学习方法

1. 建立逻辑思维
   1. 多分析
   2. 理解逻辑算法
   3. 多练、多敲、多读
2. 编程习惯
3. 大胆动手
4. 保持好奇心
5. 乐观面对错误

就业能力

1. 分析能力
2. 独立思考能力
3. 程序阅读能力
4. 程序排错能力
5. 独立编写程序能力
6. 团队协作能力
7. 学习能力
8. 善于总结的能力

专业技能

PHP

1. 服务器安装
   1. PHP（外文名:PHP: Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。语法吸收了C语言、Java和Perl的特点，利于学习，使用广泛，主要适用于Web开发领域。
2. 环境配置
3. 分界标示符
   1. PHP分界标示符确定PHP脚本的开始和结束位置，在PHP语言中有四种不同风格的分界标示符可以让PHP代码嵌入到HTML代码中，分别如下 ：
      1. PHP标准分界符：<?php ?>
      2. PHP简写分界符：<? ?>
      3. <% %>
4. 标识符/关键字/数据类型
   1. 标识符：由字符，数字，下划线组成，首字母必须是字符或下划线。
      1. 变量以$开头。 $score = 10;
      2. PHP是区分大小写的
      3. 语句以分号结束
   2. 数据类型
      1. 字符串(string):字符串是字符序列，比如：”hello wrold”.  
         字符串可以是引号内的任何文本，可以使用单引号或双引号。
      2. 整数(integer)：整数是没有小数的数字。
         1. 整数规则：
            1. 整数必须有至少一个数字(0-9)
            2. 整数不能包含逗号或空格
            3. 整数不能有小数点
            4. 整数正负均可
            5. 可以用一种格式规定整数：十进制、十六进制（前缀是0x）或八进制（前缀是0）
         2. 浮点数(double):是有小数点或指数形式的数字。
         3. 布尔(boolean)：是true或false
            1. 常用于条件测试。
         4. 数组(array)：在一个变量中存储多个值。
         5. 对象(object)：是存储数据和有关如何处理数据的信息的数据类型。
         6. NULL：特殊的NULL值表示变量无值。
            1. NULL值标示变量是否为空。也用于区分空字符串与空值数据库。

定义文件头，防止出现乱码：

//定义文件头

header("content-type:text/html;charset=utf-8");

1. 函数
   1. PHP的真正力量来自它的函数：它拥有超过1000个内建的函数。
   2. PHP用户定义函数
      1. 除了内建的PHP函数，我们可以创建我们自己的函数。
      2. 函数是可以在程序中重复使用的语句块。
      3. 页面加载时函数不会立即执行。
      4. 函数只有在被调用时才会执行。
   3. 在PHP创建用户定义函数
      1. 用户定义的函数声明以function开头：
      2. 语法  
         function fnName(){  
          被执行的代码;  
         }  
         //函数名能够以字母或下划线开头（而非数字）。  
         //函数名对大小写不敏感。
   4. PHP函数参数
      1. 可以通过参数向函数传递信息。
      2. 参数被定义在函数名之后，括号内部。可以添加任意多参数，只要用逗号隔开即可。
   5. PHP函数的返回值
      1. 如需使函数返回值，使用return 语句
2. 数组/对象
   1. 数组：能够在单独的变量名中存储一个或多个值，并且能够通过引用下标号来访问某个值。
   2. 在PHP中创建数组
      1. 在PHP中，array()函数用于创建数组：  
         array();
      2. 在PHP中，有三种数组类型：
         1. 索引数组：带有数字索引的数组
         2. 关联数组：带有指定键的数组
         3. 多维数组：数组嵌套数组
   3. 索引数组
      1. 索引是自动分配的（索引从0开始）：  
         $cars = array(“Volvo”,”BMW”,”BYD”);
      2. 手动分配索引：  
         $cars[0] = “Volvo”;  
         $cars[1] = “BMW”;  
         $cars[2] = “BYD”;
      3. 获得数组的长度：count()函数
         1. Count()函数用于返回数组的长度（元素数）：  
            <?php  
            $cars = array(“Volvo”,”BMW”,”BYD”);  
            echo count($cars);  
            ?>
   4. 关联数组：使用分配给数组的指定键的数组。
      1. $age = array(“zhangsan” => “18”,”lisi” => “19”,”wangwu => “20”);
      2. $age [“zhangsan”] = “18”;  
         $age[“lisi”] = “19”;  
         $age[“wangwu”] = “20”;
      3. 数组的打印 print\_r(数组名)
3. 页面输出
   1. 使用print
      1. 只能输出一个字符串，并始终返回1
         1. <?php  
            print “<h2>PHP is fun!</h2>”;  
            print “Hello world!<br/>”;  
            print “I’m about to learn PHP!”;  
            ?>
   2. 使用echo
      1. 能够输出一个以上的字符串
         1. <?php  
            echo “<h2>PHP is fun!</h2>”;  
            echo “Hello world!<br/>”;  
            echo “This”, “ string”,” was”,” made”,” with multiple parameters.”;  
            ?>
   3. 注：echo比print稍快，因为它不返回任何值。
4. 生成动态页面
   1. PHP作为一个动态网页编程语言，它不仅能编写后台程序。同时也具备编写动态网页的能力，跟JSP一样，这是它们想当年流行起来的最重要的原因，没有之一。
   2. 什么叫动态网页呢？
      1. 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？
         1. 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：  
            <!DOCTYPE html>  
            <html>  
            <head>  
            <meta charset="UTF-8">  
            <style type="text/css">  
             \*{padding:0;}  
             .menulist{  
             list-style:none;width:80px;border:1px solid #ddd;border-bottom:none;  
             }  
             .menulist li{  
             font-family:"微软雅黑";border-bottom:dashed 1px gray;padding:5px;  
             }  
            </style>  
            </head>  
            <body>  
             <ul class="menulist">  
             <li>数码产品</li>  
             <li>生活家居</li>  
             <li>母婴用品</li>  
             <li>厨房用具</li>  
             </ul>  
            </body>  
            </html>
      2. PHP写出来是这个样子：  
         <!DOCTYPE html>  
         <html>  
         <head>  
         <meta charset="UTF-8">  
         <style type="text/css">  
          \*{padding:0;}  
          .menulist{  
          list-style:none;width:80px;border:1px solid #ddd;border-bottom:none;  
          }  
          .menulist li{  
          font-family:"微软雅黑";border-bottom:dashed 1px gray;padding:5px;  
          }  
         </style>  
         </head>  
         <body>  
         <?php

//假设我们从数据库取得了数据，存入了一个数组当中

$arr = array("数码产品","生活家居","母婴用品","厨房用具");

?>

<ul class="menulist">

<?php

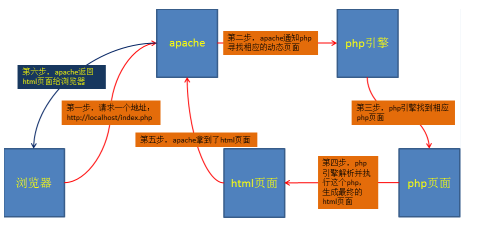
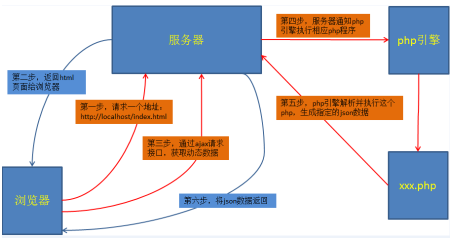
for($i = 0; $i < count($arr);$i ++){

?>

<li><?php print $arr[$i];?></li>

<?php }?>

</ul>  
 </body>  
 </html>

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。  
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。

注：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。

1. 解析JSON数据
   1. 现在JSON格式在web开发中越来越受重视，特别是在使用ajax开发项目的过程中，经常需要将json格式的字符串返回到前端，前端解析成JS对象（JSON）
   2. Eval()方法：解析JSON数据的最常用方法是使用javascript的eval()方法，代码如下：  
      function toJson(str){  
       var json = eval(‘(‘ + str + ‘)’);  
       return json;  
      }  
      该方法存在性能和安全方面的问题，不建议使用。
   3. JSON.parse()方法  
      这种方法只支持IE8/Firefox3.5+/Chrome4/Safari4/Opera10以上版本，这些浏览器都已经接近W3C标准，默认实现了toJSON方法。代码如下：  
      function toJson(str){  
       return JSON.parse(str);  
      }
2. 使用JSON生成动态页面  
   <body>  
    <p>姓名：<span></span></p>  
    <p>年龄：<span></span></p>  
    <p>身高：<span></span></p>  
    <p>电话: <span></span></p>  
   </body>  
   <script type="text/javascript">  
    window.onload = function(){  
    var json = {"name":"zhangsan",  
    "age": 18,  
    "height":175,  
    "phone" : 13111111111  
    };  
    var oSpan = document.getElementsByTagName("span");  
    oSpan[0].innerHTML = json.name;  
    oSpan[1].innerHTML = json.age;  
    oSpan[2].innerHTML = json.height;  
    oSpan[3].innerHTML = json.phone;  
    };  
   </script>

应用： php接收表单数据