

php 第一课 一种程序语言

从 C 语言说起

回忆

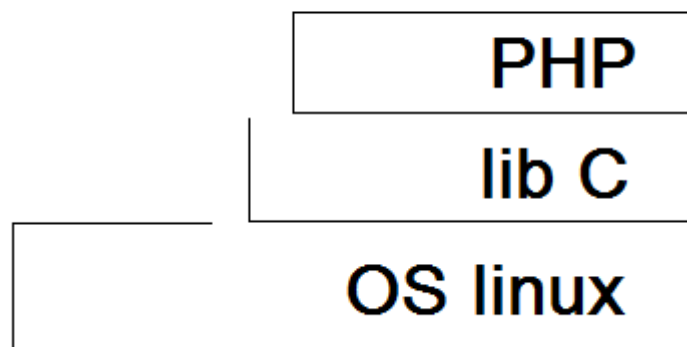
大名鼎鼎的 c 语言, 大家都知道, 语法也比较了解
大家都知道 c 语言需要编译,php 则不需要
但是 php 的语法确和 c 一样一样的!!

解释语言

不同于 c 语言, php 不用编译,叫解释语言,那么谁来解释呢?

php 解释器,也就是我们安装的 php

我们安装的 php 也就是 php 解释器?php 解释器其实是一个语言虚拟机
它完成 php 程序的解释并通过调用底层的 c 语言库实现功能



思考

c 语言程序谁在解释?可执行程序谁在解释?

强类型和弱类型

回忆

说 C 语言是强类型的,php 是弱类型的
什么意思呢,指 php 里的变量,不必指明什么类型?

问题

怎么做到的?

php 解释器作了手脚,就像我们让人翻译文章,要是翻译者都知道你说的任何一个人的职务,那你就不用事先说出来了

仅此而已

面向对象

C++支持面向对象

Php 支持面向对象

Php 有什么区别吗

做为 一门语言,看来没什么区别!!

Php 的安装

[回忆] apache/mysql 安装

<http://cn2.php.net/distributions/php-5.3.0.tar.gz>

实践!!!

[视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bphp01-%E4%B8%80%E9%97%A8%E7%A8%8B%E5%BA%8F%E8%AF%AD%E8%A8%80.rar>

php 第二课 写一个 php 程序

Hello 程序的出生

回忆

学习 C 语言的时候,第一个总是要写的 hello world 程序

示例

echo 输出函数

```
vi hello.php
<?php
    echo 'hello world!';
?>
php hello.php
```

运行

1.用 php 解释器

2.浏览器打开

将 hello.php 拷贝到 web 目录下!

OK,成功写了第一个程序

[视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bphp02-hello%E7%A8%8B%E5%BA%8F%E7%9A%84%E5%87%BA%E7%94%9F.rar>

完成一个 submit

回忆

我们说 http 有两种数据 POST 和 SET, POST 是通过表单提交的!

试验

构造一个注册表单,提交给 php 处理并提示数据检查是否通过

1.构造一个简单的 html,加上表单

```
vi submit.html
<html>
<form action="submit.php" method="post">
Name:<input type="text" name="name"><br>
Phone:<input type="text" name=" phone"><br>
<input type="submit" value="Add">
</form>

</html>
```

检查这个 html 的输出

2. 撰写 submit.php

```
vi submit.php
```

```
<?php
```

```
function is_mobile($str){  
return preg_match("/^((\d{3}\d{3})|(\d{3}\d{3}))?13\d{9}$/", $str);  
}
```

```
function checkdata($name, $phone)  
{  
    if($name == "" || $phone == "")  
        return false;  
    if(false == is_mobile($phone))  
        return false;  
    return true;  
}
```

```
function main()  
{  
    $name = $_POST['name'];  
    $phone = $_POST['phone'];  
    if ( false == checkdata($name, $phone) )  
        echo 'Err!!' . '<br>';  
    else  
        echo 'OK!!' . '<br>';  
}
```

```
main();
```

```
?>
```

3. 联合测试, 调试

OK, 完成!!

[视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bphp03-%E8%A1%A8%E5%8D%95.rar>

构造一个类

目标

构造一个数据类,将姓名和个人信息存储如数据库,并可提供查询接口!

背景

学习数据库操作,用 PDO 实现数据库操作

[提示]PDO 需要 php 编译时加上支持

```
./configure --with-php-config=/usr/local/bin/php-config --with-pdo-mysql=/usr/local/mysql
```

创建数据表 person

```
mysql> create table person (  
    name VARCHAR(100),  
    phone VARCHAR(20)  
);
```

撰写 db.php

Vi db.php

```
<?php  
/*DB*/  
define("_DBHOST", "localhost");  
define("_DBPORT", "3306");  
define("_DBUSER", "root");  
define("_DBPWD", "");  
  
var $_db;  
var $_dsn;  
var $_err;  
  
function init_db($sysname)  
{  
    try {  
        $_dsn = "mysql:host=" . _DBHOST . ":" . _DBPORT . ";dbname=" . $sysname;  
        $_db = new PDO($_dsn, _DBUSER, _DBPWD);  
    } catch (Exception $e) {  
        _err = $e->getMessage();  
        return false;  
    }  
}
```

```
function QuerySQL($sql)
```

```

{
    try {
        $sth = _db->query($sql);
        $s = $sth->fetchAll(PDO::FETCH_ASSOC);
        return $s;
    } catch (PDOException $e) {
        _err = $e->getMessage();
        return false;
    }
}

```

```

function ExecSQL($sql)
{
    try {
        $count = _db->exec($sql);
        return $count;
    } catch (PDOException $e) {
        _err = $e->getMessage();
        return false;
    }
}
?>

```

[视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bphp04-php%E5%92%8C%E6%95%B0%E6%8D%AE%E5%BA%93.rar>

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bphp05-PDO%281%29.rar>

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bphp06-PDO%282%29.rar>

类设计

class person

member

\$name

\$phone

function

person(\$name, \$phone);

insert();

var list();

类实现

vi person.php

<?php

```

include("db.php");
class person {
    var $_name;
    var $_phone;
    var $_OK;
    function person($name, $phone)
    {
        If ( false == Init_db("test_db"))
            $_OK = false;
        $_name = $name;
        $_phone = $phone;

        $_OK = true;
    }

    function insert()
    {
        If($_OK == false)
            return false;
        $sql = "insert into person values('".$_name."', '".$_phone."')";
        return ExecSQL($sql);
    }

    function list()
    {
        If($_OK == false)
            return false;
        $sql = "select * from person";
        return QuerySQL($sql);
    }
};
?>

```

测试这个类

OK,完成

[视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bphp07-%E7%B1%BB%E7%9A%84%E8%AE%BE%E8%AE%A1.rar>

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bphp08-%E7%B1%BB%E7%9A%84%E5%AE%9E%E7%8E%B0.rar>

让多个程序协同起来

目标

将 submit.php, submit.html, db.php, person.php 联合起来,实现一个个人信息的上传,查看的小系统!

实施

修改 submit.php 调用数据库接口,提交数据

vi submit.php

```
<?php
include("person.php");
function is_mobile($str) {
return preg_match("/^((\d{3} \d) | (\d{3} \-)) ?13\d{9} $/", $str);
}

function checkdata($name, $phone)
{
    if($name == "" || $phone == "")
        return false;
    if(false == is_mobile($phone))
        return false;
    return true;
}

function main()
{
    $name = $_POST['name'];
    $phone = $_POST['phone'];
    if ( false == checkdata($name, $phone) )
        echo 'Err!!' . '<br>';
    else
    {
        person1 = new person($name, $phone);
        if (false == person1.insert() )
        {
            echo 'Err!!' . '<br>';
            return;
        }

        echo 'OK!!' . '<br>';
    }
}
```

```
main();  
?>
```

添加 list.php 提供列出接口

```
vi list.php  
<?php  
Include("person.php");  
  
function main()  
{  
    person1 = new person($name, $phone);  
    $arr = array();  
    if (($arr = person1.list()) == false )  
    {  
        echo 'Err!!' . '<br>';  
        return;  
    }  
    for each ($person in $arr)  
    {  
        echo 'name:' . $person['name']. ' ,  phone:' . $person['phone'].'<br>';  
    }  
}  
  
main();  
?>
```

修改 submit.html 添加列出链接

```
vi submit.html  
<html>  
<form action="submit.php" method="post">  
姓名:<input type="text" name="name"><br>  
手机:<input type="text" name=" phone"><br>  
<input type="submit" value="add">  
</form>  
<center> <a href="list.php">list</a></center>  
</html>
```

这就是我们开发 LAMP 程序的过程的真实写照!!

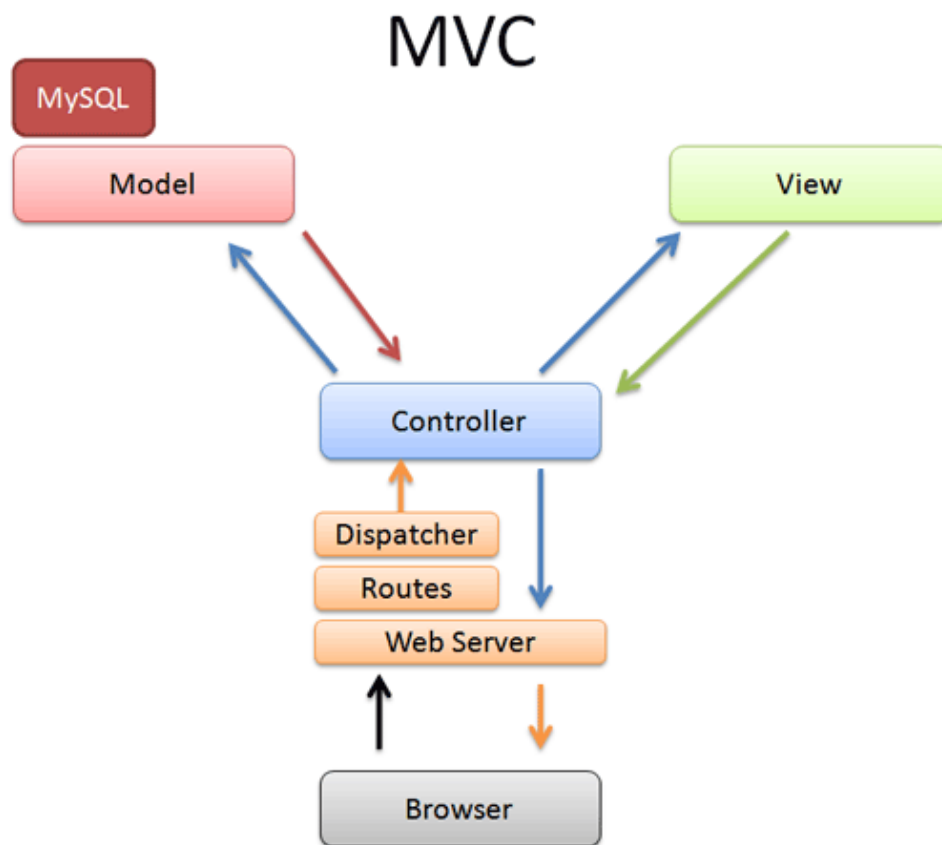
php 第三课 MVC 是什么?

MVC 是什么

名词解释

MVC (**Model-View-Controller**，模型—檢視—控制器模式) 是軟體工程中的一種軟體架構模式。它把軟體系統分為三個基本部分：模型 (Model)，檢視 (View) 和控制(Controller)。

图示



问题

为什么有 MVC?

MVC 不是 php 的不是 c 的不是 java 的,只是一种开发模式!!

一种经过总结的结构!

回忆

我们经过很多年开发后,会遇到各种各样的问题... 比如

多个人协作问题

各个工种的分工问题

代码分离和分工

代码频繁修改

很多重复代码

修改一个东西找不到头绪

.....

等等等

于是

我们总是自己不断的想办法解决这些问题!!

于是,产生了 MVC

MVC 的根本目的,其实为了少写一点代码!

MVC 将不动作用的模块分开, 结构更加清晰!

MVC 也将各个工种分开!

当然, 具体的感受只能是具体的实践!

介绍一个 MVC 开发框架

说这么多也没有用!

我们来简单介绍一个 MVC 框架!

示例

哈哈