web技术

# 计算机基础知识

## 计算机体系结构

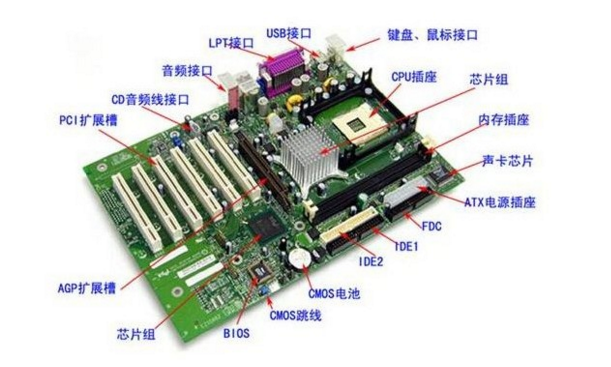
**运算器：**计算机中执行各种算术和逻辑运算操作的部件。运算器的基本操作包括加、减、乘、除四则运算，与、或、非、异或等逻辑操作，以及移位、比较和传送等操作，亦称算术逻辑部件。

**控制器：**是指按照预定顺序改变主电路或控制电路的接线和改变电路中电阻值来控制电动机的启动、调速、制动和反向的主令装置。由程序计数器、指令寄存器、指令译码器、时序产生器和操作控制器组成，它是发布命令的“决策机构”，即完成协调和指挥整个计算机系统的操作。

**存储器：**是计算机系统中的记忆设备，用来存放程序和数据。

**I/O设备：**输入和输出设备

## 计算机主板结构



## 进制转化

计算机都是2进制（0、1）

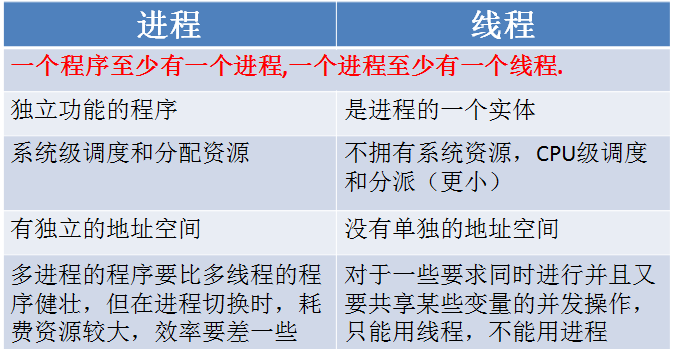
学会进制转化

技巧：8421（杀手锏）

大学：除2、8、16再转（效率太慢，容易出错）

现实：在线转化工具

## 进程/线程



### 多线程

多线程：软件或者硬件上实现多个线程并发执行的技术。

同一时间执行多于一个线程，进而提升整体处理性能。

### 线程同步

同步：就是在发出一个功能调用时，在没有得到结果之前，该调用就不返回，同时其它线程也不能调用这个方法

* 1. A执行到一定程度时要依靠B的某个结果，于是停下来，示意B运行；
  2. B按照代码逻辑依次执行后，将结果给A；
  3. A再继续操作。

### 死锁

死锁：

* 1. A执行到一定程度时要依靠B的某个结果，于是停下来，示意B运行；
  2. B按照代码逻辑依次执行中需要A的某个结果，示意A运行，等到A运行后再运行B后面的代码，并将结果给A；
  3. A必须等到B的结果才能继续执行

于是问题出来了：A执行不下去，因为没有B的结果；B执行不下去，因为没有A的返回结果，于是死锁

## 软件环境：开发环境、测试环境、生产环境

开发环境：

1. 每个开发自己的PC就是一个独立的开发环境；
2. 开发好自己模块后上传到对应服务器组成一个更大部件或系统

测试环境：

1. 一般提供独立一台服务器作为测试环境
2. 有的公司提供2台，一台应用服务器，一台数据库服务器
3. 有的公司测试环境相当麻烦和相当庞大

生产环境：

1. 产生生产力的环境，公司也叫现场环境或客户环境
2. 测试环境测得没问题后往往升级给生产环境后会出一堆问题

## 软件结构

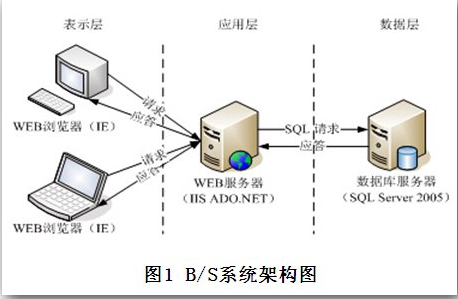
### C/S

* C/S （Client/Server）结构，即大家熟知的客户机和服务器结构。通过这种架构可以充分利用两端硬件环境的优势，将任务合理分配到Client端和Server端来实现，降低了系统的通讯开销。
* 实际应用：QQ、网游、手机App、火车票窗口购票系统、迅雷播放器

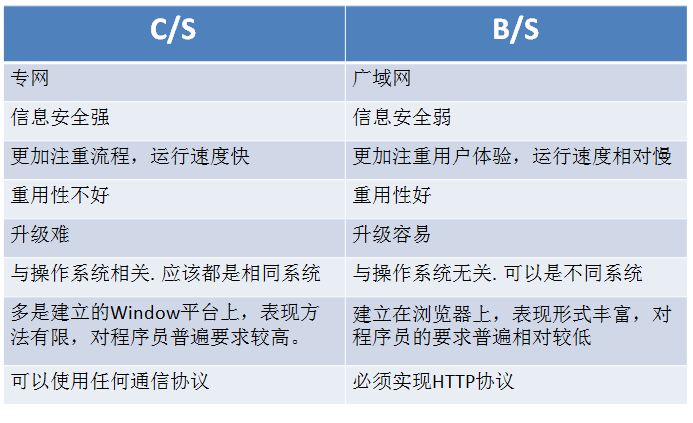


### B/S

* B/S结构（Browser/Server），是WEB兴起后的一种网络结构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。
* 淘宝、百度、12306购票网站、www.cctv.com、爱奇艺、页游



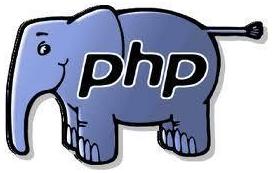
### B/S和C/S的不同



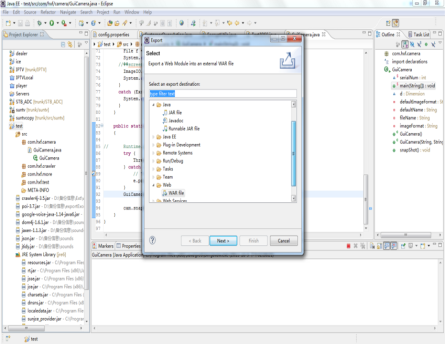
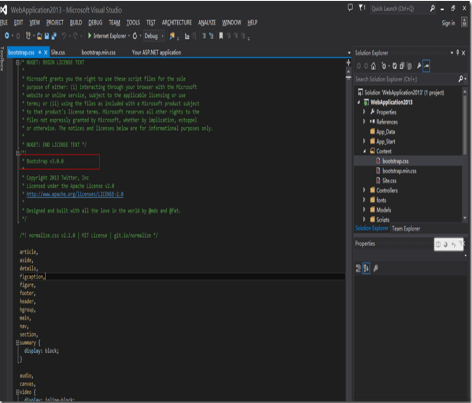
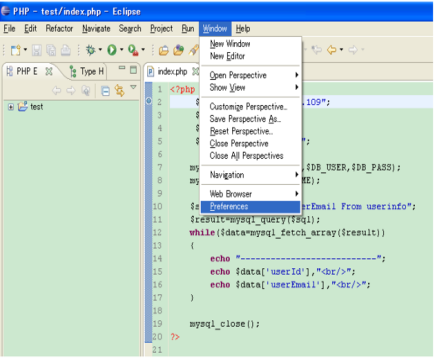
### B/S和C/S的误区

* C/S和B/S并没有本质的区别：
* B/S属于C/S，浏览器只是特殊的客户端；
* B/S和C/S各有千秋，谁也无法取代谁！

## 常见开发语言：C/C++、Java、C#、.NET、ASP、JSP、PHP



## 软件IDE：Eclipse、EPP，VS

## WEB服务器：Tomcat、Apache、IIS 、JBoss、 Weblogic

### Tomcat

* Tomcat是Apache 软件基金会的Jakarta项目中的一个核心项目，由Apache、Sun 和其他一些公司及个人共同开发而成。
* **免费的开放源代码的轻量级Web 应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。**
* **Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。**
* Tomcat 部分是Apache 服务器的扩展，但它是独立运行的，所以当你运行tomcat 时，它实际上作为一个与Apache 独立的进程单独运行。
* **默认端口：8080**

目录结构：

* /bin：存放启动和关闭的脚本文件
* /conf：存放各种全局配置文件，其中包括server.xml、tomcat-users.xml和web.xml等
* /lib：存放所需的所有jar文件
* /logs：存放Tomcat执行时的日志文件
* /temp：存放运行时所产生的临时文件
* /webapps：Web发布目录，默认情况下把Web应用文件放于此目录
* /work：JSP生成的Servlet源文件和字节码文件放到这个目录下

### Apache

* Apache是世界使用排名第一的Web服务器。
* 它可以运行在几乎所有的计算机平台上，由于其跨平台和安全性被广泛使用，是最流行的Web服务器端软件之一。ac
* 它快速、可靠并且可通过简单的API扩充，将Perl/Python等解释器编译到服务器中。
* **默认端口：80**

目录结构：

* /bin：存放启动和关闭的脚本文件
* /conf ：存放各种全局配置文件
* /htdocs ：编译安装时站点目录
* /logs ：默认日志文件存放目录。包括错误日志(error\_log)和访问日志（access\_log）
* /modules ：模块目录，apache的各种扩展so

### IIS：

* IIS是由**微软公司**提 供的基于**运行Microsoft Windows**的**互联网**基本服务。
* IIS还包含了：Gopher Server、TP Server。
* 默认端口：**80**
* **和apache的端口一致，所以不能同时启动IIS和apache，这点在工作中要非常注意。如果要同时启动，只能修改其中一个的默认端口**

### JBoss

是一个基于J2EE的开放源代码的应用服务器。 JBoss代码遵循LGPL许可，可以在任何商业应用中免费使用，而不用支付费用。JBoss是一个管理EJB的容器和服务器，支持EJB 1.1、EJB 2.0和EJB3的规范。但JBoss核心服务不包括支持servlet/JSP的WEB容器，一般与Tomcat或Jetty绑定使用。

### Weblogic

WebLogic是用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。将Java的动态功能和Java Enterprise标准的安全性引入大型网络应用的开发、**集成、部署和管理**之中。

## 数据库：Oracle 、MySQL、MSSQL

### MySQL优缺点



### MSSQL优缺点



### Oracle优缺点



### 更简单一点理解

* MySQL：功能强，使用简单，管理方便，运行速度快，可靠性高，安全保密等特点。
* MSSQL：只支持微软平台，数据量不及上两者，可用性最好，但是性能不及上两者，适用于中型、小型企业及商业应用。
* Oracle ：庞大，功能强大，支持产品范围广，支持平台多，支持数据量最大，适用于大型及超大型应用

# 基本概念

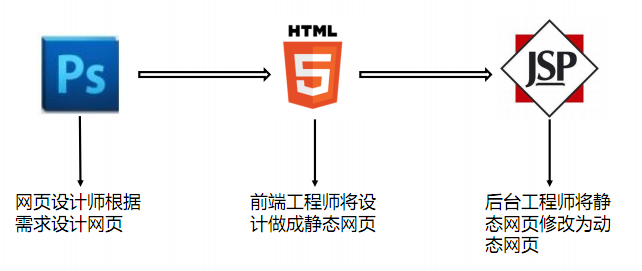
1、前端/后端

前端：客户展示层，也是用户最直观的一层

后端：系统逻辑管理层，提高对前端功能、数据、逻辑业务的各种管理

（注意：不是所有后端功能都有界面）

2、软件基本研发过程

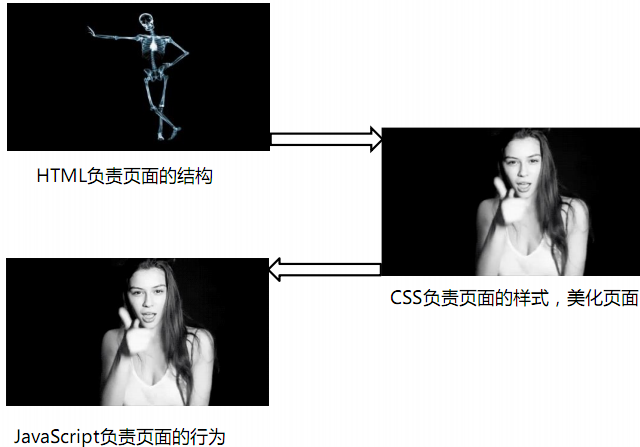


网页设计师：他们往往是由UI（美工）设计，

研发工程师：实现从UI到程序

|  |
| --- |
| 作为测试，我们需要：   1. 了解前端，才能更好的和UI沟通 2. 了解后端，才能更好和程序员沟通 3. 误区：前端好学   以前纯html或者div+css时还不算难，现在要求响应式布局，还要适应各种移动设备（屏幕大小不一）等，同时，很多公司也要求前端要会JQuery，Ajax等，所以越来越难   1. 现在企业的要求是   开发：“全栈”  测试：“全能” |

根据WSC标准，一个网页是由结构（html）、表现（css）、行为（JavaScript）构成



W3C：万维网联盟（World Wwid Web Consortium），专门为了定义网页相关的标准而成立。定义了html、css、dom、http、xml等标准

http://www.w3school.com.cn/

# 静态

## 认识html

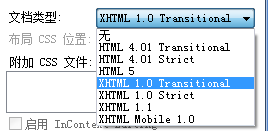
html：（Hypertext Markup Language）超文本标记语言,其基本机构：

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>无标题文档</title>  </head>  <body>  </body>  </html> |

|  |  |
| --- | --- |
| F:\教材开发\01 资料素材\图片\书籍中使用的图片\提示.jpg | 1. HTML标签成对出现，如<head></head> 2. 所有要展示内容都写在body之间：<body>内容</body> 3. 在标签之间写具体内容 4. 可以使用<标签 /> 5. 标签内可以套标签   <h1>欢迎来到<b>门道</b></h1>  **注意：b是和h1的子标签** |

### doctype

|  |
| --- |
| **<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"**  **"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>无标题文档</title>  </head>  <body>  </body>  </html> |



分别为：过渡版、严格版、mobile版

默认采用html4.0.1的过渡版

### 中文乱码问题

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; **charset=utf-8**" />  <title>无标题文档</title>  </head>  <body>  </body>  </html> |

|  |  |
| --- | --- |
| F:\教材开发\01 资料素材\图片\书籍中使用的图片\提示.jpg | 1. 乱码问题是所有开发头痛的问题 2. **注释三码合一，否则绝对乱码** 3. **国际：utf-8（最最常用，以后你工作的系统几乎全是这个编码）**   **简体中文:gb2312、gbk、gb18030**  **繁体中文：big5** |

### meta

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  **<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**  <title>无标题文档</title>  </head>  <body>  </body>  </html> |

|  |  |
| --- | --- |
| F:\教材开发\01 资料素材\图片\书籍中使用的图片\提示.jpg | 1. 其中keywords和description最重要，是SEO优化时一个非常重要地方 2. 示例：http://www.smysmjj.com |

### html、head、title、body

html：所有标签的父标签，旗下2个标签：head、body

head：头标签，只能有一个

title：用于显示网页标题

body：主体标签，页面中所有能看到的都写在body中

### 元素（标签）

一个完整的标签就称为一个元素，你也可以简单理解成：**元素=标签**

**例如：**

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>无标题文档</title>

</head>

我们一般叫：head标签、meta标签、head下的title标签

### 属性

可以为标签设置属性，从而提供标签附加信息

|  |
| --- |
| <h1 title="门道">欢迎来到门道科技</h1> |

|  |  |
| --- | --- |
| F:\教材开发\01 资料素材\图片\书籍中使用的图片\提示.jpg | 1. 需要设置在开始标签和自结束标签中 2. **以"名称=值"存在** 3. **了解一个元素后，就要去了解他对应的属性** |

**对于测试，必须掌握的2个属性：**

**id：如人的身份证，不可以同名，只能被调一次**

**name：如人的姓名，一般建议不同名，一般通过名字拿到这个标签的其他属性**

**class：如人的衣服，可以同名，可以被调N次，**

|  |  |
| --- | --- |
| F:\教材开发\01 资料素材\图片\书籍中使用的图片\提示.jpg | 对于测试学习id、name、class的意义：   1. **自动化识别控件时，很大一部分就通过id或class去识别，其中id最常见**   **很多开发不喜欢对控件写id和name，导致识别控件困难**   1. **编写css时，id用”#”,class用“.”来设置标签样式** |

### 注释

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>无标题文档</title>  </head>  <body>  <!-- 这是注释部分 -->  门道科技集团  </body>  </html> |

|  |  |
| --- | --- |
| F:\教材开发\01 资料素材\图片\书籍中使用的图片\提示.jpg | 1. 注释不会显示在网页中，目的：方便维护 2. **注释不能嵌套** |

### 表单

及其下属各控件：

<form>

### 表格

及其下属各控件：

<table>

### 其他

### html的不足

我们有了骨架，没有肌肉

人：都是由206块骨头构成，但是却衍生出了不同形态的人，而这个形态的控制，就靠css

## 认识css

了解csszengarden(禅意花园)，同样的一套“骨架”，全世界工程师们设计出了丰富多彩的网页

http://www.csszengarden.com/

http://www.mezzoblue.com/zengarden/alldesigns/

### css的三种引入方式

1、行内引入

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>无标题文档</title>  </head>  <body>  <p **style="color:#F00; "**></p>  </body>  </html> |

优点：最直接最简单的一种，直接对HTML标签使用style=""

缺点：HTML页面不纯净，文件体积大，不利于SEO，后期维护也不特别方便

2、内嵌引入

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>无标题文档</title>  **<style type="text/css">**  **body,div,a,img,p{margin:0; padding:0;}**  **</style>**  </head>  <body>  <p **style="color:#F00; "**></p>  </body>  </html> |

优点：作用于单页，如果整站页面不多，CSS代码也不多，这种样式还是很不错的

缺点：每个页面都要定义的，如果一个网站有很多页面，每个文件都会变大，后期维护难度也相对大了一些

3、链接引入

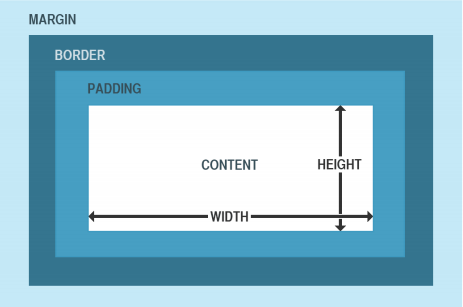
|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>无标题文档</title>  **<style type="text/css">**  **body,div,a,img,p{margin:0; padding:0;}**  **</style>**  **<link type="text/css" rel="stylesheet" href="style.css" />**  </head>  <body>  <p **style="color:#F00; "**></p>  </body>  </html> |

特点：使用频率最高，最实用的样式，实现了html和css的分离，使得前期制作和后期维护都十分方便

4、导入样式（不推荐）

|  |  |
| --- | --- |
| F:\教材开发\01 资料素材\图片\书籍中使用的图片\提示.jpg | 1. **就近原则** 2. **理论上：行内>内嵌>链接>导入** 3. **实际上：内嵌、链接、导入在同一个文件头部，谁离相应的代码近，谁的优先级高** |

### 盒子模型



## div+css与table+css的优缺点

作为一个Web 设计师，你是否好意思承认自己的代码中使用了 Table？如果是，你是一个有勇气的人！！！

因为：兼容性不好！！！臃肿！！！ 不利于SEO！！！页面渲染慢！！！

div+css在网页设计中有优缺点

优点：

1. 网页载入比较快，由于div+css中的css富含丰富的样式，定位和表现更加灵活。
2. 页面内容与样式分离可以是网页代码减少，使页面的样式的调整变得更加方便。
3. 表现和结构分离，在团队开发中更容易分工合作而减少相互关联性。

缺点：

1. 还没有实现所有浏览器的统一兼容，使得设计变得更复杂。
2. 比table定位复杂得多，多层嵌套的div会严重影响网页代码的可阅读性，对于网页设计高手也很容易出现问题
3. css通常放在一个或几个外部文件中，有可能相当复杂和比较庞大，如果css文件调用异常，那么整个网页将变得惨不忍睹。

## 响应式布局之bootstrap

1. Ethan Marcotte在2010年5月份提出
2. 一个网站能够兼容多个终端，而以前是每个终端一个版本
3. 目的：核心为是为了解决移动互联网浏览而诞生的。

## 实战：学会修改页面

## 作业：完成规定布局修改和了解框架

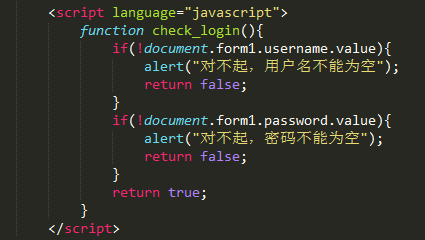
# 动态

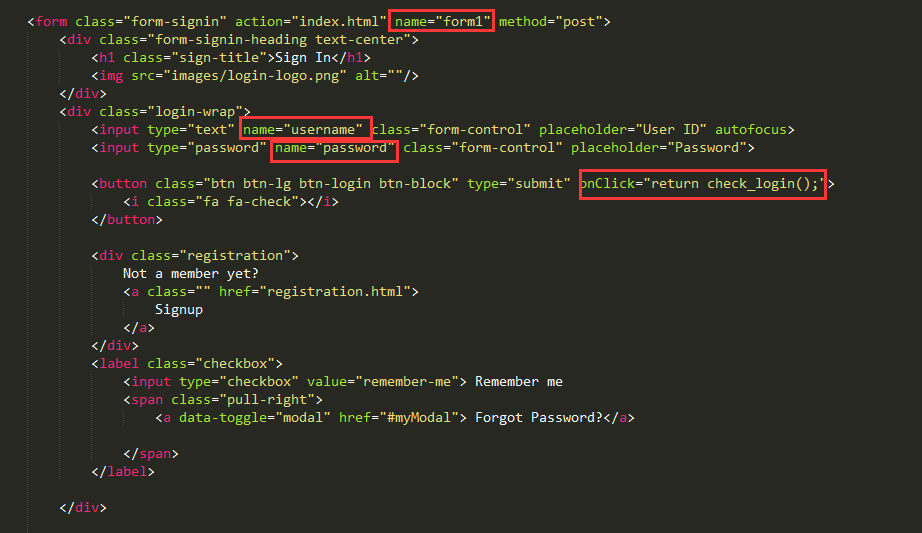
## 静态网站的不足

页面还是“死”页面，没有任何的“用户交互”

* 1. 用户输入错误信息没有提示
  2. 用户在提交前或后给提示
  3. 信息没有通过读取数据库获得，而且直接写在了页面了，无法动态修改或删除信息

## js验证





## JQuery和Ajax(了解)

jQuery是一个快速、简洁的JavaScript框架，是继Prototype之后又一个优秀的**JavaScript代码库（或JavaScript框架）**。jQuery设计的宗旨是“write Less，Do More”，即倡导写更少的代码，做更多的事情。**它封装JavaScript常用的功能代码，提供一种简便的JavaScript设计模式**，优化HTML文档操作、事件处理、动画设计和Ajax交互。

jQuery的核心特性：

1. 具有独特的链式语法和短小清晰的多功能接口；
2. 具有高效灵活的css选择器，并且可对CSS选择器进行扩展；
3. 拥有便捷的插件扩展机制和丰富的插件
4. 兼容各种主流浏览器，如IE 6.0+、FF 1.5+、Safari 2.0+、Opera 9.0+等。[1]

Ajax 即“Asynchronous Javascript And XML”（**异步 JavaScript 和 XML）**

Ajax = **异步** JavaScript 和 XML。

Ajax 的核心特性：**异步传输**

1. 前台：无需重新加载整个网页的情况下，能够更新部分网页的技术。
2. 后台：与服务器进行少量数据交换，Ajax 可以使网页实现异步更新。

不重新加载整个网页的情况下，对网页的某部分进行更新。

1. 传统的网页（不使用 Ajax）如果需要更新内容，必须重载整个网页页面（同步）。

**案例赏析：（待续）**

## 选择一门开发语言

现在，我们已经有了前端页面，也对页面中如果用户输入不准确便给错误提示做了处理，但是还缺少一个非常重要的特性**“交互”，于是我们迫切需要选择一门合适的开发语言进行开发**

**企业如何选型语言、数据库、web服务器？**

语言：java

数据库：首选oracle、中小型项目mysql

web服务器：多样化（Tomcat、Apache、Jboss、WebLogic）

语言：php

数据库：首选mysql

web服务器：Apache

语言：.net

数据库：mssql

web服务器：IIS

## 用SQL写一个登录

## post/get



## cookie/session

## 作业：完成登录，有基础的完成CURD

# 部署

## 部署在服务器

1、部署在公司服务器注意事项

2、部署在虚拟主机注意事项

## 学会看日志

对于测试工程师，通过分析日志，我们能辅助定位bug所在，因此学会看日志也就成了必会的基本功

日志级别：debug、error、info

日志系统：log4j、slf等

日志中，一般记录

1、异常，如：界面操作不报错，通过查看日志，发现日志抛出了异常，通过读异常，定位具体方法和代码行出错

2、SQL语句，如：信息准确，但登录不上，通过查看日志中暴露出的SQL语句，发现是SQL有问题

3、数据，如：界面没有显示该数据，查看日志，发现数据传过来了，那说明是开发调用时的代码有问题

另外：不同的系统，程序员写日志的内容不一样，要具体问题具体分析，更有甚者有一套完整的日志系统和分析、归纳、整理的日志系统，用于分析用户行为（腾讯模式的成功）

如：一般公司要求，凡有SQL的地方都要日志输出，这样方便定位是否是SQL有问题导致的错误或SQL性能低下导致性能问题。

## F12调试

## 升级

升级总原则：能够回退到升级前版本