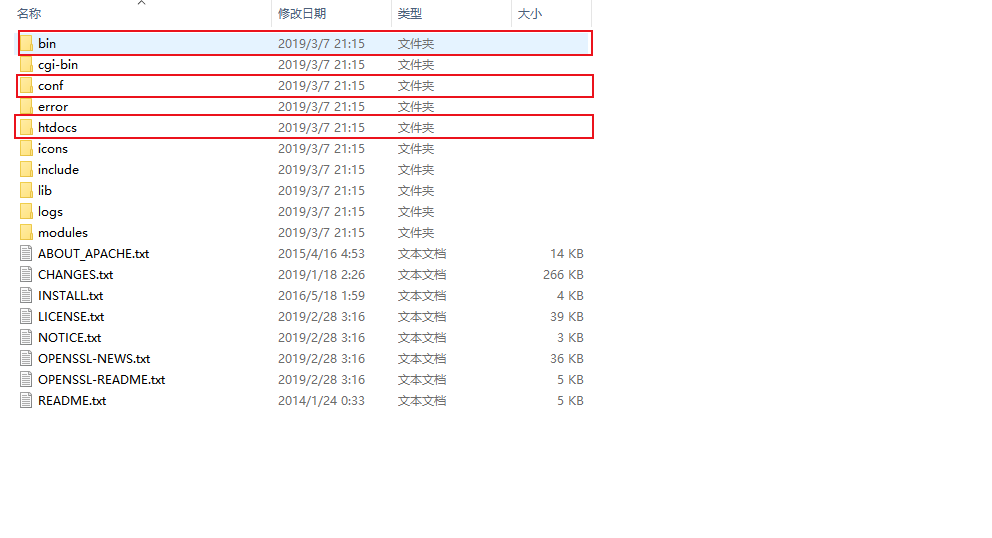
# PHP学习笔记

## 序言：PHP环境搭建

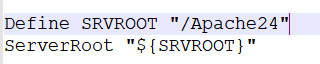
学过jsp和Java的servlet的同学应该搭建过jsp运行环境，相同之处在于，这两者都是对web环境的搭建，在我看来PHP环境搭建稍微复杂一点，因为要下载的东西比较多，要配置的东西也比较多。

|- 首先第一步：下载Apache。学过jsp的同学对Apache应该不陌生，没错，jsp就是运行在Apache的Tomcat服务器上的。但是此处并不是要下载Tomcat，而是下载Apache本身。具体的下载教程此处不做过多解释，百度即可。

|- 下载完Apache后，解压缩即可，然后放到自己喜欢的目录下，接下来是对Apache目录结构的讲解，以及对Apache的配置。

|- Apache解压后，目录中的这几个文件夹必须注意：bin目录、conf目录和htdocs目录。bin目录放着支持Apache运行的所有命令，如果进入目录就可以看到大量的可执行文件和dll库（因为Apache是用c语言写的）；conf目录中包含了Apache所有的配置文件，稍后对Apache的配置就是在这个文件中；htdocs目录与我们的关系就大了，这是我们以后写代码的地方，同样的，学过jsp的同学对虚拟目录这个词应该不陌生，当然没有学过也不要紧，虚拟目录就是我们放置项目的地方，而且虚拟目录是可以自己配置的。

|- 接下来就是对Apache的配置。进入conf目录，选择httpd.conf，修改sevroot的值，大概在文件的第38行，将这个值修改为自己的Apache所在的绝对路径，我这里是：D:\Apache。把“/Apache24”修改为这个路径即可。

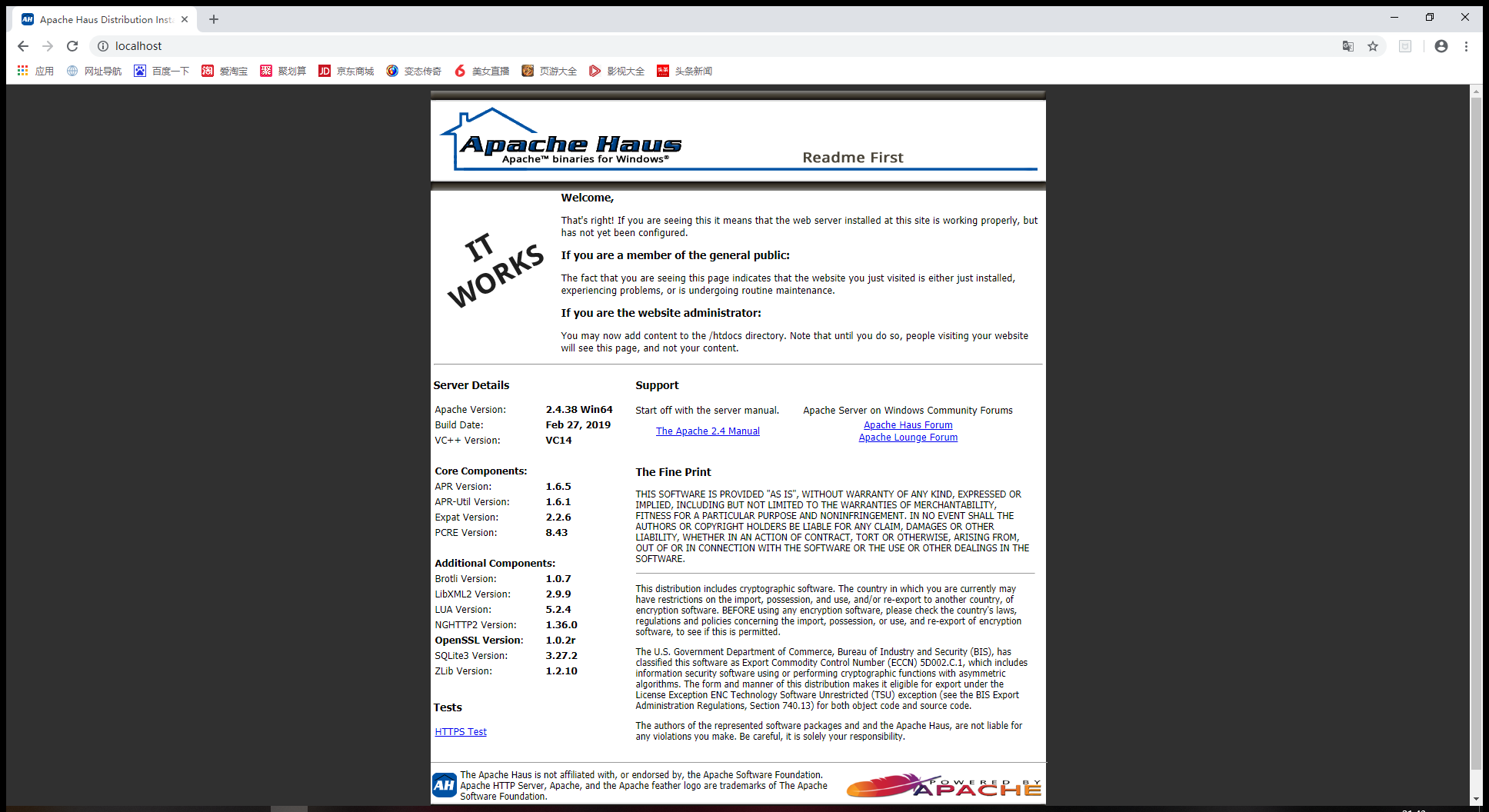


|- 修改端口号和ServerName（仍然是httpd.conf文件）。Apache默认选择的是80端口，一般可以不修改，你可以修改为其他不会被占用的端口.如果端口号不做修改则ServerName也不用修改，两者的修改是同步的，Listen后面是端口号，localhost后面的端口要与这个端口保持一致，我在这里就使用80端口。

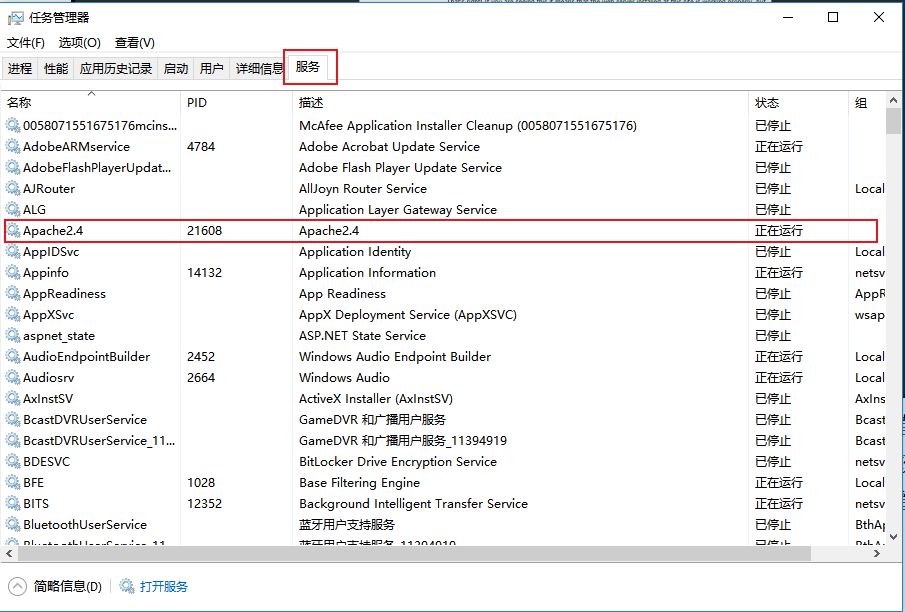




|- 此时配置已经完成，只要启动Apache服务，我们就可以在浏览器中访问Apache了，启动Apache服务方法：进入命令行，切换到Apache的bin目录（利用cd命令），执行httpd.exe。此时Apache服务已经启动，打开浏览器，在地址栏输入“localhost:80”（输入内容根据上面对端口和ServerName的配置而定）。如果能看到这样的画面，那么恭喜你，Apache安装配置成功。



|- 你可能会发现，在命令行执行httpd.exe后，命令行没有进行返回，会一直执行知道你按下Ctrl+Z。这样的话，很多人会感觉很不爽，感觉有东西在占用这自己的系统资源。还有一种方式，将Apache安装成系统服务，然后通过指令开启关闭服务。首先以管理员方式打开命令行，同样还是进入到Apache的bin目录下，执行命令httpd -k install将Apache安装为系统服务，然后打开任务管理器，在服务这一项中找到Apache，这里可以启动或关闭Apache服务。



|- Apache安装完成才是第一步，接下来下载PHP。既然是与Apache进行搭档，那么下载的时候最好选择线程安全的版本。下载完后直接解压到自己喜欢的目录，我这里放在D:/Apache/PHP 7.3.3文件夹（我这种放置方式可能并不妥当，建议建一个大的文件夹，将Apache和PHP放在同级目录下）。然后复制一下PHP根目录中的php.ini-development或php.ini-production为php.ini，作为PHP的配置文件（复制到PHP根目录即可）。



|- 打开php.ini进行配置（配置文件中英文的分号表示注释，删除分号即取消注释）。首先修改拓展文件的路径，大概在750行（extension\_dir = “ext”），将这句代码取消注释，或者将右边的值直接修改为PHP的绝对路径（extension\_dir = "D:\Apache\PHP 7.3.3\ext"），建议选择修改为绝对路径。

|- 下一步，修改需要加载的拓展文件，只需要对部分代码取消注释即可。注意标红的都是没有取消注释的。（大概在第900多行）

|  |
| --- |
| extension=php\_bz2.dll  extension=php\_curl.dll  extension=php\_fileinfo.dll  extension=php\_gd2.dll  extension=php\_gettext.dll  ;extension=php\_gmp.dll  ;extension=php\_intl.dll  ;extension=php\_imap.dll  ;extension=php\_interbase.dll  ;extension=php\_ldap.dll  extension=php\_mbstring.dll  extension=php\_exif.dll ; Must be after mbstring as it depends on it  extension=php\_mysql.dll  extension=php\_mysqli.dll  ;extension=php\_oci8\_12c.dll ; Use with Oracle Database 12c Instant Client  extension=php\_openssl.dll  ;extension=php\_pdo\_firebird.dll  extension=php\_pdo\_mysql.dll  ;extension=php\_pdo\_oci.dll  extension=php\_pdo\_odbc.dll  extension=php\_pdo\_pgsql.dll  extension=php\_pdo\_sqlite.dll  extension=php\_pgsql.dll  ;extension=php\_shmop.dll    ; The MIBS data available in the PHP distribution must be installed.  ; See http://www.php.net/manual/en/snmp.installation.php  ;extension=php\_snmp.dll    extension=php\_soap.dll  extension=php\_sockets.dll  extension=php\_sqlite3.dll  ;extension=php\_sybase\_ct.dll  extension=php\_tidy.dll  extension=php\_xmlrpc.dll |

|- 下一步设置默认时区（950行左右），设置date.timezone = Asia/Shanghai

|- 下一步配置sll（1900行左右），设置openssl.cafile = cacert.pem

|- 接下来在Apache的conf目录下打开httpd.conf配置一下Apache，让Apache和PHP能协同工作。首先设置DocumentRoot（大概250行），默认是htdocs目录（配置这个就是配置虚拟目录）。这个目录是你之后部署项目的目录，最好用绝对路径表示，我这里将其修改为“C:\Users\fuhao\Desktop\software\_engineering\web”。我以后的项目全部放在这个目录下。

|- 修改默认索引，用以支持PHP（大概280行），在两个标签中间添加入下：

DirectoryIndex index.html index.php index.htm

|- 开启rewrite功能，将这行代码注释取消即可（大概160行）

LoadModule rewrite\_module modules/mod\_rewrite.so

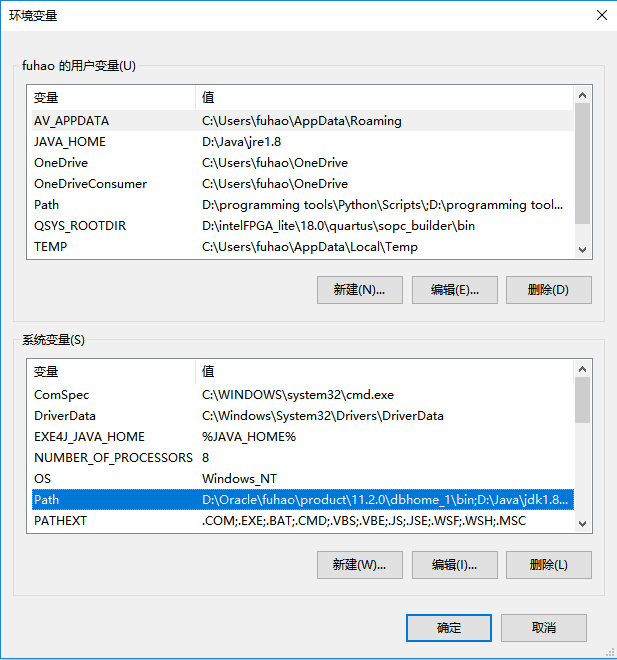
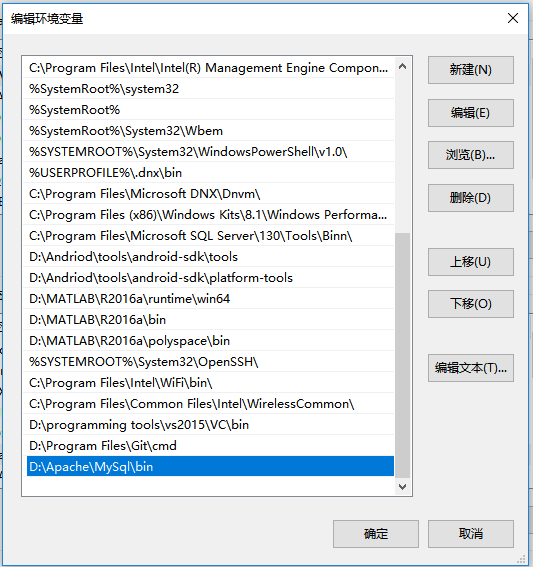
|- 加载PHP模块，注意绝对路径，如果是PHP 7，在httpd.conf文件末尾添加如下代码

|  |
| --- |
| LoadModule php7\_module D:\Apache\PHP 7.3.3\ php7apache2\_4.dll  <IfModule php7\_module>  PHPIniDir "D:\Apache\PHP 7.3.3"  AddType application/x-httpd-php .php  AddType application/x-httpd-php-source .phps  </IfModule> |

如果是PHP 5则做相应的更改即可。

|- 最后一步是下载MySQL数据库，用PHP开发时，PHP+Apache+MySQL是一套架构。MySQL的下载很简单，直接去官网下载即可，下载完后解压缩，放到自己喜欢的目录，然后配置一下环境变量。

|- 配置环境变量也不难，桌面鼠标右键“我的电脑”，选择属性，再选择高级系统设置，再选择环境变量，再系统变量中找到Path变量（如下左图），双击Path变量，在右侧找到新建，把MySQL的bin路径放到新建的变量中（如下右图），一直确认即可。

|- 环境变量配置完后，在MySQL根目录下找到一个叫做my.ini的配置文件，如果没有就去百度复制一个。打开my.ini，把下面两个变量按照自己的安装路径进行修改，basedir就是MySQL的根目录。



|- 然后需要对MySQL进行第一次登录，第一次登录需要做的事在第五部分讲解了，直接去看。

这样的话，PHP环境搭建完毕！

## 一、PHP代码

所有的php代码都需要写在**<?php 代码 ?>**这一标签中间，所有写在标签之外的内容都属于HTML代码。

php的每一句代码都以分号结束，但是最后一句代码可以不写分号。

**php注释：**

1. // 当行注释
2. # 当行注释
3. /\* 多行注释 \*/

### php向网页输出

下面的代码会在网页上输出hello world

|  |
| --- |
| <?php  **echo** 'hello world';  ?> |

用echo向网页输出的实际上是html代码，如果输出内容种包含html标签，则会被浏览器渲染，下面的代码会在网页上输出加粗的hello world

|  |
| --- |
| <?php  **echo** '<b>hello world</b>';  ?> |

### 1.2、php变量

所有的php变量以$符号开头，用字母数字下划线组成，下面这段代码与上面执行结果相同，只不过这里利用了一个a0变量，“=”在这里是赋值运算符。**无论是在定义还是在使用变量的时候都必须带有$符号**。

|  |
| --- |
| <?php  $a0 = '<b>hello world</b>';  **echo** $a0;  ?> |

### 1.3、php数据类型

php中共有九种基础数据类型，包括四种标量类型，三种复合类型，两种特殊类型。

|  |  |
| --- | --- |
| 四种标量类型 | 举例 |
| int类型（整型） | $a0 = 1; |
| float类型（浮点型） | $a1 = 1.5; |
| bool类型（布尔型） | $a2 = true; |
| String类型（字符串类型） | $a31 = ‘Hello World’;（单引号方式） |
| $a32 = “Hello World”;（双引号方式） |

字符串用双引号时，会识别字符串中的其他变量名

例如：

|  |
| --- |
| <?php  $a0 = 1;  $a1 = 'string$a0';  $a2 = "string$a0";  echo $a1;  echo '<br/>';  echo $a2;  ?> |
| string$a0 string1 |

|  |  |
| --- | --- |
| 三种复合类型 | 举例 |
| 数组类型 | $a41 = [1, 2, 3];（方括号方式） |
| $a42 = array(1, 2, 3)（函数方式） |
| 对象类型 | $a5 = new stdClass; |
| 回调类型 |  |

|  |  |
| --- | --- |
| 两种特殊类型 | 举例 |
| 资源类型 |  |
| 空类型 | $a7 = null; |

var\_dump()函数能够输出一个变量以及变量的类型。

|  |
| --- |
| <?php  $a0 = 1;  $a1 = 'string$a0';  $a2 = "string$a0";  var\_dump($a0, $a1, $a2);  ?> |
| int(1)  string(9) "string$a0"  string(7) "string1" |

### 1.4、数据类型的转换

在执行特定的数学运算的时候，如果进行运算的两个变量（常量）的数据类型不相同，则会自动进行数据类型的转换。

在进行数学运算（如加减乘除）时，除浮点类型外，其他的类型均当作整型处理。

强制类型转换，可以在一个值前面加上目标的类型，实现强制类型转换。

### 1.5、数组

数组里的每个元素可以是任意的类型（包括数组），也就是说一个数组中的元素可以是不同类型的。数组的下标（也称为键）可以是整型或字符串型。

**数组分类**：

|- **索引数组**：数组的键自动生成，由0开始

$arr = [1, 2, 3];

|- **关联数组**：数组的键可由程序员指定（整型或字符串型）

$arr = [ ‘a’ => 1, ‘b’ => 2, ‘c’ => 3];

**数组的增删改查**：

|- 增：向数组中添加新的键值对，**数组名[键] = 值;**

|- 删：从数组中删除一个键值对，**unset(数组名[键])**;

|- 改：修改数组中的一个元素， **数组名[键] = 新值**;

|- 查：查找数组中的一个元素， **数组名[键]**;

数组包含大量的系统函数，用来对数组进行各种操作。

## 二、数据的发送与接收

数据从浏览器端向服务器端发送，利用get和post两种方式进行发送。

以form.html作为浏览器端，formsave.php作为服务器端，通过浏览器端提交表单，服务器端接收数据为例。

form.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset='UTF-8'>  <title>数据发送与接收</title>  </head>    <body>  <form action='formsave.php' method='post'>  <input type='text' name='aa' />  <input type='text' name='bb' />  <input type='submit' value='提交'>  </form>  </body>  </html> |

这里的form标签中间就是要提交的表单，而form标签中的action属性表示表单将要提交到何处，这里表单需要提交到formsave.php中，method属性表示以何种方式进行提交，主要有两种方式：get和post。

但是，想要formsave.php能接收到表单中两个输入框的内容，还需要input标签的name属性明确标识，这样才能完成表单的提交和接收。

formsave.php

|  |
| --- |
| <?php  $aa = $\_POST['aa'];  $bb = $\_POST['bb'];  var\_dump($aa, $bb);  ?> |

当表单提交后，formsave.php想要接收输入框的内容，必须用原name属性进行标识，而且要通过 **$\_POST[]** 进行内容的获取。而通过[]获取值，说明$\_POST是一个数组，原name属性就是数组里面的键，表单内容就是数组里面的值。

当表单的form标签的method属性选择get时，服务器端进行接收时必须使用$\_GET进行接收。当使用get方式进行发送时，表单中的数据会直接在网址中显示。

这里的$\_POST和$\_GET都是php预定义变量。

## 运算符和流程控制

### 3.1、运算符

重点：

|- 字符串连接运算符“.”，用来连接两个字符串；

|- 绝对相等判断运算符：===，判断运算符左右两端的数据的类型是否相同，数据值是否相等；

|- 相等判断运算符：==，判断运算符左右两端数据的值是否相等（可以通过自动转换类型）

|  |
| --- |
| $a0 = ‘1’;  $a1 = 1;  $a0 == $a1; // 返回true |

### 3.2、流程控制

if分支语句

|  |
| --- |
| if(判断条件){  条件为真执行;  } elseif(判断条件) {  条件为真执行;  } else {  条件为假执行;  } |

switch-case分支语句

|  |
| --- |
| switch(变量){  case 值1:  变量和值1相同，则执行该段代码;  break;  case 值2:  ...  default:  ...  break;  } |

for循环控制，与c语言，java语言相同

foreach循环，用于数组和对象上，提供遍历数组的简单方式

|  |
| --- |
| $a = [‘a’ = 1, ‘b’ = 2, ‘c’ = 3];  foreach($a as $key => $value){  echo $key;  echo $value;  } |

## 四、函数与对象

### 4.1、函数的定义与使用

函数定义：

|  |
| --- |
| function 函数名(参数) {  函数体  } |

注意：所有的php代码都需要写在<?php 代码 ?>中，函数定义也不例外。

函数不仅可以传递参数，还可以有返回值，在函数体内用return语句返回。

|  |
| --- |
| <?php  function test() {  echo ‘hello’;  }  test();  ?> |
| <?php  $a = ‘hello’;  function test($input) {  echo $input;  }  test($a);  ?> |
| <?php  function add($a, $b) {  return $a + $b;  }  echo add(10, 5);  ?> |

### 4.2、类与对象

类的定义

|  |
| --- |
| class 类名称 {  定义属性;  定义方法;  } |

对象实例化

|  |
| --- |
| $obj = new 类名称(); |

|  |
| --- |
| <?php  // 定义类Person  class Person {  // 定义类属性  public $name = “张三”;  public $age = 18;  // 定义类方法  public function toString() {  return $this->name . $this->age;  }  }  // 实例化Person对象  $per = new Person();  // 输出属性name的值  echo $per -> name;  echo $per -> age;  // 调用类方法  echo $per -> toString();  ?> |

利用 -> 运算符，从类对象中直接调用类的属性和类的方法，但是只有public权限的属性和方法才能直接调用。

在类的内部，可以利用$this表示当前对象获取类属性，调用类方法。

### 4.3、魔术方法

魔术方法指的是没有明确代码调用，但是却执行的方法。例如：类的构造方法

|  |
| --- |
| <?php  // 定义类Person  class Person {  // 定义类属性  public $name = “张三”;  public $age = 18;  **// 定义类的构造方法**  **public function \_\_construct($name, $age) {**  **$this->name = $name;**  **$this->age = $age;**  **}**  // 定义类方法  public function toString() {  return $this->name . $this->age;  }  }  // 实例化Person对象  $per = new Person();  // 输出属性name的值  echo $per -> name;  echo $per -> age;  // 调用类方法  echo $per -> toString();  ?> |

类的构造方法在实例化对象的时候会自动调用分配内存。可以利用构造方法为实例化对象做一些初始化。与其他函数方法一样，构造方法也可以传递参数。

同时强调一下，php中所有的魔术方法都是以两个下划线开头。

### 4.4、包含文件

利用include包含其他文件，**include(文件路径);**

当利用include包含其他文件后，就如同两个文件合并为一个文件一起执行。一般将类或函数定义在一个单独的文件，然后利用include将其包含进来，可以直接使用所定义的类和函数。

include和include\_once的区别：

当同一个文件被多次加载的时候，include会报错，表示类或函数重复定义，而include\_once只会将文件加载一次，不会报错。

input.php（定义一个类，包含从客户端发送来的表单的属性及验证）

|  |
| --- |
| <?php  class input {  private $user;  private $content;    public function \_\_construct($user, $content) {  $this->user = $user;  $this->content = $content;    if(!$this->isValid($this->user)){  die('用户名不合法');  }    if(!$this->isValid($this->content)) {  die("留言内容不合法");  }  }    // 判断参数内容是否合法  public function isValid($input) {  // 敏感词数组  $mgc = ['张三', '李四', '王五'];  if($input == '') {  // 参数为空，不合法  return false;  }    foreach($mgc as $value) {  if($input == $value) {  // 参数为敏感词，不合法  return false;  }  }    return true;  }  }  ?> |

formsave.php（包含input.php，实例化input类对象，完成验证）

|  |
| --- |
| <?php  include('input.php');  $content = $\_POST['content'];  $user = $\_POST['user'];  $input = new input($user, $content);  ?> |

## 五、MySql数据库

启动数据库服务的步骤很繁琐。

首先从MySQL官网上下载解压缩版的压缩包，下载完成后解压到自己喜欢的目录中，然后将MySQL的bin目录配置到环境变量的path变量中；

然后从文件管理器中进入到MySQL的解压目录，新建一个my.ini配置文件。配置文件中需要设置basedir和basedata两个目录；（百度有教程）

然后以管理员身份启动命令行，进入到MySQL的bin目录，输入如下指令：

|- **net start mysql**

如果是第一次安装，需要先执行mysqld -install安装MySQL服务，后再执行一条mysqld -initialize指令来创建上述的basedata目录，之后再执行上述指令连接MySQL服务。

当使用完数据库之后要关闭服务，在命令行中输入**net stop mysql**指令关闭服务。

当然，这其中可能会出现一些意想不到的问题。比如说第一次连接启动数据库服务后，用root用户登录时，会让你输入密码，但是你并不知道初始密码是多少，这时你随便输入一个密码，会产生错误信息，在MySQL目录下搜索\*.err，错误日志中会告诉你初始密码（这在网上都有教程），第一次登陆MySQL后需要修改密码才能进行各种操作，修改密码网上也有教程，这里不做过多叙述。我要强调的是，在修改完密码后，要修改密码的加密模式，不然在navicat软件上无法登陆，如何修改加密方式，在网上也有详细教程。

到此为止，root用户的密码修改完毕，加密方式也修改完毕，只需要启动数据库服务就能直接登陆数据库了，无论是从命令行还是软件。

## 六、前端学习

HTML

CSS

JavaScript：基础语法、jquery、DOM、Ajax

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |