今天的目标：

<?php

echo “hello world”;//配置好了apache

echo “<br>现在时间：”.date(“Y-m-d H:i:s”);//配置好了php

$link =mysql\_connect(‘localhost’,”root”,”123”);//配置好了mysql

?>

目标2：

能够建立多个站点，比如：

[www.abc.com 能访问一个网站](http://www.abc.com能访问一个网站)

[www.php39.com](http://www.php39.com) 能访问另一个网站

**PHP介绍**

缩写：Personal Home Page =====>php:HypeText Preprocessor

历史: 1995---1997---1998----2000----2005PHP5

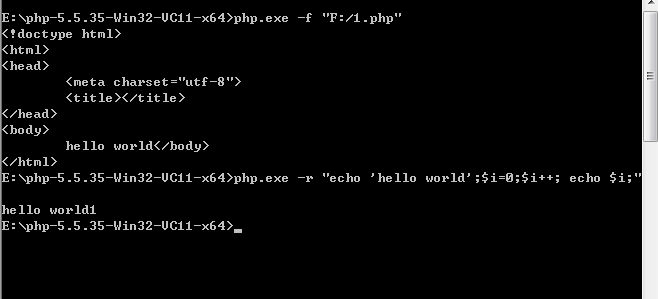
应用范围：

**PHP的应用领域**

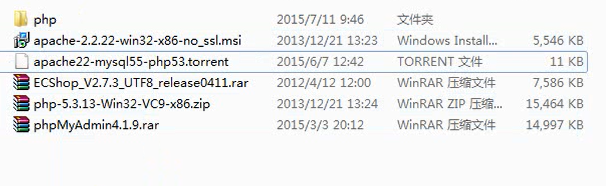
**命令行脚步**

运行php文件：php.exe -f “php文件路径”

运行php代码：php.exe -r ”php脚本代码”



Php 作为一门语言，本身可以是一个纯绿色版的“文件夹”--称之为php语言包

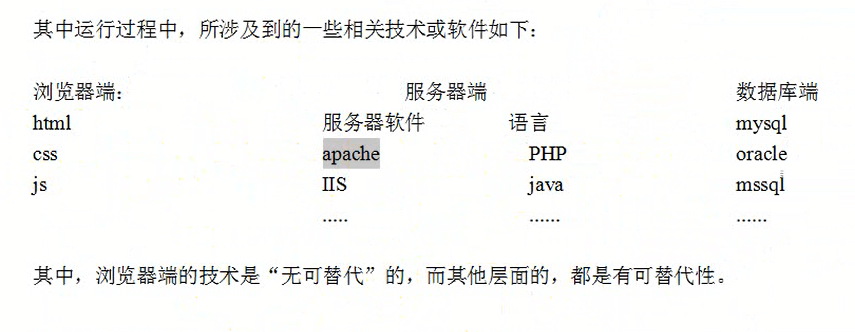


Web 服务端脚步：主要作用

应用程序图形界面

**PHP运行环境**

Web软件运行环境

**浏览器运行流程**

浏览器输入地址，回车（发送请求）

根据规则找到对应web服务器。规则如下：

1、首先在本机hosts文件中找对应的IP

Hosts文件位置：操作系统/system32/drivers/etc/

注意：需要以管理员身份打开编辑器，才能编辑保存它

1. 如果hosts中没找到，则到互联网上找对应的IP（DNS服务器中）
2. 如果还是没找到，则终止请求，返回“找不到服务器”提示

Web服务器取得对应的文件

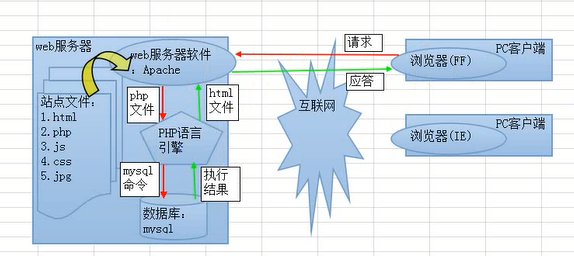
如果需要，对该文件进行一定的处理（比如php代码执行）

将文件（或执行结果文件）返回给浏览器（返回应答）

浏览器接收文件

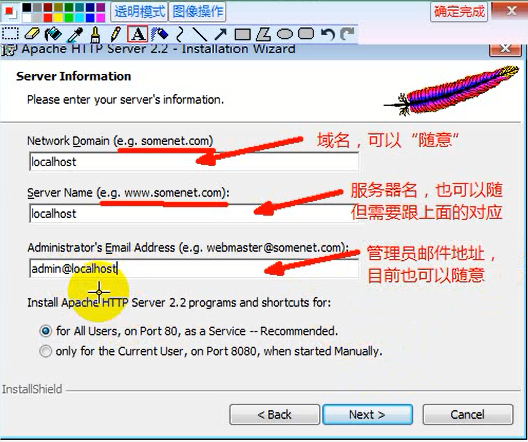
浏览器处理文件（显示、执行）

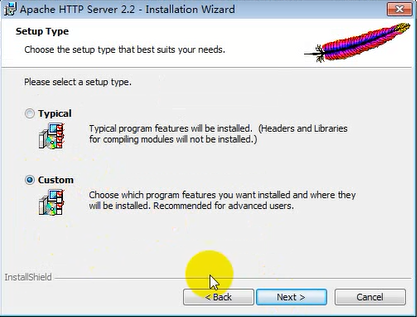
## PHP网页运行原理

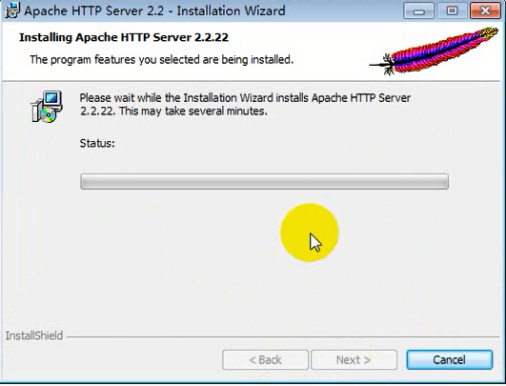


## Apache安装





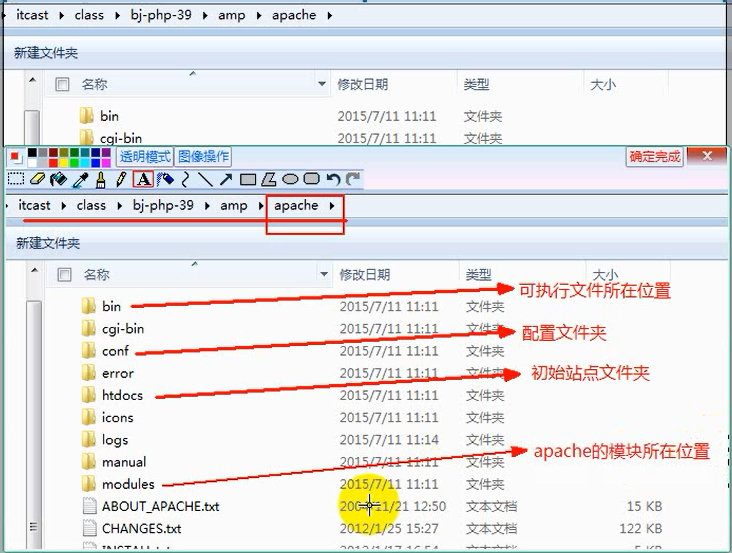






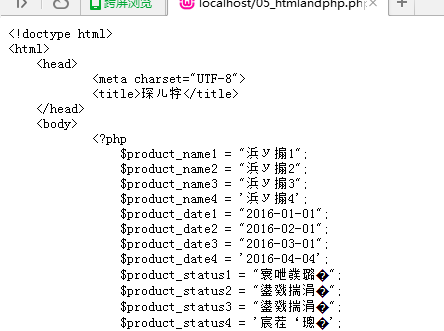
安装成功 浏览器出现 it works!

Apache 安装后的目录结构



## 配置Apache以运行php程序

将刚才的网页1base.php 放到apache的htdocs文件中，并通过浏览器“浏览”：



发现，根本无法运行php！

其实我们就需要配置apache了：

分2步

第一步

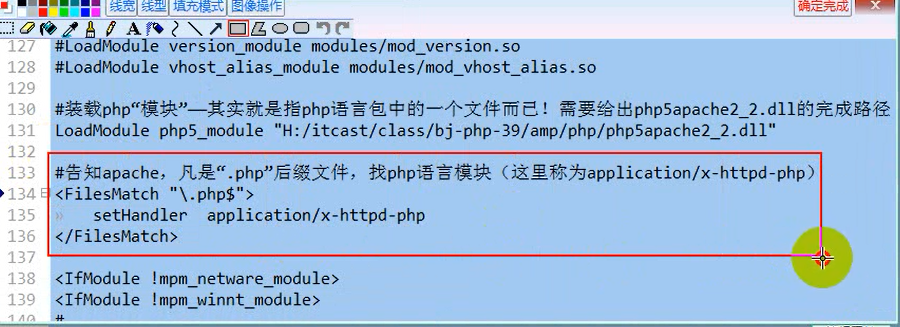
装载php“模块”——其实就是指php语言包中的一个文件而已

php5apache2\_4.dll

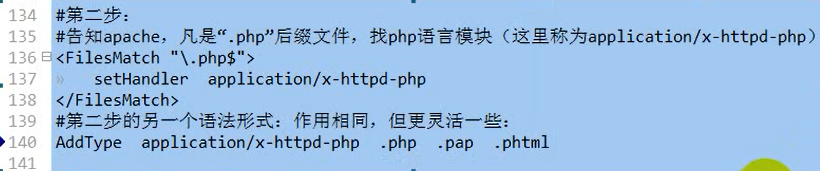
LoadModule php5\_module "E:/wamp/php/php5apache2\_4.dll"

第二步

执行php后缀的文件，去“找”这个php模块处理（执行）固定写法



另一种写法



重启apache z再次刷新页面

### 补充一个检测apache的配置语法的命令：

命令所在位置

Apache 安装位置/bin/httpd.exe

命令使用方式：

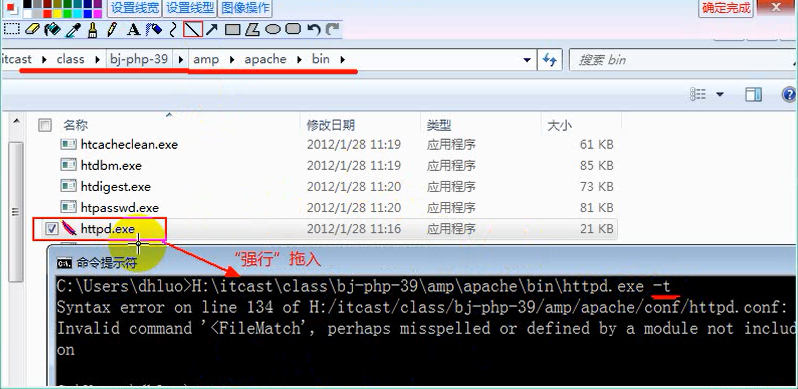
Httpd,exe -t

命令结果

如果配置文件httpd.conf没有语法错误，就返回: syntax OK

如果有错误，则提示第几行错误！

用法1：强行拖入法



用法2：配置 环境变量法

和java配置环境变量一样

## PHP时区配置

运行php代码发现时间不对相差8个小时

这是因为我们还没有配置php语言模块（语言引擎）的运行时区！

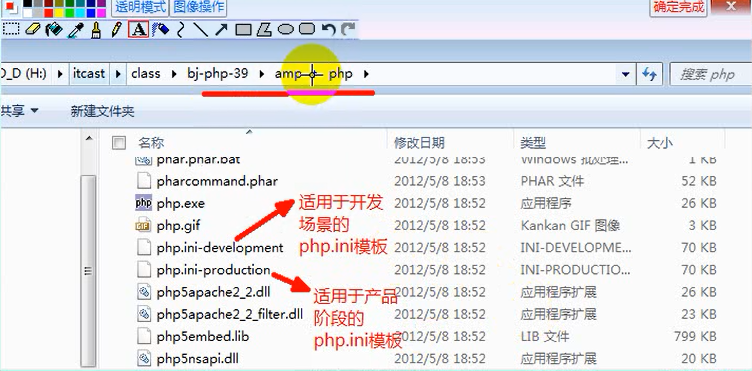
需要区修改php的配置文件:

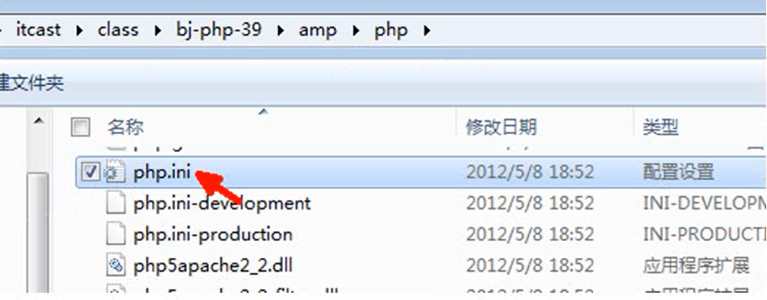
Php配置文件的位置:

Php语言包/php.ini

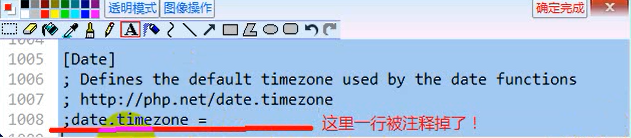
初始，并没有该文件的直接版本，而是提供了2个选择性版本

我们采用php.ini-development,改其后缀名就可以了！最好复制一份





然后打开该文件，并查找关键字：timezone



修改为



在apache的配置文件中（httpd.conf）,设置php.ini文件的路

让apache的配置知道，php.ini的所在位置（此时不要写出php.ini的文件名,只要目录）

PHPIniDir “H:/itcast/class/bj-php-39/amp/php”

然后，还是要重启apache,然后，重新刷新刚才显示时间的网页

## Php的模块配置

理解一件事情：

Php(语言模块)是apache的众多模块之一

Mysql是php语言中的众多模块之一

目标：运行链接mysql语句

第一步：

在php.ini文件中，设定模块所在的路径（查找关键字extension\_dir）:

初始时被注释

; Directory in which the loadable extensions (modules) reside.

; http://php.net/extension-dir

; extension\_dir = "./"

; On windows:

; extension\_dir = "ext"

修改为:

; Directory in which the loadable extensions (modules) reside.

; http://php.net/extension-dir

; extension\_dir = "./"

; On windows:

extension\_dir = "F:/wamp/php/ext"

第二步:

打开mysql模块就可以了(查找关键字php\_mysql.dll)

;extension=php\_mysql.dll

修改为:

extension=php\_mysql.dll

安装mysql 即可

端口监听

Listen 80

主机 （服务器名的设置）

ServerName www.aa.com:80

站点的设置 （服务器文件夹的设置）

DocumentRoot "D:/www" 设置站点

<Directory "D:/www">

Options Indexes

Order Deny,Allow

Allow from all

DirectoryIndex base.php 设置主页面

</Directory>

# 目录访问权限

一个站点,就是一个”目录”。

每个目录,都应该给其设定“访问权限”，语法如下：

<Directory “要设置的权限的目录路径”>

#设置当一个请求中没有给定请求的文件名又没有默认网页(主页)的时候，显示文件列表

Options Indexes

#设置权限控制的先后顺序，只有2种:Deny,Allow（先拒绝，后允许）Allow,Deny（先允许，后拒绝）

Order Deny,Allow

#Deny用于设定要拒绝访问的来源地址或服务器,几乎没有实际意义!

Deny from 拒绝名单 比如:deny from 192.168.3.8 172.180.4

Allow from 允许名单 比如: Allow from all

#设置该文件夹下的”默认网页”(首页),可以设置多个,用于请求中不带文件名的时候自动使用该文件作为返回文件

DirectoryIndex index.php index.html main.php default.php abc.php 123.html

</Directory>

## 目录访问权限中的”分布式权限”:

对于某个目录的访问权限,我们在apache中使用Directory来设定,但可以不够方便：

1. apache 配置修改,必须重启apache
2. 某个站点中的“下级目录权限”访问权限，不方便单独设置，

这种，可以称为“集中式权限”：

分布式权限，就是：针对某个网站的任何一个下级目录，都可以单独去设置访问权限，并且可以不重启apache就生效。

做法：分2步：

第1步：

在该站点对应目录权限的Directorty配置中添加一条：

<Directory "D:/www">

Options Indexes

Order Deny,Allow

Allow from all

DirectoryIndex base.php

#下一行，表示当前目录，允许分布式权限配置

AllowOverride all

</Directory>

第2步:

在该目录下的任何子目录中,创建一个特殊的文件(.htaccess)，在该文件中写所需要的“权限内容”(几乎跟Directory一样)

Deny from all

Allow from 192.34.27.1 //只允许这个访问这个目录

# 多站点的配置

分3步:

第一步：在apache的主配置文件httpd.conf中，“打开”多站点虚拟主机配置文件项(关键字:vhosts)

初始： # Virtual hosts

#Include conf/extra/httpd-vhosts.conf

修改为:

# Virtual hosts

Include conf/extra/httpd-vhosts.conf 载入这个文件

第二步：

在httpd-vhosts.conf文件中，设定希望提供“多站点”服务的IP地址和端口号,通常默认就是：\*：80，其中“\*”代表所有IP，默认其实apache 安装好之后就设置好了，无需手动设置（关键字NameVirtualHost）

NameVirtualHost \*:80

第三步：

然后就一个站点一个站点的加上！形式如下:

<virtualhost \*.80>

#这里设置站点1的各项信息

</virtualhost>

<virtualhost \*.80>

#这里设置站点2的各项信息

</virtualhost>

注意：

1. 先把httpd-vhosts.conf文件中的2个初始安装生成的站点，删除!!!
2. 每个站点的设置项，主要就3个:

ServerName,

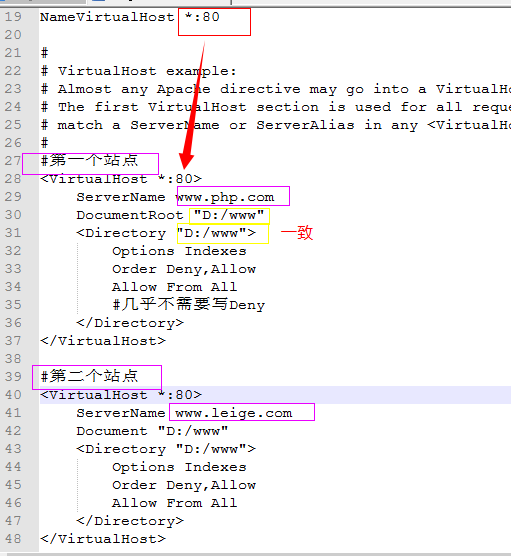
DocuemntRoot;

<Directory>...</Directory>

1. 在多站点配置中,第一个站点被称为“默认站点”，如果某个请求的域名无法跟任何一个站点“匹配”，就会当做请求的是默认站点去处理（应答）
2. 一旦实现了多站点配置，则最初安装的时候的“默认站点”就失效了，我么可以将“localhost”当做一个普通站点重新设置就可以了，也可以将其放在第一个位置又重新作为

默认站点

例如:



一个小细节知识



# 目录别名（虚拟目录）的设置

某个站点www.abc.com所对应的文件夹下有一个目录为d1，则可以这样访问该目录：

[www.abc.com/d1](http://www.abc.com/d1)

这个d1就称为就称为“真实目录”

虚拟目录就是:

该站点下不存在某个目录,但却可以使用同样的语法形式，去访问，比如：

[www.abc.com/d2](http://www.abc.com/d2) //假设该站点目录下根本就没有d2这个目录,此时却可以访问

实现这种技术，就是所谓“目录别名”(虚拟目录)的设置，如下：

在一个站点中，做如下2步:

第一步：

设置目录别名的名字和其对应的“实际真实目录（文件夹位置）”:

Alias /目录别名的名字 “实际真实目录（文件夹）的位置”

第二步:

设置该真实目录的访问权限（否则是禁止访问的）：

<Directory “真实目的路径”>

...

</Directory>

举例：

