今天学习的内容

1:复习昨天重点知识

HTML5新特性

(1)新的语义标签

(2)增强型表单

(3)视频和音频

(4)Canvas绘图

(5)SVG绘图

(6)地理定位

navigator.geolocation{}

(7)拖放API

拖动源对象 dragstart/drag{0,0}/dragend

拖动目标对象 dragenter/dragover{阻止默认行为}/dragleave/drop

(8)WebWorker

(9)WebStorage

(10)WebSocket

2:作业

3:今天的目标

HTML5新特性--WebWorker

程序:Progam 指可以CPU执行的代码，存储在外存中.

进程:Process/Task 指程序被OS调入内存， 分配执行空间，

随时供CPU调用执行.

线程:Thread :线程进程内执行代码基本单位

进程和线程:

(1)进程是操作系分配内存基本单位

(2)线程是CPU执行代码的基本单位

(3)线程必须处理某个进程内部

(4)一个进程内必须至少有一个线程也可以有多个

(5)一个操作系统可能同时存在几千个线程，它们是"并发执行的"--宏观看同时执行，微观上看依次执行.

Chrome浏览器中的线程模型

一个Chrome进程内，至少有6个线程，可以"并发"向WEB服务器发起HTTP请求，以获得所需的资源

---资源请求线程

还有**一**个线程负责将所有**内容绘制**到浏览器页面中--UI主线程--不允许多线程同时绘图，防止内容布局错乱.

<button>按钮1</button>  
<script src="3.js">很耗时的JS任务</script>  
<button>按钮2</button>

现象:JS执行过程中，按钮1可见，按钮2不可见

原因:浏览器中执行代码只有一个线程--UI主线程

解析办法:创建新的线程，由它来执行耗时JS任务;

UI主线程继续执行后续的HTML渲染.

<button>按钮1</button>  
<script>

var w = new Worker("3.js");

</script>

<button>按钮2</button>

3:Worker线程缺陷

浏览器不允许Worker线程操作任何的DOM、BOM对象

原因:浏览器只允许UI主线程操作DOM/BOM，若多个线程同时都可以DOM/BOM，页面混乱.

所以:类以JQUERY脚本不同使用Worker来加载执行.

Worker线程可以给UI主线程发数据消息

UI主线程

var w = new Worker("6.js");

w.onmessage = function(e){

e.data;//接收数据

}

Worker线程

postMessage(strinMsg);

UI主线程发数据消息Worker

WebSocket