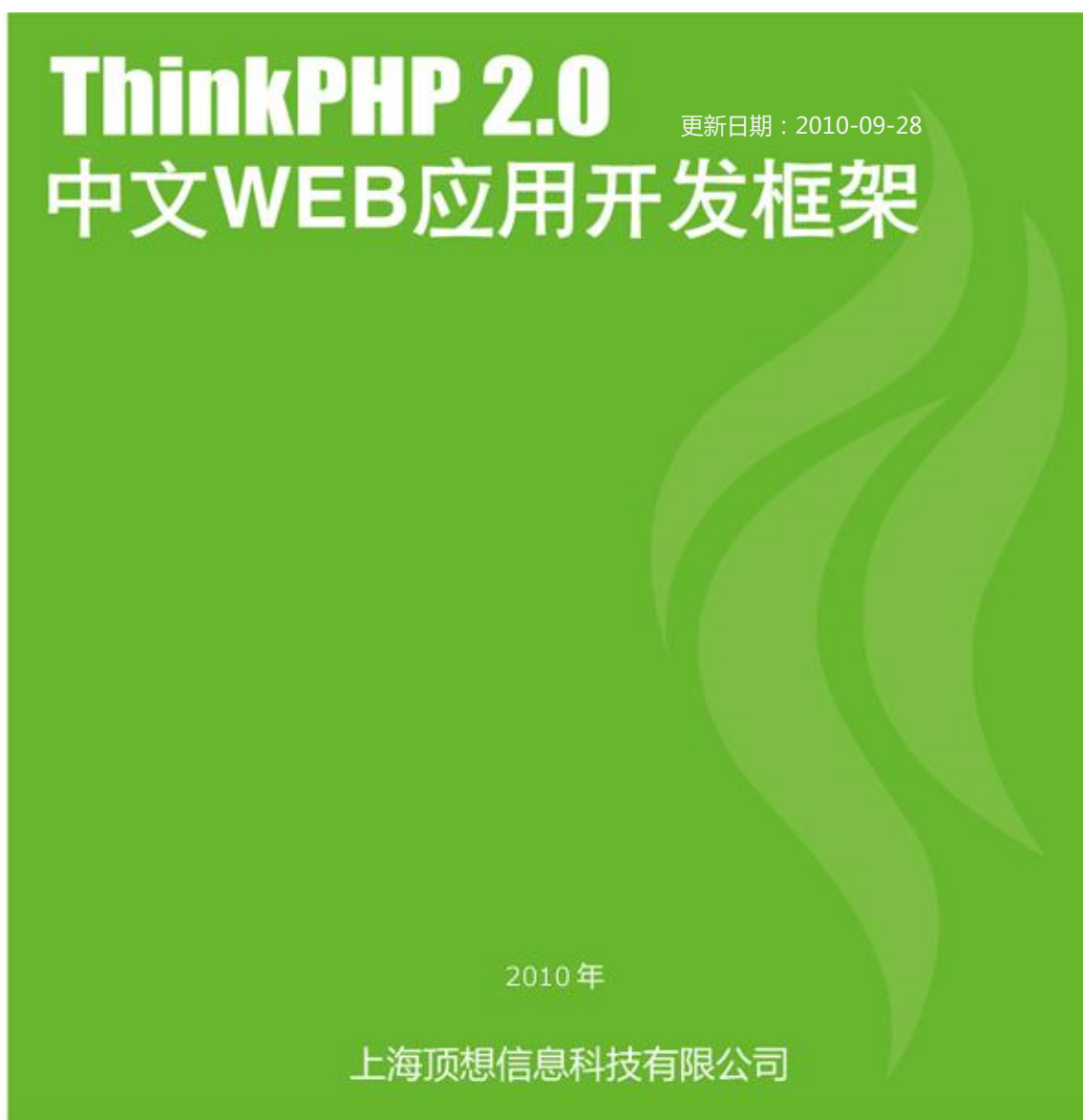




TP 自定义标签示例说明



1. 在项目 lib 文件夹中建立一个 TagLib 的目录并在其中新建一个标签解析类

例如我们增加一个对数据库查询的标签库，使能在模板中直接读取数据库，这样方便美工或者在模板中直接处理查询而不用经过 Action 赋值指定，我们暂且命名为：TagLibArticle.class.php 代码如下

(命名的规则须遵守前面控展名为 TagLibXXX 的样式，便于系统的调用)。

类库一般由二部分组成：

- 1) 定义标签库的参数，如传入参数，层次参数，或者闭合参数之类的。
- 2) 标签的处理，即根据传入的参数做过若干处理后最后输出给模板，需要注意的是：

I 标签命名须以 “_” 开头，且不能和别的标签名重复。

II 确保标签的正确性，标签都是必须符合 php 输出标准，否则可能造成比较严重的错误

如果无法显示或者输入的是空白或不是想要的内容，一般是此处解析错误造成。

示例：

```
import('TagLib');

class TagLibArticle extends TagLib{

    // 标签定义

    protected $tags = array(

        // 标签定义：

        //attr 属性列表 close 是否闭合（0 或者 1 默认 1） alias 标签别名 level 嵌套层次

        'article'=>array('attr'=>'name,field,limit,order,where,sql,key,mod','level'=>3),

    );

    //定义查询数据库标签

    public function _article($attr,$content) {

        $tag = $this->parseXmlAttr($attr,'article');

        $result = !empty($tag['result'])? $tag['result']:'article'; //定义数据查询的结果存放变量

        $key = !empty($tag['key'])? $tag['key']:'i';

        $mod = isset($tag['mod'])? $tag['mod']:'2';

        if ($tag['name'])
```

```

{ //根据用户输入的值拼接查询条件

    $sql = "M({'$tag['name']}')->";

    $sql .= ($tag['field'])?"field({'$tag['field']}')->":".";

    $sql .= ($tag['order'])?"order({'$tag['order']}')->":".";

    $sql .= ($tag['where'])?"where({'$tag['where']}')->":".";

    $sql .= "select()";

    }else{

        if (!$tag['sql']) return "; //排除没有指定参数

        $sql .= "M()->query({'$tag['sql']}')";

    }

    //下面拼接输出语句

    $parsestr = '<?php $_result='.$sql.'; if ($_result): $.key.=0;';

    $parsestr .= 'foreach($_result as $key=>$result):';

    $parsestr .= '++$.key.'; $mod = ($.key.' % $.mod.);?>';

    $parsestr .= $content; //解析在 article 标签中的内容

    $parsestr .= '<?php endforeach; endif;?>';

    return $parsestr;

}

}

```

2. 在项目的配置文件夹中加载标签，让系统启动时自动加载此配置标签

在项目 conf 下面添加 taglibs.php 的标签，加载自定义的标签。

格式按 ‘自定义标签名’ => ‘标签位置’ 指定

注：如果打开后，系统不解析标签，可以查看惯例配置中 TAGLIB_LOAD 是否设为 true(默认是开启的)，如果置为 false 是不会加载任何自定义标签的。

```
return array(

    'article'=> '@.TagLib.TagLibarticle' // 定义标签解析映射，TagLib 表示 lib 下的 TagLib 目录

);
```

3. 在模板中使用标签

要使用自定义标签库，需首先载入，载入有两种方式，一为自动载入，二为手动载入

- 1) 自动载入，需在系统的config.php配置中载入自定义标签：

```
'TAGLIB_BUILD_IN' => 'cx,article', //默认只载入 cx 标签
```

- 2) 手动加载，在需要自定义标签的html模板页面中指定：

```
<tagLib name="article" /> //默认系统会加载 cx 标签
```

然后就可以直接使用标签了：

调用方式为：<标签名：方法名 参数>内容</标签名：方法名>(如果定义的标签为非闭合标签则不需要后面的</标签名：方法名>)，请根据需要灵活指定。

```
<table>
<tr>

    <td class="tLeft" width="8%">序列</td>

    <td class="tLeft" width="12%">标题</td>

    <td>内容</td>

</tr>
    <article:article name="form" limit="10">
        <tr>
            <td>{$article.id}</td>
            <td>{$article.title}</td>
            <td>{$article.content}</td>
        </tr>
```

```
</article:article>  
</table>
```

这样就可以直接通过模板中的标签直接调用数据库显示内容了。

显示结果：

序列	标题	内容
1	ThinkPHP2.0发布	祖国60周年华诞献礼，ThinkPHP2.0新版正式发布了！
2	新版套装销售	ThinkPHP新版发布超值纪念套装火热征订中！还送2G主机空间！

相信通过上面的讲解，您也可以设计出更为方便、强大的自定义标签。

ThinkPHP 文档小组 2010-9-28

[HTTP://ThinkPHP.CN](http://ThinkPHP.CN) | WEB 应用开发最佳实践框架

大道至简，开发由我 WE CAN DO IT , JUST THINK