今天学习内容

1:复习学子商城项目

2:今天学习的内容

2.1:html5 三个新特性—web Worker/web Storage/web Socket

2.2: html5 三个新特性—web Worker—理论多，代码少3行

程序:保存在外存中代码 1.html/1.css/1.js

进程:将程序调入内存中并且分配空间,在内存中运行程序称为 “进程”.

线程:处理进程内存，用于执行代码

并发:“宏观上看同时执行””微观上看依次交替执行”

并行:同时执行

chrom:中线程模型:请求资源线程—多个

运行js代码,渲染页面内容-1个

<button>按钮1</button>

<script src=”1.js”></script>

<button>按钮2</button>

解决方案:创建一个并发新线程，让它来执行耗时js任务

代码:

var w1 = new Worker(参数运行js地址);

#浏览器不允许Worker线程中操作(代码)不能有任何DOM和BOM对象.

原因:浏览器只允许UI线程操作DOM/BOM，若多个线程同时来操作DOM和BOM页面混乱.所以类似JQUERY也不能使用Worker中.

(\*)UI线程发送数据给Worker

woker(收)

onmessage = function(e){e.data}

ui(发)

var w6 = new Worker(“6.js”);

w6.postMessage(stringmsg);

(\*)Worker线程发送数据UI

UI(收数据)

var w6 = new Woker(“6.js”);

w6.onmessage = function(e){

e.data..

}

Woker(发数据)

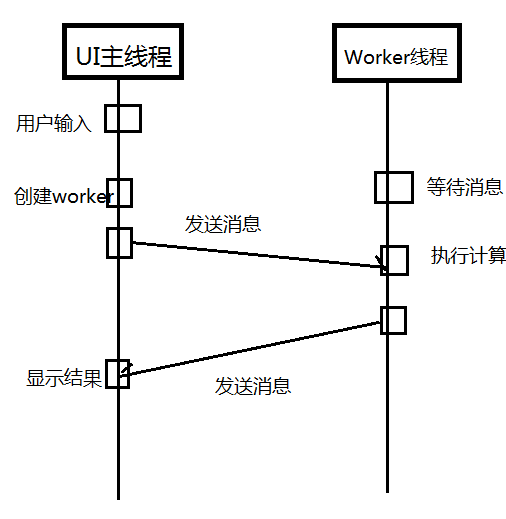
postMessage(执行结束);

练习:在HTML中有一个input,再添加按钮按钮"开始计算累加和"

,点击按钮时，创建一个Worker线程计算出用户输入数字

的累加和，下一个div显示计算结果input用户输入数字

button 开始计算累加和 div 显示计算结果



|  |
| --- |
| 项目中worker使用  1:只要js中有DOM和BOM代码不能使用Worker  2:Worker适合执行耗时JS任务, 复杂计算，加密和解密.. |

2.3: html5 三个新特性—web Storage

在浏览器中存储当前用户专用数据:访问历史，内容定制，样式定制

在客户端存储数据可以使用技术

1. Cookie技术：浏览器兼容性好;操作复杂;不能超4KB

document.cookie="uid=10"; 保存一对数据cookie

document.cookie; 读取cookie数据

(2)Flash存储:依赖于Flash播放器

(3)H5 webstroage :不能超过8MB,操作简单

(4)IndexDB :可存大量数据，不是标准技术

|  |
| --- |
| Session:会话.“一个操作过程称为会话”  浏览器从打开某个网站的第一个页面开始（会话开始），中间可以打开多个页面(会话进行中)，直到关闭浏览器(会话结束),整个过程称为一次”会话”. |

webstorage 为浏览器提供二个对象

1. window.sessionStorage:类数组对象

作用:在同一个会话中所有页面之间共享数据,将数据保存sessionStorage中只限本次会话使用.

如:当前登录用户编号/权限列表/购物车信息/昵称

注意:安全性高数据不要存储sessionStorage

只要安全性高数据不要存储客户端

保存数据

sessionStorage[key] = val; //key数据名val 数据值

sessionStorage.setItem(key,val);

获取数据值

var val = sessionStorage[key];

var val = sessionStorage.getItem(key);

其它

sessionStorage.removeItem(key); 删除一对数据

sessionStorage.clear(); 清空所有数据

sessionStorage.length; 数据个数

sessionStorage.key(i); 获取第i个数据名

(2)window.localStorage

保存数据在磁盘上，可供此次会话以及及后续的会话中的页面共同使用,即使浏览器关闭也不会消失—永久存在：

作用:当前客户端对应所有会中共享数据,如:主机位置

保存数据

localStorage [key] = val; //key数据名val 数据值

localStorage.setItem(key,val);

获取数据值

var val = localStorage [key];

var val = localStorage.getItem(key);

其它

localStorage.removeItem(key); 删除一对数据

localStorage.clear(); 清空所有数据

localStorage.length; 数据个数

localStorage.key(i); 获取第i个数据名

练习:

创建index.html 首页

有一个下拉菜单,”请选择您喜欢的主题”,”蔚兰天空”,

”芭比公主”,”暗默主题”,各对应一个class名;

.blue{background:#ddf,color:#33a}

.pink{background:#fdf,color:#a3a}

.dark{background:#333,color:#eee}

option value=”pink”

document.body.className = pink

用户选中 select.onchange

创建 usercener.html 用户中心

打开此页面即可应用index.html中一样的主题样式,

启使重新启动浏览器，直接访问usercener.html仍是

之前选中的主题.

localStorage中若数据发生修改，会触发事件 onstorage事件，

可以监听此事件，实现监视localStorage数据修改.

2.4: html5 三个新特性—web Socket

HTTP协议:属于”请求-响应”，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，在些场景中不适合”实时走势图”

解决方案:”长轮询-心跳请求”—定时器+ajax

websocket协议:属于”广播-接收”模块，客户端连接到服务器不再断开，永久连接双方随时向对方发送消息.

ws:服务器端:php/java/node.js

ws:客户端:php/java/html5 提供对象

使用html5创建ws协议客户端

//连接ws服务器

var socket = new WebSocket(“ws://127.0.0.1:9001”);

//向服务器发送数据

socket.send(stringMes);

//接收服务器数据

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

//断开到服务器连接

socket.close();

3.1:Bootstrap [css 与js框架 响应式布局]—(栅格布局系统)

3.2:什么是响应式网页

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，上网，视频 - HTML

4G:

5G:...

3.3:Responsive Web Page:响应式网页/自适应的网页2010年提出,

一个网页，会自动根据用户浏览设备不同，自动必变布局,

可以被PC/PAD/PHONE 正常浏览

3.4:响应式网页必备

(1)流式布局 float:left;

(2)可以改变尺寸图片文字 font-size:1em;

响应式布局不足:代码更加复杂，不适合大型网页

3.5:如果测试网页

(1)使用真实物理设备

(2)使用第三方模拟器软件

(3)使用chrome 内置设备模拟器

3.5:如何编写响应式网页

(1)声明viewport元标签

<meta name="viewport" content="width=device-width, initial-scale=1">

|  |
| --- |
| viewport 元标签  早期3G手机为浏览器大尺寸网页，只能强行页面缩小,  导致图片/文字/链接  IOS提出”视口”概念:虚拟窗口，大小与设备相同，如果  浏览大尺雨网页，滑动.  Android 也借鉴该概念 |

#width=device-width 视口宽度=设备宽度

# initial-scale=1 表示初始化视口大小1原始大小

#user-scalable=no 用户是否可以调整视口大小

(2)使用流式布局

实现方法:float:left

(3)所有容器使用相对尺寸，不用绝对尺寸

.container{

/\*width:500px;\*/

width:50%;

}

(4)所有文字使用相对尺寸，不用绝对尺寸

body{

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

font-size:2em;

}

(5)所有图片使用相对尺寸，不用绝对尺寸

img{

width:120px;

width:25%;

}

(6)（最重要原则）使用CSS3 Media Query技术

3.6: CSS3 Media Query技术

Media:指浏览网页的设备,如screen(pc/pad/phone),print,tv,tty. （屏幕，打印机，电视）

Media Query:查询当前浏览网页设备类型，以及特性(对比度，尺寸，手持方向等),而选择性执行某些css代码，而忽略另外一些.

媒体查询使用方法两种

1. 根据媒体查询结果执行不同的css文件

<link media=”screen” href=”1.css”/>

<link media=”print” href=”2.css”/>

1. 根据媒体查询结果执行css片段中不同部分

作业1：学子商城,产品添加