**今天学习的内容**

**1:复习昨天的内容**

**2:作业--(重点)**

**3:今天的目标**

**3.1:html5新特性---拖放API**

**拖放源对象:(会动)**

**dragstart 拖动开始**

**drag 拖动中**

**dragend 拖动结束**

**dragstart\*1+drag\*n+dragend\*1**

**拖放目标:(不动)**

**dragenter 拖动进入**

**dragover 拖动悬停在上方**

**dragleave 拖动离开**

**drop 拖动释放**

**拖动过程:dragenter\*1+dragover\*n+dragleave\*1**

**拖动过程:dragenter\*1+dragover\*n+drop\*1**

**注意:必须阻止dragover的默认行为 drop才可能触发**

**练习:使用拖动事件的源对象和目标对象触发7个事件**

**实现"拖动删除效果"**

**提示:在刚开始拖动(src.ondragstart)记录被拖动源对象id,目标释放(target.ondrop)根据此id删除对象**

**3.2:html5新特性---地理定位**

**Geolocation:地理定位,使用js获取浏览器所在地址坐标(经度;纬度;海拨;速度),用于实现LBS Location Based Service 如果饿了么,滴滴打车...**

**手机浏览器**

**(1)首先手机GPS芯片与网络通信定位精度米**

**(2)次选手机浏览器通信基站定位获取,精度公里**

**PC浏览器**

**(3)通过电脑IP地址计算获取位置,定位公里**

**HTML5提供新对象:用于获取当前浏览器定位信息**

**window.navigator.geolocation{**

**getCurrentPosition:fn 获取当前定位信息**

**watchPostion:fn 监听定位变化**

**clearWatch:fn 清除监听**

**}**

**3.3:html5新特性---地理定位**

**百度(百度地图) 腾讯(腾讯地图)**

**百度开发者中心 http://lbsyun.baidu.com/**

**(1)注册百度地图开发帐户手机**

**(2)创建网站申请访问密码(密钥) AccessKey**

**(3)在自己网页中嵌入百度地图 API**

**百度地图开发流程**

**(1)创建地图实例**

**var map = new BMap.Map("container")**

**(2)创建坐标点**

**var point = new BMap.Point(经度,纬度)**

**(3)以坐标点为中心显示**

**map.centerAndZoom(point,15);**

**3.4:html5新特性---Web Worker**

**程序:proram 指可以被CPU执行代码，存储外储**

**进程:process/task 将程序调用内存中并分配空间，随时供CPU执行**

**线程:thread 线程是进程内部执行代码单位**

**chrome浏览器中线程模型**

**一个chrome进程内部至少有6 个线程在向web服务器发起请求**

**以获取资源**

**还有一个线程负责所有内容绘制并且执行代码(css/js) -UI主线程**

**观察如下代码**

**<button>按钮一</button>**

**<script src="x.js">很耗时js任务</script>**

**<button>按钮二</button>**

**问题:在网页执行过程中绘制内容和执行js同一个线程,**

**有可能先显示按钮等待再显示其它内容**

**解决方案:创建新线程执行耗时JS任务**

**<button>按钮一</button>**

**<script>**

**new Worker("x.js"); #创建Worker并且起动新线程执行x.js**

**</script>**

**<button>按钮二</button>**

**注意:html5 要求如果程序Worker线程，程序中不能出现**

**任何DOM/BOM 代码**

**-Worker线程数据传递**

**(1)UI 线程发送 Worker 接收**

**(2)Worker线程发送 UI线程**

**- UI 线程发送 Worker 接收**

**UI 发送**

**var w = new Worker("x.js");**

**w.postMessage(stringMsg)**

**Worker 接收**

**onmessage = function(e){e.data}**

**-Worker发送 UI 接收**

**UI 接收**

**var w = new Worker("js/05.js");**

**w.onmessage = function(e){e.data}**

**Worker 发送**

**postMessage(stringMsg)**

**练习:在HTML中创建input元素(输入数值)**

**创建button按钮"开始计算累加和"**

**点击此按钮,创建 worker线程来计算用户输入数字的累加和**

**计算完成将结果发送UI，在 button按钮下<p>显示结果**

**(1)创建html**

**(2)创建06.js 计算累加和**

**(3)接收发送数据**

**小结:**

**(1)项目中何时使用Worker**

**-如果js 代码中出现DOM或者BOM代码不能用Worker(jquery)**

**-Worker适合执行耗时JS任务(复杂计算/统计信息)**

**3.5:html5新特性---Web Storage**

**在浏览器中存储当前用户专用数据[购物车]**

**注意:在客户端存储数据最基要求,安全性要求低数可以保存浏览器**

**密码:安全性高数据 购物车:安全性低数据**

**客户端存储数据**

**(1)Cookie 浏览器兼容性好,不能超4KB,操作复杂**

**(2)Flash 依赖Flash播放器**

**(3)Web Storage 不能超8MB,操作简单**

**(4)IndexDB 存储量大，还不标准**

|  |
| --- |
| **Session:会话(操作过程)浏览器从打开某个网站的第一个页面开始**  **(会话开始)中间可能打开多个网页(会话中)直到关闭浏览器(会话结束)**  **整个过程称为:"浏览器与web 服务器的一次会话"** |

**webstore技术中浏览器为用户提供二个对象**

**-sessionStorage**

**此对象仅供此次会话所有页面共同使用，一旦关闭浏览器则**

**数据消失**

**(1)获取数据 var value = sessionStorage.getItem(key)**

**var value = sessionStorage[key];**

**(2)修改数据 sessionStorage.setItem(key,value)**

**sessionStorage[key] = value;**

**(3)清除数据 sessionStorage.clear();**

**sessionStorage.removeItem(key)**

**(4)其它 sessionStorage.length 数据个数**

**var key = sessionStorage.key(i);**

**练习:使用sessionStorage保存用户名**

**创建index.html 右上角显示 "请登录" 超链接**

**创建login.html 输入用户名 提交按钮**

**用户点击提交按钮 提示:登录成功 3s后跳转首页**

**setTimeout()+location.href**

**-localStorage**

**此对象供此次会话以及后续会话使用，即使浏览器关闭数据存在**

**作业1:使用 echarts 完成销售统计图(柱状)要求指定不同颜色**

**百度**