

02. 开发规范和目录结构

学习要点：

1. 开发规范
2. 目录结构

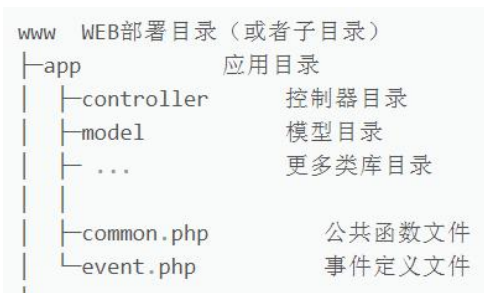
本节课我们来学习一下 ThinkPHP6.0 的编码开发规范以及框架的目录结构。

一. 开发规范

1. ThinkPHP6.0 遵循的是 PSR-2 的命名规范和 PSR-4 的自动加载；
2. 目录和文件的规范如下：
 - a. 目录名（小写+下划线）；
 - b. 类库和函数文件统一以 .php 为后缀；
 - c. 类的文件名均以命名空间定义，并且命名空间的路径和类库文件所在路径一致；
 - d. 类（包含接口和 Trait）文件采用驼峰式命名（首字母大写），其它采用小写+下划线命名；
 - e. 类名（包括接口和 Trait）和文件名保持一致，统一采用驼峰式命名（首字母大写）；
3. 函数和类、属性命名规范如下：
 - a. 类的命名采用驼峰法（首字母大写），如：User、UserType；
 - b. 函数的命名使用小写字母和下划线（小写字母开头）的方式，如：get_client_ip；
 - c. 方法的命名使用驼峰法（首字母小写），如：getUserName；
 - d. 属性的命名使用驼峰法（首字母小写），如：tableName；
 - e. 特例：以双下划线__打头的函数或方法作为魔术方法，如：__call 和 __autoload；
4. 常量与配置的规范如下：
 - a. 常量以大写字母和下划线命名，如：APP_PATH；
 - b. 配置参数以小写字母和下划线命名，如：url_convert；
 - c. 环境变量定义使用大写字母和下划线命名，如：APP_DEBUG；
5. 数据表和字段的规范如下：
 - a. 数据表和字段采用小写加下划线方式命名；
 - b. 并注意字段名不要以下划线开头，如：think_user 表和 user_name
 - c. 字段不建议使用驼峰和中文作为数据表及字段命名；

二. 目录结构

1. ThinkPHP6.0 支持多应用模式部署，app 是应用目录；
2. 默认情况下，是采用的单模式，如下图结构：



3. 多应用模式下，结构如下图：



4. 上图中，**app_name** 可以有多个，即多应用模式；

5. 在目录结构上，只确保对外可访问的仅 **public** 目录；



6. 在 **app** 目录中，还提供了一些文件，这些文件将在后面逐步讲到；

