目录

1：网页生成原理 1

2：服务器安装、环境配置 7

3：分界标识符 7

4：注释、变量、数据类型、运算符、流程控制 8

5：函数 9

6：数组、类与对象 9

应用： 11

7：页面输出 11

8：生成动态页面 11

9：PHP解析JSON数据 11

10：使用JSON生成动态页面 11

应用： 12

综合应用： 12

## 1：网页生成原理

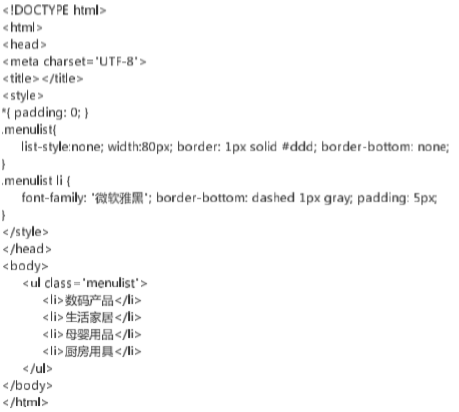
先说一说PHP，PHP是一门后台语言,标准的变量、数据类型、运算、选择结构、循环结构、函数、数组;跟大多数的编程语言是一样的。唯一的差别，基本都在一些语法上。

PHP作为一个动态网页编程语言,它不仅能编写后台程序(phtml)。 同时也具备编写动态网页的能力,跟JSP（java ）一样,这是它们想当年流行起来的最重要的原因,没有之一。 那什么叫动态网页呢?

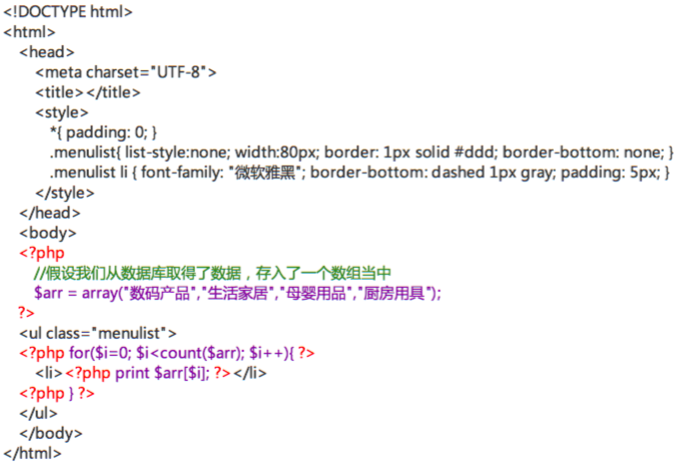
在很久很久以前,前后端不分离的年代,那个时候还没有所谓的前端工程师。 服务器如何将它的数据以网页的形式展示给用户呢? 例如我们的页面上需要展示四个商品类别 如图：



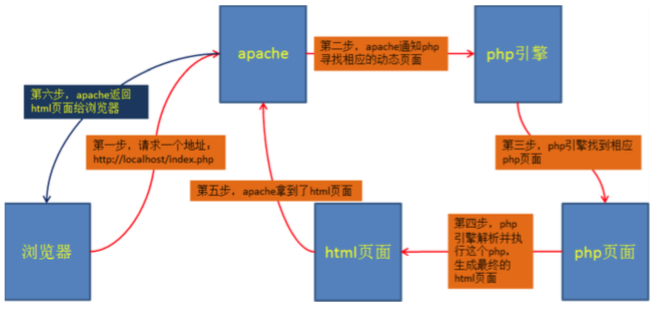
但这些类别信息,是服务器从数据库查询出来的。最终服务器返回给浏览器的静态网页应该是这样子的：



php写出来是这个样子：



仔细观察,这里所有紫色的部分,全部为php的代码。而这些代码都使用了<?php ...... ?>包裹起来。 最重要的是,PHP的代码跟HTML的代码混合了,它的大概原理如图所示:



这样一来,后台的开发人员,就可以像写后台程序一样的写网页,岂不是很方便。 曾经,这种巧妙的解决方案,流行了很多年。但是随着行业的发展,随着WEB2.0时代的到来。

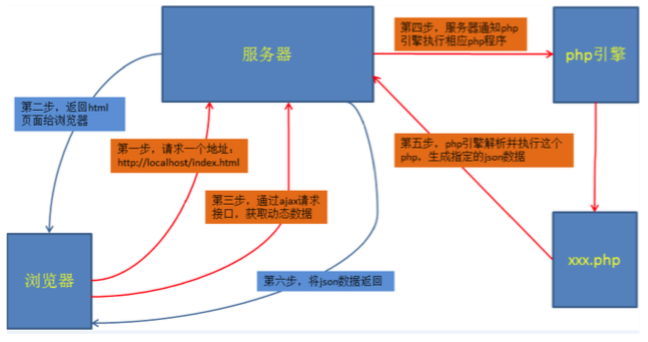
随着网页交互的场景越来越多,随着用户体验的重要性逐步提升,网页上的工作量变得越来越大了。 因为页面不仅要求越来越美观,同时页面上的各种交互特效也越来越丰富(JS或Flash);

可是你再看看PHP这样的动态网页,对于一个特效要求极高的网站,它可维护性好吗? 它便于网页重构吗? 它代码可读性强吗? NO! 这些优点全都没有。

于是那几年开始出现了一个叫做RIA的名词: 丰富的互联网应用程序。

传统的动态网页已经很难实现RIA了, 或者说实现它的成本非常高。那么它需要借助Flash的力量,或者JS的力量

于是人们开始慢慢的将前后端进行分离开来。 后台语言只负责后台程序接口的编写。而一部分人只负责页面以及交互特效的完成。而网页需要的动态数据,可以通过ajax来获取。 这样一来,一个负责编写页面的人,即使他不懂后台语言,也能完成复杂的前端工作。因为网页上再也不会出现后台代码了。 大致的原理如下:



**有一个结论大家必须要搞清楚,PHP是后台语言,只能在特定环境里运行,浏览器是执行不了的。**好了,了解原理以后,我们看一看它的语法。

## 2：服务器安装、环境配置

lamp：linux(操作系统)+mysql（数据库）+apache（服务器）+php（语言环境）。 wamp- xampp

linux：一套免费使用和自由传播的操作系统。

windows：美国微软公司开发的操作系统。

macOS：美国苹果公司开发的完备而独立的操作系统。

apache：世界使用排名第一的web服务器软件。由于其跨平台和安全性被广泛使用。

nginx：是一个高性能的HTTP和反向代理的服务器。

mysql：一款关系型数据库管理系统，目前属于Oracle旗下产品。

php：PHP（“PHP: Hypertext Preprocessor”，超文本预处理器的字母缩写）是一种被广泛应用的开放源代码的多用途脚本语言，它可嵌入到 HTML中，尤其适合 web 开发。

使用集成环境wamp，详细讲解该软件的使用方法。

如果localhsot 打不开，把httpd.conf 文件 复制到 C:\wamp\bin\apache\Apache2.4.4\conf 这个目录里面，替换原来的即可。

## 3：分界标识符

PHP标记：当解析一个文件时，PHP = 寻找起始和结束标记，也就是 <?php 和 ?>，这告诉 PHP 开始和停止解析二者之间的代码。此种解析方式使得 PHP 可以被嵌入到各种不同的文档中去，而任何起始和结束标记之外的部分都会被 PHP 解析器忽略。

指令分隔符：PHP 需要在每个语句后用分号结束指令。

## 4：注释、变量、数据类型、运算符、流程控制

**注释：**

单行注释：//、#

多行注释：/\*…\*/。

**变量：**PHP中的变量用一个美元符号后面跟变量名来表示。变量名是区分大小写的。变量名规则：字母或者下划线开头，后面跟上任意数量的字母、数字、或者下划线。

**来自PHP外部的变量：**

读取表单提交过来的数据，我们可以通过$\_GET、$\_POST、$\_REQUEST。

读取浏览器发送过来的cookie，可以通过$\_COOKIE。

**数据类型：**

PHP支持8种原始数据类型。

四种标量类型：boolean（布尔型）、integer（整型）、float（浮点型）、string（字符串型）

两种复合类型：array（数组）、object（对象）

两种特殊类型：resource（资源）、NULL（无类型）

**运算符：**

算术运算符：＋（加）、－（减）、＊（乘）、／（除）、％（取余）、++（自增）、--（自减）。

比较运算符：==、===、!=、!==、<、>、<=、>=。

逻辑运算符：!、&&、||

字符串运算符：.

赋值运算符：＝

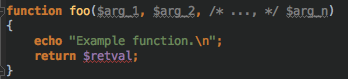
**流程控制：**

if、switch、while、do-while、for、break、continue。

**包含文件**：include、include\_once、require、require\_once

## 5：函数

用户自定义函数：

****

内置函数：print\_r、var\_dump等。

## 6：数组、类与对象

**数组：**

操作数组：用方括号的语法新建/修改。

添加新单元：可以通过在方括号内指定键名来给数组赋值实现，也可以省略键名，在这种情况下给变量名加上一对空的方括号([])。

修改单元：通过其键名给该单元赋一个新值。

删除单元：对其调用unset()函数。

遍历数组：foreach。

**了解类与对象的关系：**

在面向对象编程中，类和对象是最基本、最重要的组成单元。类实际上是表示一个客观世界某类群体的一些基本特征抽象。对象就是表示一个个具体的东西，如同现实生活中的马，马表示的是一个类。因为马本身属于一个广义的概念，不是一匹匹具体的马，而某一匹具体的马就可以称为对象。

对象是以类为模版创建的，或者说对象是以类为蓝图实例化的。

举个例子来说，在奔驰工厂里，要想生产出汽车，工程师们首先要设计出一个汽车的图纸，然后按照设计图规定的结构去生产汽车。这样生产出来的汽车结构和功能都是一样的，但是每一辆汽车都有自己的特征，如颜色、内饰等都会存在一些差异。在这个例子中，设计图纸就是一个类，它规定了汽车应该有的一些部件，如发动机、车轮和车架等信息的定义。而根据这个模型生产出来的一辆辆奔驰车就是一个个对象，它们可以被用户使用。

**创建类**：每个类的定义都以关键字 class 开头，后面跟着类名，后面跟着一对花括号，里面包含有类的属性与方法的定义。

创建对象：就是类的实例化。创建一个实例：必须使用new关键字。

**类的属性和方法访问控制：**

对属性或方法的访问控制，是通过在前面添加关键字 public（公有），protected（受保护）或 private（私有）来实现的。被定义为公有的类成员可以在任何地方被访问。被定义为受保护的类成员则可以被其自身以及其子类和父类访问。被定义为私有的类成员则只能被其定义所在的类访问。

属性的访问控制：类属性必须定义为公有，受保护，私有之一。

方法的访问控制：类中的方法可以被定义为公有，私有或受保护。如果没有设置这些关键字，则该方法默认为公有。

## 应用：

1：掌握基本语法。

2：掌握网页生成原理。

3：掌握对象概念。

## 7：页面输出

## 8：生成动态页面

## 9：PHP解析JSON数据

使用foreach语法结构。

## 10：使用JSON生成动态页面

在页面中使用foreach，遍历JSON数据显示到界面上。

当然很多时候,我们的接口并不会返回一个标准的html页面,而是一个json格式的字符串 这主要是为AJAX准备的,对于JS来讲,处理JSON格式的数据是最方便的。

## 应用：

1：将JSON数据展示到界面。

## 综合应用：

1：掌握PHP语法。

2：掌握动态网页的编写。