**今天学习的内容**

**1:复习昨天知识重点**

**2:作业**

**2.1:用户列表**

**2.2:登录**

**3:今天的目标**

**3.1:localStorage,本地存储对象(跨会话级存储)**

**在浏览器能管理外存(磁盘)中存储用户数据，可供此次会话及后续会话的页面共同;即使浏览器关闭了也不会消失--永久存在.**

**作用:用户样式风格(qq空间)**

**localStorage[key]=val; //保存一个数据**

**localStorage.setItem(key,val); //保存一个数据**

**var val = localStorage[key]; //读取一个数据**

**var val = localStorage.getItem(key); //读取一个数据**

**localStorage.removeItem(key); //删除一个数据**

**localStorage.clear() //清除数据**

**localStorage.length; //数据个数**

**localStorage.key(i); //获取第i个key**

**localStorage中若数据发生了修改,会触发事件 window.onstorage事件,可以监听事件，实现监视localStorage数据改变目的.--不能监视sessionStorage数据修改.**

**练习:修改用户样式**

**创建04\_index.html,有一个下拉菜单"请选择您喜欢的主题",**

**"蔚蓝天空","芭比公主","暗黑主题"**

**下拉列表选项对应 class名**

**.blue{background:#ddf;color:#33a;};**

**.pink{background:#fdf;color:#a3a;};**

**.dark{background:#333;color:#eee;};**

**用户选中某个选项(select.onchange) 为当前body使用指定样式**

**创建05\_usercenter.html打开此页面即可应用04\_index.html中一样的主题，即使重启浏览器，直访问05\_usercenter.html仍是之前主题.**

**返回04\_index.html,也应用之前选中主题**

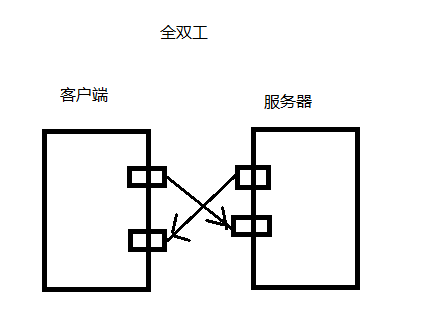
**3.2: HTML5新特性--WebSocket (服务器-客户端)**

**--代码不复杂重点在原理的理解**

**HTTP协议:属于"请求-响应"模型,只有客户端发起请求消息,服务器才会返回响应消息，没有请求就没有响应一个请求，只能得到一个响应，有些场景此模型就力不从心 "股票实时走势应用".**

**解决方案:长轮询(Long-Polling)/心跳请求，定时器+AJAX--请求过于频繁，服务器压力太大，不够频繁，客户端数据延迟太大。**

**WebSocket:属于"广播--收听"，客户端连接到服务器就不再断开,永久连接，双方就随时向对方发送消息，且是全双工不对等发送.**

****

**ws协议的应用程序分为客户端和服务器端程序**

**ws客户端应用**

**主动发起连接请求，保持连接，向对方发送消息,**

**并接收消息**

**(php;java;nodejs/html5)**

**使用HTML5创建ws协议客户端应用**

**1:连接WS服务器**

**var socket = new WebSocket("ws://127.0.0.1:9001");**

**2:向服务器发送消息**

**socket.send(stringMsg);**

**3:接收服务器发来消息**

**socket.onmessage = function(e){**

**console.log(e.data);**

**}**

**4:断开到WS服务器连接(可选)**

**socket.close();**

**练习:单词测试系统**

**1:用户可以在save.html中不停录入新单词**

**2:进入test.html开始测试，需要对之前录入单词进行**

**测试**

**3:提交答案后,在result.html显示测试过成绩**

**提示:单词录入需要永久保存,而此测试结果只要保存会话中**

****

**3.3:Bootstrap --->(重点&难点)**

**3.4:什么是响应式网页**

**1G:只能通话**

**2G:可以通话，短信**

**2.5G:GPRS 可以通话，短信，上网--WML**

**3G:IOS/Android(Linux) 可以通话，短信，视频，上网-HTML**

**京东:www.jd.com m.jd.com**

**Responsive Web Page:响应式网页/自适应网页,2010年提出,**

**一个网页，会自动根据用户浏览设备不同，自动改变布局，**

**可以PC/PAD/PHONE正常浏览.**

**响应式网页必备**

**(1)流式布局: float:left; display:inline-block;**

**(2)可以改变尺寸图片**

**(3)可以改变大小的文字**

**响应式网页的不足**

**代码更加复杂，不适合大型页面**

**3.5：如何编写响应网页----近几天重点&难点**

|  |
| --- |
| **Viewport (视口)**  **早期3G手机为浏览器大尺寸网页，只能强行把页面缩小，**  **导致图片、文字、超链都很小.**  **IOS提出了“视口”视口用于盛放网页内容，尺寸可以随意大小.**    **Android也借鉴该概念** |

**如何做网页手机适配**

**(1)声明viewport元标签**

**<meta name="viewport"**

**content="width=device-width, initial-scale=1.0, user-scalable=no ">**

**width=device-width:表示视口宽度是设备屏幕宽度**

**initial-scale=1.0 表示初始缩放比例**

**user-scalable=no 表示用户是否可以调整缩放比例**

**(2)使用流布局**

**实现方法:float:left; display:ineline-block**

**(3)所有容器使相对尺寸,不用绝对尺寸**

**.container{**

**/\*width:1440px;\*/**

**width:100%;**

**}**

**(4)所有文字使用相对尺寸,不用绝对尺寸**

**body{**

**/\*font-size:12px;\*/**

**font-size:1.2em;**

**}**

**(5)所有图片使用相对尺寸, 不用绝对尺寸**

**img{**

**width:92px;**

**width:50%;/\*指定父容器中的宽度占比\*/**

**}**

**(6){最重要原则}使用CSS3 Media Query (媒体查询)技术**

**Media:指浏览网页设备,如screen(pc/pad/phone),print**

**Media Query :查询出当前浏览网页的设备类型,以及特性**

**(如解析度，对比度，尺寸，手持方向等)**

**不同，而有选择性的执行某些CSS代码,而忽略另外一些**

**媒体查询两种使用方法**

**(1)根据媒体查询结果执行不同外部CSS文本**

**<link media="screen and (min-width:768px) and (max-width:991px)" rel="stylesheet" href="pad.css" />**

**(2)根据媒体查询结执执行不同CSS片段**

**@media screen and (min-width:768px) and**

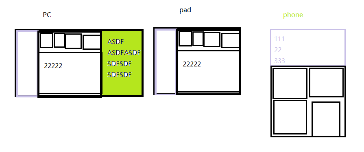
**(max-width:991px){**

**选择器{样式设定}**

**}**

**作业1:学子商城:用户列表**

**作业2:使用CSS 3 Media Query编写响应式网页**

****