学习方法

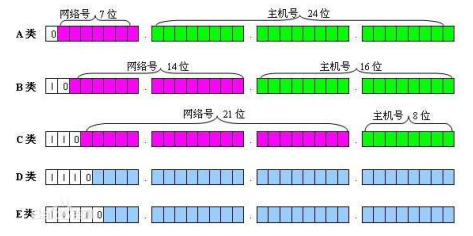
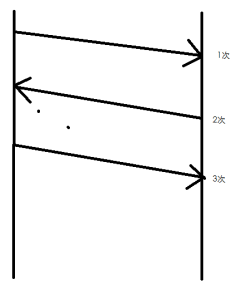
1. 建立逻辑思维
   1. 多分析
   2. 理解逻辑算法
   3. 多练、多敲、多读
2. 编程习惯
3. 大胆动手
4. 保持好奇好
5. 乐观面对错误

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

cookie

1. 通信协议
   1. 通信协议是什么？
      1. 交流的规则
      2. 举例：汉语、英语、法语、德语
2. TCP/IP
   1. IP协议Internet Protocol
      1. 4个字节，一共32位  
         子网掩码的作用？255.255.255.0  
         确认两台计算机是否处于同一网段  
         
   2. TCP、UDP的区别
      1. 如果TCP比作是打电话，那么UDP就是在发短信  
         TCP面向连接的协议（通信之前必须建立连接）  
         于是，TCP相对可靠，它建立连接的过程称为3次握手  
         第一个特点：三次握手，建立连接  
           
         第二个特点：所有的消息，需要对方确认送达。  
         “土豆，土豆，我是茄子，收到请回答”  
         “茄子，茄子，我是土豆，收到消息”  
         当消息发送失败，则对当前消息开始进行重复发送，直至收到回应为止。  
         “茄子，茄子，我是土豆，我被人油炸了，我改名叫薯片，收到请回答”  
         ……  
         “茄子，茄子，我是土豆，我被人油炸了，我改名叫薯片，收到请回答，第2遍”  
         ……  
         “茄子，茄子，我是土豆，我被人油炸了，我改名叫薯片，收到请回答，第3遍”  
         ……  
         “薯片，薯片，我收到消息”  
         因此可以确保数据的准确送达  
         如：局域网游戏，往往都有这样的特点，当多人联机对战时，若有一个人掉线，其他所有玩家进入读秒状态，那么说明玩家和玩家之间采用了TCP协议。因为对于游戏来讲，它不允许丢失任何数据，不然有可能出现两边不一致的情况：我这边画面一刀把你砍死了，你那边画面却吃了个大血瓶抗住了….
      2. UDP面向数据报的协议（不可靠的协议）  
         无需建立连接  
         发送消息也无需对方确认  
         无法保证数据的发送顺序，以及准确率  
         数据的发送顺序是a…….b……..c……..d  
         由于网路延迟的原因，对方收到数据的顺序有可能是b…..d…….a  
         UDP通常用于视频、语言等通信（丢掉了一帧画面是无所谓的）
3. HTTP（无状态的协议）
   1. 基于TCP协议的一种高级协议，用于客户端和服务器直接的通信
4. Cookie的概念
   1. Cookie的诞生，为什么叫会话跟踪技术？
      1. 在一次会话从开始到结束的整个过程，全程跟踪记录客户端的状态（例如：是否登录、购物车信息、是否已下载、是否已点赞、视频播放进度等等）
      2. 你把cookie就当成是第一次跟服务器连接后，服务器发给你的身份牌，上面就记录跟你有关的信息，以后只要再跟服务器通信，必须带着这个身份牌。这样一来，关于你是谁？有没有登录过？购物车里有什么信息？服务器当然就很容易知道了。
5. Cookie的特点
   1. 只能使用文本文件（如果浏览器可以随意在客户端机器上生成文件，比如身份牌是个定时炸弹，安全问题将会变得非常严重）
   2. 文件有大小限制（4KB）（文件若没有大小限制，比如身份牌的重量是140斤，挂脖子能不能累死？）
   3. 数量限制（一般浏览器，限制大概在50条左右，你家的门禁卡里能存的下一部蓝光高清吗？）
   4. 读取有域名限制  
      不可跨域读取，只能由来自写入cookie的同一域名的网页可进行读取。  
      简单的讲就是，谁写的cookie，谁才有权利读取  
      （身份牌是我发你的，当然只有我能读取，你媳妇儿的手机自动连接了邻居老王家的wifi，你知道这意味着什么吗？）
   5. 时效限制：每个cookie都有时效，最短的有效期是，会话级别：就是当浏览器关闭，那么cookie立即销毁  
      （安全学基本理论：密码锁每次打开都需要重新输入密码，输入一次密码，以后就不再验证，就没有安全可言。问：信用卡为什么会有过期时间？）
6. Cookie的使用
   1. 添加一个cookie
      1. Document.cookie = “username=123”;//一次写入一个键值对，一次写入多个，只有第一个生效，或者采用&符号链接user=张志林&password=123&name=xxxx;
      2. 保存形式：key=value
      3. 如果有中文编码问题，使用encodeURIComponent(“xxxx”)编码，再使用decodeURIComponent(document.cookie)解码，能解决中文乱码问题
   2. Cookie失效时间
      1. 如果不加这个参数，默认情况下为临时cookie，也就是关闭浏览器则cookie失效。
      2. Expires=date,指定在什么时间，cookie被自动清理，如果手动调整系统的时间，让cookie消失，可以进行恢复，此时为假删除，当覆盖某个键值对，并设置失效时间过期时，则cookie被删除，不能恢复
      3. Var date = new Date():  
         date.setDate(date.getDate() + 7);  
         document.cookie = “user=” + encodeURIComponent(“张志林”) + “;expires=” + date;
      4. Cookie的完整存储形式  
         name=value [;expires=date] [;path=path] [;domain=michael.com][;secure];  
         中括号是可选，name=value是必选。
      5. Cookie的路径问题  
         在同一路径下的网页可以共享cookie，路径不同时，不能访问
      6. Cookie的域名问题
         1. 必须在绑定域名的服务器上才可以设置域名
         2. 并且只能设置绑定的域名，也就是说，不同服务器间的cookie文件不共享
      7. Secure安全设置：指明必须通过安全的通信通道来传输（https）才能获取cookie.

第二节：应用

1. 一周内免登陆
2. 商品浏览记录

第三节

1. Cookie的封装
   1. //如何创建cookie  
      function setCookie(key,value,expires,path,domain,secure){  
       //取出键值对  
       var cookie = encodeURIComponent(key) + '=' + encodeURIComponent(value);  
       if(expires instanceof Date){  
       cookie += ";expires=" + expires;  
       }  
       if(path){  
       cookie += ";path=" + path;  
       }  
       if(domain){  
       cookie += ";domain=" + domain;  
       }  
       if(secure){  
       cookie += ";secure";  
       }  
       document.cookie = cookie;  
       }

//创建一个cookie的失效时间  
 function setCookieDate(day){  
 var date = null;  
 if( typeof day == 'number' && day > 0){  
 date = new Date();  
 date.setDate(date.getDate() + day);  
 }  
 return date;  
 }

//获取指定键的cookie值  
function getCookie(key){  
 //将用户查找的key进行编码并连接一个=  
 var cookieName = encodeURIComponent(key) + '=';  
 var cookieValue = "";  
 //查找到要截取的起始位置  
 var cookieStart = document.cookie.indexOf(cookieName);  
 if ( cookieStart > -1){  
 //查找到要截取的结束位置  
 var cookieEnd = document.cookie.indexOf(";",cookieStart);  
 //如果没有查找到;，则结束位置就是整个字符串的长度  
 if ( cookieEnd == -1){  
 cookieEnd = document.cookie.length;  
 }  
 //将需要的cookieValue截取出来并解码  
 cookieValue = decodeURIComponent(document.cookie.substring(cookieStart + cookieName.length,cookieEnd));  
 }  
 return cookieValue;  
 }  
//删除cookie  
function removeCookie(key){  
 document.cookie = key + "=;expires=" + new Date(0);  
 }

1. 购物车的应用
2. XSS攻击
   1. Xss(cross-site scripting)攻击指的是攻击者往Web页面里插入恶意html标签或者javascript代码，当用户浏览该页或者进行某些操作时，攻击者利用用户对原网站的信任，诱骗用户或浏览器执行一些不安全的操作或者向其它网站提交用户的私密信息。  
      比如：攻击者在论坛中放一个看似安全的链接，骗取用户点击后，窃取cookie中的用户私密信息；或者攻击者在论坛中加一个恶意表单，当用户提交表单的时候，却把信息传送到攻击者的服务器中，而不是用户原本以为的信任站点。
   2. XSS攻击的危害包括盗取各类用户帐号，如机器登录帐号、用户网银帐号、各类管理员帐号  
      控制企业数据，包括读取、篡改、添加、删除企业敏感数据的能力  
      盗窃企业重要的具有商业价值的资料  
      非法转账  
      强制发送电子邮件  
      网站挂马  
      控制受害者机器向其它网站发起攻击

第四节：应用

1. 照片墙的记录
2. 购物车的多条记录保存

第五节：综合应用

1. 掌握cookie的安全策略
2. 掌握cookie的封装
3. 掌握cookie跟json的配合