

PHP

- PHP 一门后端语言
- 为什么要学习一个后端语言呢？
 - 目前市场上的需求，要求前端人员掌握一个后端语言
 - 方便和后端开发人员进行交互

基本组织架构

- 在讲后端语言之前，我们简单的了解一下我们基本的组织架构
- 我们是一个 **前端开发工程师**
- 还有一个工作叫做 **后端开发工程师**
- 我们一个网站的组织架构基本上由下面的步骤完成
 - **用户 => 前端 => 后端 => 数据库**
- 整个过程中
 - 用户向前端人员要一个网页
 - 前端人员准备一个网页给用户，但是网页中的数据是找后端人员要的
 - 后端人员接受到前端人员要数据以后，去数据库里面找到对应的数据，给到前端人员
 - 前端人员拿到数据以后渲染在页面上
 - 最后把这个页面给到用户看
- 比如： 我们浏览一个新闻网站
 - 用户输入网址
 - 前端人员就要把对应的页面给到用户，在页面打开的过程中，想后端人员索要新闻信息
 - 后端人员接收到前端人员索要新闻信息以后，就去数据库中找到对应的新闻信息数据给前端人员
 - 前端人员接收到后端人员给的新闻信息以后，使用我们的办法吧新闻信息数据渲染在页面上
 - 页面就打开了，用户就可以看到是一个新闻网站了
- 比如： 用户登陆一个网站
 - 当用户书写完表单内容以后，点击提交按钮的时候
 - 前端人员拿到用户填写的内容，把数据整合好传送给后端人员
 - 后端人员接收到数据以后，去数据库中进行比对，看看有没有对应的数据
 - 然后告诉前端人员，你给我的用户名和密码是否正确
 - 前端吧信息反馈给客户看到
 - 如果正确就是跳转页面
 - 如果不正确提示用户名或者密码有问题

服务器的简单理解

- 我们不可能把数据库丢在用户的电脑上
- 一个是太大，一个是不安全
- 所以我们要把数据库放在网络的另一端（远程）
- 所以当前端人员向后端人员索要数据的时候需要网络
- 我们也不可能把页面也放在用户的电脑上
- 所以用户也是需要网络来向前端人员所有页面

服务器的认识

- 我们现在的市场上有一些常见的服务器可以承载内容
- 我们目前比较常用的就是 `Apache` 和 `Tomcat`
- 我们今天要学习的就是 `Apache` 服务器
- 是一个和 `php` 语言合作比较紧密的一个服务器

到底什么是服务器

- 其实说白了，服务器就是一个电脑，当他跑起来一些程序的时候，就变成了一个服务器
- 只不过会跑一些特殊的程序，需要一些特殊的环境，电脑不他一样而已
- 换句话说，我们自己的电脑，跑一些特殊的程序的时候，也可以当作一个小型的服务器来用
 - 只不过计算能力/存储能力/转存能力没有专业的服务器电脑厉害而已

服务器是怎么访问的呢

- 涉及到我们的 `url` 地址中的一些内容了
- 比如我们的经常访问的 `www.baidu.com`
- 这个里面就包含很多的内容，我们看到的这个只是一个省略后的内容
- 是浏览器帮我们省略掉了一些内容
- 全部的地址应该是 `https://www.baidu.com:443`
- 发现多了两个东西 `https://` 和 `:443`
- 其实一个简单的 `url` 地址是由三部分组成的
 - 传输协议
 - 域名
 - 端口号

传输协议

- 我们常见的传输协议是 `http` 和 `https`
- 他们是限制用户和服务器之间交流传输数据的方式和规则
- 也是我们前端和后端人员交互的规则

- 规则
 - 建立连接通道
 - 相互通信
 - 关闭连接通道
- 只不过 `http` 是一种常见协议，不是很安全
- `https` 是一种加密传输协议

域名

- 之前我们说过，服务器就是一个在 **网络那一头** 的一个电脑
- 以前，没有域名的概念，大家都是使用 `IP` 地址来访问
- 也就是 **网络那一头** 的那个电脑的 `IP`
- 都是一堆数字，不方便记忆
- 后来就有了一个 **万维网**，他把每一个 `IP` 地址配套了一个英文的名字
- 方便用户记忆
- 所以说，域名就代表着 **网络那一头** 那个电脑的 `IP` 地址
- 其实也就是我们要访问哪一个服务器

端口号

- 人家的服务器电脑也是有很多的文件夹的
- 不同的文件夹里面存储着不同的内容
- 可能有个 `a` 文件夹，里面存储的是首页
- 可能有个 `b` 文件夹，里面存储的是一些数据
- 所以说，你光找到服务器电脑还不行，还得找对文件夹才可以
- 大家都把文件夹编上号存储了 `0 ~ 255` 一共 256 个
- 每个文件夹里面还有对应的小文件夹 `0 ~ 255` 一个 256 个
- 那么一共就有 $256 * 256$ 个文件夹，也就是从 `0 ~ 65535`
- 所以我们的端口号就有 `65536` 个，分别对应着 `0 ~ 65535`
- 大家都把 `80` 端口号作为一个网站的默认端口号
- `http` 协议默认是 `80` 端口号
- `https` 协议默认是 `443` 端口号

百度

- 到现在，我们在看百度的完整 `url` 地址
- `https://www.baidu.com:443`
 - 以一种加密传输协议来通讯
 - 访问的是网络上 `IP` 的值为 `119.75.217.109` 的那个电脑
 - 访问的是第 `443` 号文件夹
- 这个文件夹里面存储着一个百度的页面
- 我们就能看到这个页面了

PHP 基础语法

- 我们一定要知道，php 是另一个语言了，不再是我们的 `html` / `javascript` 了
- 一个别的语言就有人家语言的规则

php文件

- 我们在写 `javascript` 的时候，是一个 `.js` 文件
- 我们在写 `html` 的时候，是一个 `.html` 文件
- php 的代码写在一个 `.php` 后缀的文件中

php文件的书写

- 所有的 php 代码都要写在一个 php 的范围内
- 要求以 `<?php` 开头
- 要求以 `?>` 结尾

```
1. <?php
2.
3. # php 的代码写在这里
4.
5. ?>
```

简单了解 php 的语法

- 每个语言都会有自己的语法
- 接下来我们就简单了解一下 php 的语法规则
- `php` 里面有一个必须要注意的点 每一个语句后面都要有 `;`

定义变量

- 在 php 中没有 `var` 关键字给我们定义变量
- 直接使用 `$` 来确定一个变量

```
1. <?php
2.
3. # 下面就是一个定义了一个变量，并且赋值为 100
4. $变量名 = 100;
```

```
4.  # 变量名就是 $num
5.  $num = 100;
6.
7.  $boo = true;
8.
9.  # 下面是一个字符串
10. $str = "你好 php";
11.
12. ?>
```

条件语句

- 在 php 中使用条件语句和 js 基本一致

```
1.  <?php
2.
3.  $boo = true;
4.
5.  if ($boo) {
6.      echo '你好, 欢迎光临!';
7.  } else {
8.      echo '您还没有登陆';
9.  }
10.
11. ?>
```

循环语句

- 在 php 中循环语句和 js 基本一致

```
1.  <?php
2.
3.  $num = 5;
4.
5.  for ($i = 0; $i < $num; $i++) {
6.      echo 'hello php';
7.  }
8.
9.  ?>
```

字符串拼接

- 在 php 中, 字符串拼接不再是使用 `+` 进行拼接了, 而是使用 `.` 进行拼接

```
1. $str = 'hello';
2. $str2 = 'world';
3. $str3 = $str . $str2;
4. echo $str3;
5. # 得到的就是 hello world
```

数组

- 在 php 中的数组和 js 中特别不一样

```
1. <?php
2.
3. # 创建一个数组
4. $arr = array(1, 2, 3);
5.
6. print_r($arr);
7. # Array ( [0] => 1 [1] => 2 [2] => 3 )
8. # 这个就类似于我们 js 中的数组，按照索引来的
9.
10. # 创建一个关联数组
11. $arr2 = array('name' => 'Jack', 'age' => 18, 'gender' => '男');
12. print_r($arr2);
13. # Array ( [name] => Jack [age] => 18 [gender] => 男 )
14. # 这个就类似于我们 js 中的 对象，键值对的形式
15.
16. ?>
```

MYSQL

- mysql 是一个数据库的名字
- 和 php 合作的比较好的数据库
- 之前我们说过一个问题，前端向后端索要数据，后端就是去数据库中查询数据，返回给前端
- 接下来就聊聊使用 php 操作数据库

- MySQL 是最流行的关系型数据库管理系统（非关系型数据库简略介绍）
- 关系数据库管理系统(Relational Database Management System)的特点
- 数据以表格的形式出现
- 每行为各种记录名称
- 许多的行和列组成一张表单
- 若干的表单组成database

7. 主键：主键是唯一的。一个数据表中只能包含一个主键。你可以使用主键来查询数据。

数据库的数据类型

1. * 数值类型

2.	
3.	类型 大小 用途
4.	----- ----- -----
5.	TINYINT 1 字节 小整数值
6.	SMALLINT 2 字节 大整数值
7.	MEDIUMINT 3 字节 大整数值
8.	INT或INTEGER 4 字节 大整数值
9.	BIGINT 8 字节 极大整数值
10.	FLOAT 4 字节 单精度 浮点数值
11.	DOUBLE 8 字节 双精度 浮点数值
12.	DECIMAL 小数值

13.

14. * 日期和时间类型

15.	
16.	类型 格式 用途
17.	----- ----- -----
18.	DATE YYYY-MM-DD 日期值
19.	TIME HH:MM:SS 时间值或持续时间
20.	YEAR YYYY 年份值
21.	DATETIME YYYY-MM-DD HH:MM:SS 混合日期和时间值
22.	TIMESTAMP YYYYMMDD HHMMSS 时间戳

23.

24. * 字符串类型

25.	
26.	类型 大小 用途
27.	----- ----- -----
28.	CHAR 0-255字节 定长字符串
29.	VARCHAR 0-65535 字节 变长字符串
30.	TINYBLOB 0-255字节 不超过 255 个字符的二进制字符串
31.	TINYTEXT 0-255字节 短文本字符串
32.	BLOB 0-65 535字节 二进制形式的长文本数据
33.	TEXT 0-65 535字节 长文本数据
34.	MEDIUMBLOB 0-16 777 215字节 二进制形式的中等长度文本数据
35.	MEDIUMTEXT 0-16 777 215字节 中等长度文本数据
36.	LONGBLOB 0-4 294 967 295字节 二进制形式的极大文本数据
37.	LONGTEXT 0-4 294 967 295字节 极大文本数据

操作数据库

- 之前都是对数据库的简单了解，不需要全都背下来
- 我们只要了解一下就可以了
- 接下来我们就是使用 php 链接 mysql 数据库进行数据的增删改查
- 想要操作数据库，除了需要 php 的语法以外，还需要一个 mysql 的 sql 语句
- 使用 php 操作数据库的步骤
 - i. 和数据库建立链接
 - ii. 使用 sql 语句对数据库进行操作
 - iii. 获取结果
 - iv. 和数据库的链接断开

和数据库建立链接

- 在 php 中我们使用 `mysql_connect()` 方法来建立和数据库的链接

1. `<?php`
2. `# 下面就是建立链接，$link 会得到一个链接信息`
3. `# $link = mysql_connect('ip地址', '数据库用户名', '数据库密码');`
4. `?>`
 - 我们有了链接信息 `$link` 以后就可以继续去操作数据库了

确定操作哪个哪个库

- 刚才的是和数据库建立了链接，我们还要确定操作哪个库

1. `<?php`
2. `# 下面就是确定你要操作哪个库`
3. `# mysql_select_db('你要操作的库的名称', $link);`
4. `?>`

执行 sql 语句操作数据库

- 接下来就是使用 sql 语句去这个库里面进行增删改查的操作了

1. `<?php`
2. `# 下面就是使用 sql 语句对数据库进行操作`
3. `# $res = mysql_query('你要执行的 sql 语句');`
4. `?>`
 - 这里有一个注意的点：
 - 我们拿到的结果是一个我们看不懂的处理信息
 - 需要使用 `mysql_fetch_row()` 或 `mysql_fetch_assoc()` 解析一下结果才能看得懂

■ 而使用 `mysql_fetch_row()` 或 `mysql_fetch_assoc()` 函数，我们就不必再处理

关闭链接

- 全部用完以后我们最好是关闭一下数据库链接

```
1. <?php
2.     # mysql_close($conn);
3. ?>
```

完整步骤

- 我们完整的写一下操作的步骤

```
1. <?php
2.     $conn = mysql_connect('localhost', 'root', 'root');
3.     mysql_select_db('test1913');
4.     $res = mysql_query('SELECT * FROM `student`');
5.     $row = mysql_fetch_assoc($res);
6.     mysql_close($conn);
7.
8.     print_r($row);
9. ?>
```

常用的 sql 语句

- 刚才说了怎么操作数据库
- 现在我们学习一下，操作数据库时候常用的 sql 语句
- 我们就是依靠这些 sql 语句来进行数据库操作的

查

- 查询语句

```
1. <?php
2.     # 查询 student 这个表里面的所有数据
3.     $sql = 'SELECT * FROM `student`';
4.
5.     # 查询 student 表中的数据里面 gender 为 男 的数据
6.     $sql = 'SELECT * FROM `student` WHERE `gender`="男"';
7.
8.     # 查询 student 表中的数据里面 age 大于 18 的数据
9.     $sql = 'SELECT * FROM `student` WHERE `age`>18';
```

```
10.
11. # 查询 student 表中的数据里面 age 大于 18 且 gender 为 男 的数据
12. $sql = 'SELECT * FROM `student` WHERE `age`>18 AND `gender`="男";
13.
14. # 查询 student 表中的数据里面 age 小于 22 或者 age 大于 28 的数据
15. $sql = 'SELECT * FROM `student` WHERE `age`<22 OR `age`>28';
16.
17. # 查询 student 表中的数据里面从 第几条开始 查询多少条
18. $sql = 'SELECT * FROM `student` LIMIT 0, 10';
19.
20. # 先按照条件筛选出数据以后再进行分页查询
21. # 下面是查询表中所有 age>18 且 性别为男的所有数据，查出来以后从第 10 条开始查 10 条
22. $sql = 'SELECT * FROM `student` WHERE `age`>18 AND `gender`="男" LIMIT 10, 10';
23.
24. # 查询表的模糊查询
25. # 下面表示查询表中所有数据里面 name 字段中包含 "三" 字的数据
26. $sql = 'SELECT * FROM `student` WHERE `name` LIKE "%三%";
27.
28. # 查询排序，查询的时候按照某一个字段升序或降序排序
29. $sql = 'SELECT * FROM `student` ORDER BY `age` ASC';
30. $sql = 'SELECT * FROM `student` ORDER BY `age` DESC';
31. ?>
```

增

- 增加语句

```
1. <?php
2. # 向表中增加一条数据，再增加的时候主键不能由我们书写，而是 mysql 数据库自己递增
3. $sql = 'INSERT INTO `student` VALUES(null, "张三", 18, "男", 1913, 100)';
4.
5. # 插入固定几个键的数据，其他的用默认值
6. $sql = 'INSERT INTO `student` (`name`, `age`) VALUES("李四", 22)';
7. ?>
```

删

- 删除语句

```
1. <?php
2. # 删除表中 id 为 100 的数据
```

```
3. $sql = 'DELETE FROM `student` WHERE `id`=100';
4.
5. # 删除表中 name 为 张三 的数据
6. $sql = 'DELETE FROM `student` WHERE `name`="张三"'
7. ?>
```

改

- 修改语句

```
1. <?php
2. # 更新一条 id 为 100 的数据中的 name 字段的值和 age 字段的值
3. $sql = 'UPDATE `student` SET `name`="张三", `age`=10 WHERE `id`=100'
4.
5. # 更新数据的时候让所有的数据增加一些内容
6. $sql = 'UPDATE `student` SET `age`=age+1'
7. ?>
```