# 专业学位研究生教学课程案例报告

# 案例名称：IT 技术博客网站

# 作者姓名：陈文斌

# 作者学号：2016160193

# 完成时间：2017 01 09

# 版权说明：本案例只用于教学目的，不对案例研究对象做出任何评判。

## 摘要：

该案例是一个使用WordPress搭建的一个IT技术博客网站，该网站主要提供IT技术相关的信息，通过博客的形式发布在网络中。

该网站提供注册功能，文章评论功能，站点文章搜索功能，打赏功能及广告位等功能。具备一般技术论坛的相关功能。

## 关键词：

博客网站，WordPress，中间件，插件。

# 一、案例正文

# 引言

网络上有很多的博客论坛可以使用，为什么非要使用wordpress，搭建一个独立的IT技术博客网站呢，有两个原因，一是，建立自己的品牌，需拥有自己独立的域名 ，独立的域名需要独立的网站，二是，拥有持续的广告来源，建立品牌的良性循环。

# 背景介绍

WordPress是一个免费的[开源项目](http://baike.baidu.com/view/188327.htm" \t "http://baike.baidu.com/_blank)，在[GNU通用公共许可证](http://baike.baidu.com/view/20301.htm" \t "http://baike.baidu.com/_blank)下授权发布。

WordPress 功能强大，[插件](http://baike.baidu.com/view/18979.htm" \t "http://baike.baidu.com/_blank)众多，易于扩充功能。安装使用都非常方便。

使用WordPress平台的发行商约占全球网站的10%。而WordPress官方网站的每月独立访问用户数则达到3亿。

从3.0开始，WordPress已经内置多用户[博客](http://baike.baidu.com/view/1509.htm" \t "http://baike.baidu.com/_blank)的功能，可以通过简单的设置就开设一个支持多域名的博客平台。

WordPress 是世界上使用最广泛的博客系统，是一款开源的PHP软件。因为使用者众多，所以WordPress社区非常活跃，有丰富的插件模板资源。使用WordPress可以快速搭建独立的博客网站。

WordPress 不仅仅是一个[博客程序](http://baike.baidu.com/view/1705040.htm" \t "http://baike.baidu.com/_blank)，也是一个优秀的小型[CMS](http://baike.baidu.com/view/15867.htm" \t "http://baike.baidu.com/_blank)，很多非博客网站也是用WordPress搭建的。

## 优缺点

**优点**

WordPress 功能强大、扩展性强，这主要得益于其插件众多，易于扩充功能，基本上一个完整网站该有的功能，通过其第三方插件都能实现所有功能；

wordpress搭建的博客对seo搜索引擎友好，收录也快，排名靠前；

适合DIY，如果你是喜欢丰富内容的网站，那么wordpress可以很好地符合你的胃口。

主题很多，网站上一大片都是wordpress的主题，各色各样，应有尽有！

wordpress备份和网站转移比较方便，原站点使用站内工具导出后，使用WordPress Importer插件就能方便地将内容导入新网站。

wordpress有强大的社区支持，有上千万的开发者贡献和审查wordpress，所以wordpress是安全并且活跃的。

**缺点**

wordpress源码系统初始内容基本只是一个框架，需要时间自己搭建；

插件虽多，但是不能安装太多插件，否则会拖累网站速度和降低用户体验；

服务器空间选择自由较小！

静态化较差，确切地说是真正静态化做得不好，如果要想对整个网站生成真正静态化页面，还做不好，最多只能生成首页和文章页静态页面，所以只能对整站实现伪静态化!

wordpress的博客程序定位，简单的数据库层等都注定了他不能适应大数据。

# 环境搭建

本地环境搭建，使用Wampserver + wordpress。

WampServer就是Windows Apache Mysql PHP集成安装环境，即在window下的apache、php和mysql的服务器软件。免去了开发人员将时间花费在繁琐的配置环境过程，从而腾出更多精力去做开发。在windows下将Apache+PHP+Mysql 集成环境，拥有简单的图形和菜单安装和配置环境。PHP扩展、Apache模块，开启/关闭鼠标点点就搞定，再也不用亲自去修改配置文件了，[WAMP](http://baike.baidu.com/view/1135708.htm" \t "http://baike.baidu.com/_blank)它会去做。再也不用到处询问php的安装问题了，wampserver一切都搞定了。这个软件是完全免费的，可以在其官方网站下载到最新的版本。

服务下载官网：<http://www.wampserver.com/>

WordPress官网：<https://wordpress.org/>

环境搭建步骤：

1. 安装wampserver。
2. 启动wampserver。
3. 创建mysql数据。
4. 将wordpress目录拷贝到wampserver的安装目录里的www目录里。
5. 在浏览器中输入http://localhost/wordpress/wp-admin即可进行环境的配置，输入相应的数据库，用户账户等配置信息。

# 功能实现

在其侧边栏上有相应的功能实现操作，如发表文章，添加多媒体文件，页面，评论，外观，插件，用户，工具，设置等。

首先需要选择一个主题，在外观里，有主题选项，可以先择一个自带的主题，也可以选择从网络上搜索一个主题，或自己上传一个主题。选择运用则当前网站使用的就是该主题。这里选择的是Impronta主题。

作为博客网站，文章是最实质的内容，在文章功能栏里，可以写文章，对应文章的编辑，默认的编辑功能比较的少，编辑出来的效果不是那么的好，可以下载相应的插件丰富文本的编辑功能，这里安装的是CKEditor for WordPress插件。

网站的侧边栏，是通过一系列的小工具实现的，只要把相应的小工具拖放过去，就可以实现相应的功能，非常的方便，可以通过下载插件的方式丰富侧边栏的功能。

这里处理常用的小工具外，还添加了，按日历搜索文章的功能，文章的浏览记录功能，站点品牌图片flash插件，网站的图标显示，网站加速插件，广告显示，打赏插件等。

一个有品牌影响力的网站，网站地址应该是有意义的，发表文章，很多时候使用的是中文名称，默认情况下网站地址会转化为无意义的地址，为了自动的能转换为有意义的地址，可以安装pinyin seo插件，将地址自动的转换为拼音格式。

# 代码管理

代码管理使用的是https://github.com/，可以把相关的代码提交到该网站上进行管理。

GitHub 是一个面向开源及私有软件项目的托管平台，因为只支持 Git 作为唯一的版本库格式进行托管，故名 GitHub。

作为开源代码库以及版本控制系统，Github拥有超过900万开发者用户。随着越来越多的应用程序转移到了云上，Github已经成为了管理软件开发以及发现已有代码的首选方法。

如前所述，作为一个分布式的版本控制系统，在Git中并不存在主库这样的概念，每一份复制出的库都可以独立使用，任何两个库之间的不一致之处都可以进行合并。

GitHub可以托管各种git库，并提供一个web界面，但与其它像 SourceForge或Google Code这样的服务不同，GitHub的独特卖点在于从另外一个项目进行分支的简易性。为一个项目贡献代码非常简单：首先点击项目站点的“fork”的按钮，然后将代码检出并将修改加入到刚才分出的代码库中，最后通过内建的“pull request”机制向项目负责人申请代码合并。已经有人将GitHub称为代码玩家的MySpace。

在GitHub进行分支就像在Myspace（或Facebook…）进行交友一样，在社会关系图的节点中不断的连线。

GitHub项目本身自然而然的也在GitHub上进行托管，只不过在一个私

有的，公共视图不可见的库中。开源项目可以免费托管，但私有库则并不如此。Chris Wanstrath，GitHub的开发者之一，肯定了通过付费的私有库来在财务上支持免费库的托管这一计划。

是的，我们正是这么计划的。通过与客户的接洽，开发FamSpam，甚至是开发GitHub本身，GitHub的私有库已经被证明了物有所值。任何希望节省时间并希望和团队其它成员一样远离页面频繁转换之苦的人士都会从GitHub中获得他们真正想要的价值。

在GitHub，用户可以十分轻易地找到海量的开源代码。

# 上线运行

本来是打算将网站部署在openshift上，但因为国外的网站被国内种种限制，总是出现打不开的现象，所以就没法部署在这上面了。

国内的服务器有新浪的云虚拟主机，地址为http://www.sinacloud.com/

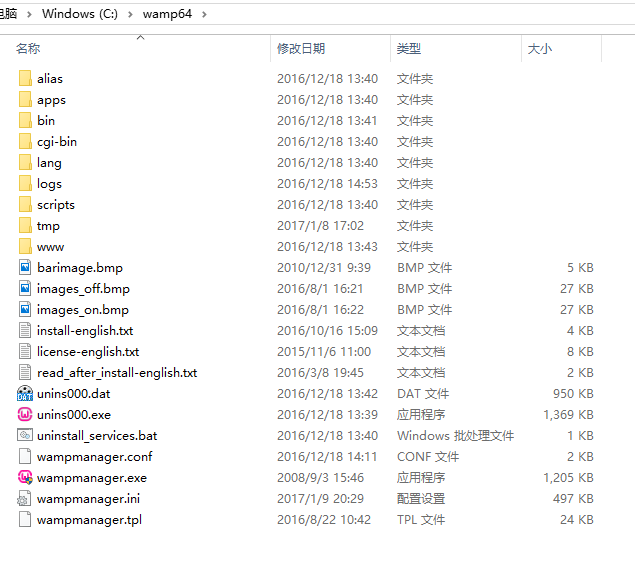
可以在上面申请空间和服务，把本地做好的代码提交上去即可以。

不过由于云服务读写权限的问题，需要调整数据库，上传资源等一些内容。

# 源码分析

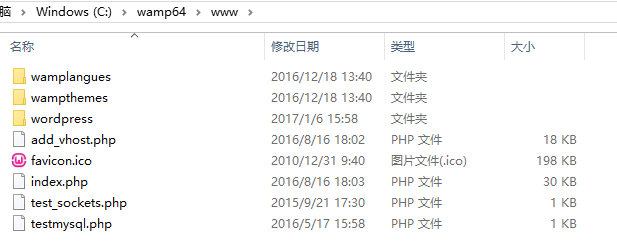
## wampserver目录结构

wampserver安装后的目录结构如下图：

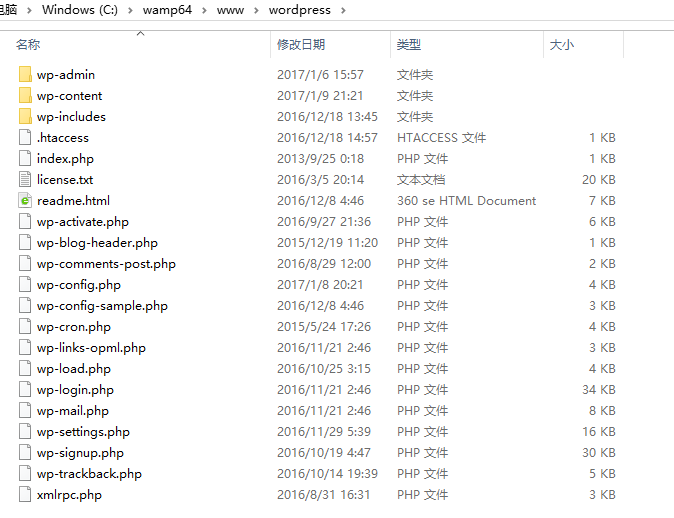


## wordpress目录结构

wordpress目录，是在www目录下，具体的目录结构如下图：



wordpress内部目录结构，如下图：



wp-admin 文件夹里是管理员的相关信息。

wp-content是实际的内容，如主题，插件，上传的图片，flash等资源。

wp-includes是框架系统的相关资源。

Wordpress是一个单入口的文件，所有的前端处理都必须经过index.php,这是通过修改web服务器的rewrite规则来实现的。这种做法的好处是显而易见的，这样URL更好看，不必为每一个url新建一个文件。

 wordpress可以分为3个阶段：

一是初始化阶段，即初始化常量、环境、加载核心文件等等；

二是内容处理阶段，即根据用户的请求调用相关函数获取和处理数据，为前端展示准备数据；

三是主题应用阶段，在这个阶段，需要展示的数据已经准备完毕，需要根据用户的请求加载相应的主题模板，即对主题进行路由。

经过这三各阶段，用户请求的页面就可以完全的展现出来了。

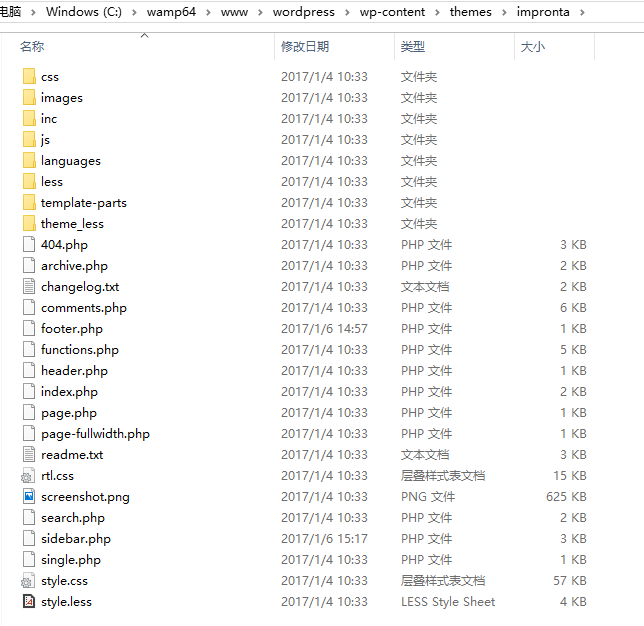
Index.php文件实际上没啥内容. 定义了常量WP\_USE\_THEMES和加载了文件wp-blog-header.php

wp-blog-header任然很简单，但是可以看到wp的三个明显的阶段，初始化阶段、内容处理阶段、主题应用阶段。

wp-load.php的功能很简单，就是加载wp-config.php. wp-config.php是需要用户自己进行设置的。设置DB设置，密钥，路径，语言等。

## 主题目录结构

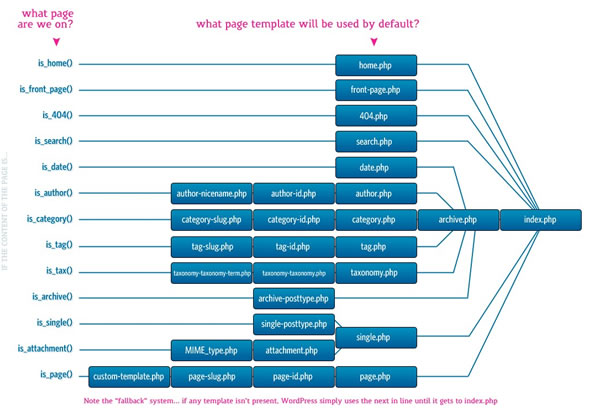
主题目录结构如下图：



当访问一个页面时，我们要看看有哪些模板文件被执行。我们都知道单篇日志是用single.[PHP](http://lib.csdn.net/base/php" \o "PHP知识库" \t "http://blog.csdn.net/zhang_red/article/details/_blank)解析内容的。 单页面是用page.php来解析的。但WordPress将根据各种实际情况来寻找不同的模板文件。

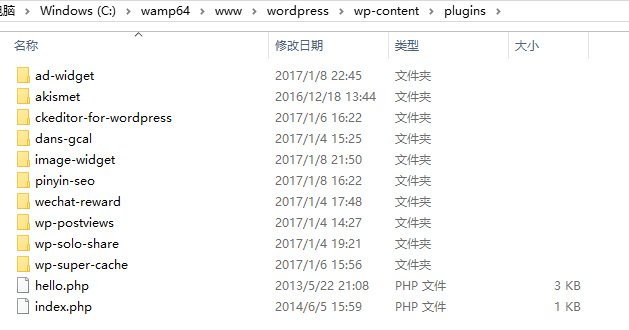
我们首先应该清楚这一点:没有 index.php和 style.css你的主题已经不再是一个有效的WP主题…所以理所当然,如果你只有这两个文件,每一个网页被访问时WP都将试图加载 index.php。

主题的文件调用结构，如下图所示：



## 插件目录结构

插件目录结构如下图：

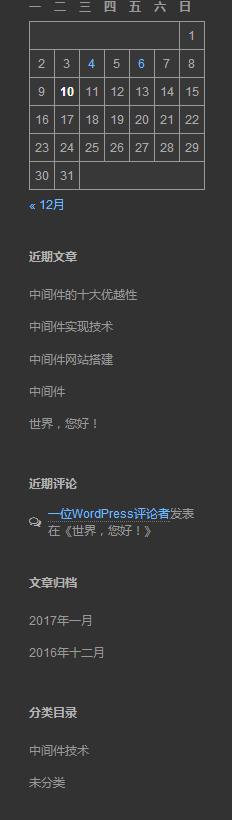


# 应用结果

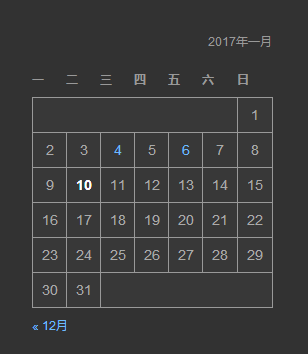
## 站点主页面



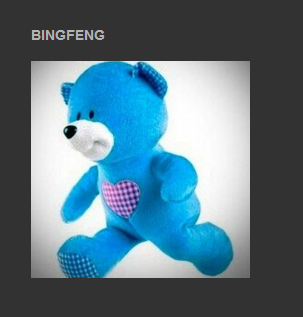
## 侧边栏功能

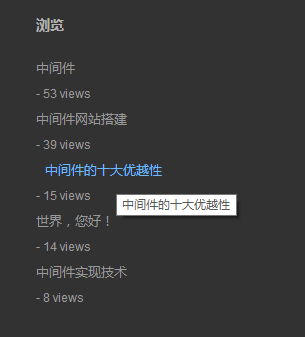
## 日历搜索



## 站点图标



## 浏览统计



## 广告连接



## 打赏评论



# 案例总结

从上面的几个步骤来看，搭建一个独立的博客网站还是挺简单的，这里关键的一点是，技术的架构上使用的是中间件模式，从模版到插件都是一个典型的中间件模式，正因为中间件模式才使得网站的搭建变的简单省开发时间，这也正验证了中间件的十大优势。

Case name: IT technology blog site

Author's name: Chen Wenbin

Author's No.: 2016160193

Completion time: 20170109

Copyright note: this case is only for teaching purposes, not to make any judgments on the case study.

Abstract:  
 The case is a use of WordPress to build a IT technology blog site, the site is mainly to provide IT related information, published in the form of blog in the network.  
The site provides the registration function, the article comments function, the site search function, play the function and advertising function. General technical forum.  
 Key word：  
 Blog website, WordPress, middleware, plug in.

# 二、教学手册

## IT 技术博客网站

## 教学目的与用途：

1. 本案例主要适用于中间件实现技术课程，也适用于WordPress建站课程。
2. 本案例的教学目的掌握中间件的原理以及实现技术。

## 启发思考题：

1. 搭建这个网站使用了什么技术？
2. 搭建该网站主要运用了什么技术思想？
3. Wordpress在网站的搭建中起到了什么作用？
4. 从中间件实现技术上如何看待这个网站的搭建？

## 分析思路：

1. 从中间件技术考虑该网站。
2. 从云服务技术考虑该网站。

## 理论依据与分析：

中间件相关理论依据。

## 背景信息：

Wordpress的强大技术支持。

## 关键要点：

1. 模版。
2. 插件。

## 建议的课堂计划：

课堂演示网站的搭建过程及所做的相关操作。

## 案例的后续进展:

定期的维护，发布相关的IT技术文档。

## 相关附件:

代码路径https://github.com/bingfengcwb/middleware-2016/2016160193

网站路径http://2016160193.applinzi.com/