

1. 设计标签映射表引用表tagmap

tagid //int 标签唯一标识

model\_id //int 数据唯一标识

model\_type //string数据类型

**数据的分类应自动成为一个标签，例如在电商系统中的商品分类，其分类包括所有的父类别都应成为该商品的标签**

**此处的设计与wordpress的不同，wordpress的标签设计是按照把标签分成文章类型（category）和普通标签（post\_tag）经行设计的，好处是可以看到文章在文章分类中的引用次数和在tag中的引用次数，同时不需要category表，即把分类表和标签表合并为一个表。坏处是不好理解，不符合自然思维。**

## 文件的上传与下载

## log日志管理与系统统计分析

## 系统时间设置

<?php echo $showtime=date("Y-m-d H:i:s");?>

时区设置有问题。两种解决方案：  
1.在php.ini中找到date.timezone，将它的值改成 Asia/Shanghai，即 date.timezone = Asia/Shanghai  
2.在程序开始时添加 date\_default\_timezone\_set('Asia/Shanghai')，这样手动设置为本地时区

找出该类型数据的所有评论（model\_type，model\_id）

对于每一个这样评论找出其子评论（parent\_id === comment\_id)

getSubCommentList($id)

subCommentList=find()->where(['parent\_id'=>$id]);

foreach()

getSubCommentList($subComment\_id)

## 系统中必须有缓存机制

## 系统支持主题功能theme

在配置文件中的components的view中指定theme，同时这里也可以指定view的实现类，即可以使用自己定义的view类，在自己定义的view类中的init中也可以指定使用的主题。

主题实质就是render函数中被渲染文件的路径映射，不同的主题放置在不同的路径下。

## 数据库增删改与查询操作

查询数据主要通过ActiveDataProvider类实现，数据的增删改主要通过ActiveRecord类来实现

## 使用BaseActiveRecord中的updateAllCounters页面浏览计数功能

Content::updateAllCounters(['view\_count'=>1],['id'=>$id]);

使用yii框架开发可以使用的模块包括

用户管理模块

用户授权模块rbac

缓存

文件上传

库表间的关系主要通过haveMany（）和haveOne（）建立，也可以通过中间表完成viaTable

<http://www.yiichina.com/doc/guide/2.0/db-active-record>

model和ActiveForm之间的变量对应，主要是通过form表单中的变量命名规则完成的，即MODEL\_NAME[ATRIBUTOR\_NAME],这样model就可以使用load（）函数将form表单中提交的数据正确赋值到model的对应属性中

ActiveForm的验证，主要通过前端和后端验证共同完成。后端验证主要流程是提交后，交由对应的数据model的验证规则验证，再将验证的提示信息，通过model重新显示给用户。

通过时间戳行为给数据库记录自动产生生成及修改记录

参考: <http://www.yiichina.com/doc/guide/2.0/concept-behaviors>

**namespace app\models\User;**

**use yii\db\ActiveRecord;**

**use yii\behaviors\TimestampBehavior;**

**class User extends ActiveRecord**

**{**

**// ...**

**public function behaviors()**

**{**

**return [**

**[**

**'class' => TimestampBehavior::className(),**

**'attributes' => [**

**ActiveRecord::EVENT\_BEFORE\_INSERT => ['created\_at', 'updated\_at'],**

**ActiveRecord::EVENT\_BEFORE\_UPDATE => ['updated\_at'],**

**],**

**],**

**];**

**}**

**}**

**若字段名不是缺省的created\_at和updated\_at则使用下面的代码**

**use** yii\behaviors\TimestampBehavior;

**use** yii\db\Expression;

/\*\*

\* @inheritdoc

\*/

**public function** behaviors()

{

**return** [

[

'class' => TimestampBehavior::*className*(),

'createdAtAttribute' => 'created\_date',

'updatedAtAttribute' => 'updated\_date',

'value' => **new** Expression('NOW()'),

],

];

}

在view中函数中使用双引号和单引号有区别

<?= $form->field($modelEcommerce, "[{$i}]website")->textInput(['maxlength' => 255]) ?>

动态form中必须使用双引号

CheckboxList使用方法为，给其付初值数组即是被选中选项

$checklist= array(‘中国’,’日本‘，’美国‘)

具体例子看backend里面的user管理

Dropdownlist联动示例，看backend中的category管理，主要用到控件的’onchange‘属性，然后产生post提交到服务器，由服务器返回数据，在通过返回的数据改变其他页面上的其他控件数据。

## 最佳实践

模型是代表业务数据、规则和逻辑的中心地方，通常在很多地方重用， 在一个设计良好的应用中，模型通常比[控制器](http://www.yiichina.com/doc/guide/2.0/structure-controllers)代码多。

归纳起来，模型

* 可包含属性来展示业务数据;
* 可包含验证规则确保数据有效和完整;
* 可包含方法实现业务逻辑;
* 不应直接访问请求，session和其他环境数据，这些数据应该由[控制器](http://www.yiichina.com/doc/guide/2.0/structure-controllers)传入到模型;
* 应避免嵌入HTML或其他展示代码，这些代码最好在 [视图](http://www.yiichina.com/doc/guide/2.0/structure-views)中处理;
* 单个模型中避免太多的 [场景](http://www.yiichina.com/doc/guide/2.0/structure-models#scenarios).

在开发大型复杂系统时应经常考虑最后一条建议， 在这些系统中，模型会很大并在很多地方使用，因此会包含需要规则集和业务逻辑， 最后维护这些模型代码成为一个噩梦，因为一个简单修改会影响好多地方， 为确保模型好维护，最好使用以下策略：

* 定义可被多个 [应用主体](http://www.yiichina.com/doc/guide/2.0/structure-applications) 或 [模块](http://www.yiichina.com/doc/guide/2.0/structure-modules) 共享的模型基类集合。 这些模型类应包含通用的最小规则集合和逻辑。
* 在每个使用模型的 [应用主体](http://www.yiichina.com/doc/guide/2.0/structure-applications) 或 [模块](http://www.yiichina.com/doc/guide/2.0/structure-modules)中， 通过继承对应的模型基类来定义具体的模型类，具体模型类包含应用主体或模块指定的规则和逻辑。

例如，在[高级应用模板](http://www.yiichina.com/doc/guide/2.0/tutorial-advanced-app)，你可以定义一个模型基类common\models\Post， 然后在前台应用中，定义并使用一个继承common\models\Post的具体模型类frontend\models\Post， 在后台应用中可以类似地定义backend\models\Post。 通过这种策略，你清楚frontend\models\Post只对应前台应用，如果你修改它，就无需担忧修改会影响后台应用。

**Controller的名称由两个单词组成时，访问的时候使用两个单词中间加横杠**

**例如：AuthItemController，访问时使用index.php?r=auth-item**

SearchModel传递$params参数时需要有formname，否则参数无法传递到searchModel中对应的变量中

/\*\*

\* Get all the comments of this user

\*/

**public function** getComments()

{

$searchModel=**new** CommentSearch();

$dataProvider = $searchModel->search(['CommentSearch'=>['userid'=>$this->id]]);

//var\_dump($this->id);

**return** $dataProvider;

}

Controller中action的访问控制

在继承的behaviors中增加access控制组件，例如：

'access' => [

'class' => AccessControl::className(),

'only' => ['logout', 'signup','contact'],

'rules' => [

[

'actions' => ['signup'],

'allow' => true,

'roles' => ['?'],

],

[

'actions' => ['contact'],

'allow' => true,

'roles' => ['@'],

],

[

'actions' => ['logout'],

'allow' => true,

'roles' => ['@'],

],

],

],

其中’？‘代表匿名用户即没有登录的客户，’@‘代表注册登录用户

在action中也可以使用can函数具体判断用户是否有该项权限

if(yii::$app->user->can('create-brand')){

//权限允许do something here

}

Else{

//权限没有do……

}

在视图中使用$dataProvider时，注意只能调用一次$dataProvider->getModels(),若调用两次则返回不同的$dataProvider，可能导致后续$dataProvider操作的错误

数据模型间的关系可以通过以下三种方式实现

hasMany()

hasOne()

当对应的关系的数据比较多时，可以使用search模型然后返回dataProvider实现数据的分页返回与排序。