复习：

HTML5新特性 重要程度(10)

1.新的语义标签 10

2.表单2.0 7

3.视频和音频 8

4.Canvas绘图 9

5.SVG绘图 9

6.地理定位 1

7.拖放API 4

8.WebWorker 4

9.WebStorage 9

10.WebSocket 2

拖放API——七个事件：

源对象可以触发的事件：

dragstart：

drag：

dragend：

目标对象可以触发的事件：

dragenter：

dragover： —— e.preventDefault()

dragleave：

drop：

今日目标：

(1)扩展：如何在网页中拖拽显示客户端的图片

(2)WebWorker

(3)WebStorage

1.面试题：如何在服务器端下载的网页中显示客户端图片？

一般情况下，网页只能显示服务器上的图片；

HTML5中，可以实现用户拖拽一张本地的图片显示在服务器端下载的网页中。

container.ondrop = function(e){

var f0 = e.dataTransfer.files[0];

var reader = new FileReader();

reader.readAsDataURL(f0);

reader.onload = function(){

///读取完成，数据在reader.result中

}

}

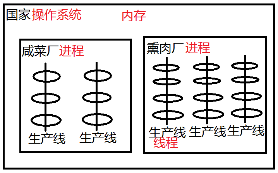
|  |
| --- |
| HTML5中提供的用于文件输入输出(I/O)的对象：  File：代表一个文件/目录对象  FileList：代表一个文件列表（类数组对象）  FileReader：用于从文件中读取内容  FileWriter：用于向向文件写出内容 |

2.面试题：Chrome浏览器的线程模型是怎样的？

**程序**：计算机可以执行的代码，存在于磁盘中——静止的

**进程**：把程序调入到内存中，等待被CPU执行——活动的

**线程**：是CPU执行进程代码的基本单位——生产任务



|  |
| --- |
| 面试题：进程和线程间的关系？  进程是操作系统分配内存的基本单位。  线程处于进程内部，是CPU执行代码的基本单位。  一个进程中至少有一个线程，也可以有多个。  多个线程间并发执行——宏观上看是“同时”执行，微观上看是“轮流”执行。 |

**Chrome中的线程模型：**

(1)Chrome中发起HTTP请求最多可以使用6个并发的线程；

(2)Chrome中负责向页面中执行绘制任务(执行HTML/CSS/JS/事件处理等代码)的只有1个线程——UI主线程。

碰到如下的代码，就有问题了：

<button onclick="console.log(111)">1</button>

<script **src="耗时.js"**></script>

<button onclick="console.log(222)">2</button>

根本解决之道：

创建一个新的线程，去执行耗时的JS任务——与UI主线程并发执行。new Thread('x.js') —— 其它语言中的类似代码。

3.HTML5新特性之八——Web Worker

**HTML5提供了一种类似创建新线程的机制：**

new Worker('x.js'); //工人线程，与UI主线程并发执行

注意：Worker线程天然缺陷！！

浏览器禁止Worker线程操作任何的BOM和DOM对象！！！——浏览器只允许UI主线程进行页面内容的渲染！——不能使用Worker加载类似jquery.js文件！！ 项目中，必需用到Worker的场景不多，比如有非常复杂耗时的且与DOM无关的运算时。

**UI主线程可以给Worker线程传递数据——发消息：**

UI线程：

var w = new Worker('x.js');

w.postMessage('StringMsg');

Worker线程：

onmessage = function(e){ //e.data //UI主线程发来的消息数据 }

**Worker线程可以给UI主线程传递数据——发消息：**

UI主线程：

var w = new Worker('x.js');

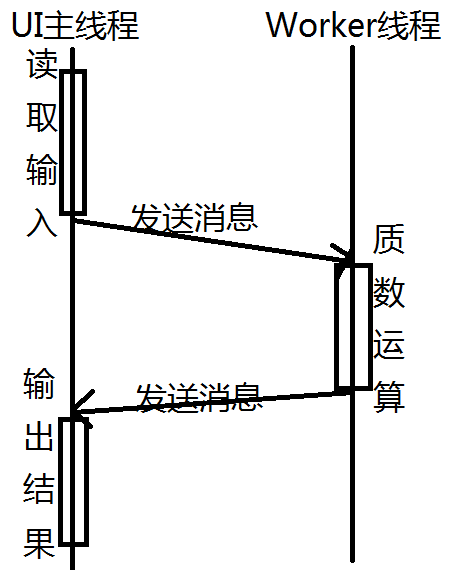
w.onmessage = function(e){ //e.data //Worker线程发来的消息数据 }

Worker线程：

.....

postMessage('StringMsg');

练习：在主界面中有一个输入框+按钮（“开始质数计算”），点击后创建一个新的Worker线程，把用户的输入传递给该线程，进行质数判定；.... 计算结果再反传给UI主线程，在一个DIV元素中显示出运算结果。



|  |
| --- |
| 总结：Worker本质是一个线程，在UI主线程之外并发执行的线程；用于执行耗时的JS任务；缺陷：不能操作BOM和DOM，只能和UI主线程发消息。 |

4.数据存储方案

Web项目中需要存储多种数据，大体上有两种存储方法：

**(1)服务器端存储：**

1)数据库技术：存储项目中的核心数据

2)Session技术：存储当前用户的信息

**(2)客户端存储：**不太紧要的与客户端有关的数据，如浏览记录、内容定制、样式定制

1)Cookie技术：使用比较繁琐、大小不能超过4KB

2)Flash技术：逐渐被淘汰

3)HTML5WebStorage技术：使用简单，大小不超过8MB

4)IndexedDB技术：客户端直接存储对象，目前还不是HTML标准技术

|  |
| --- |
| Web会话：指客户端浏览器从连接到某个Web服务器开始，一系列的请求-响应过程。直到客户端关闭浏览器，会话结束。 |

5.HTML5新特性之九——WebStorage

HTML5的WebStorage技术，提供了两个对象，用于在浏览器中存储客户端的访问数据：

**window.sessionStorage——会话级存储:**

在浏览器的内存中存储的与某个服务器间的一系列请求-响应过程中产生的数据——都是Key-Value对形式。当会话结束时(用户关闭了浏览器)，会话级数据即消失。

sessionStorage['key'] = 'value'; //存储一个数据

var v = sessionStorage['key']; //读取一个数据

sessionStorage.length //获取数据的个数

sessionStorage.setItem('key', 'value') //存储一个数据

var v = sessionStorage.getItem('key') //读取一个数据

sessionStorage.removeItem('key') //删除一个数据

sessionStorage.clear() //清除所有的数据

**window.localStorage——本地/跨会话级存储**：

在客户端文件系统/硬盘中存储客户端与服务器间的访问数据——都是Key-Value对形式。即使关闭浏览器，甚至关闭计算机仍然存在，否则会永久存在。

localStorage['key'] = 'value'; //存储一个数据

var v = localStorage ['key']; //读取一个数据

localStorage.length //获取数据的个数

localStorage.setItem('key', 'value') //存储一个数据

var v = localStorage.getItem('key') //读取一个数据

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

localStorage.clear() //清除所有的数据

练习：

(1)创建index.html，右上角有一个超链接“登录”，点击此链接跳转到login.html，用户在表单中输入登录信息，点击“提交按钮”，假设用户名和密码都正确，弹出提示“登录成功”，3秒钟跳转回首页，右上角显示出“欢迎回来：xxx 退出登录”；此时若用户在地址栏强行输入login.html，直接跳转到index.html；用户若点击了index.html上的退出登录超链接，则跳转到logout.html，提示出“您已退出登录”，3秒钟跳回index.html，继续显示“登录”超链接

setTimeout(function(){

location.href="";

},3000)

(2)实现一个页面主题的永久定制 17：55

创建index.html，指定3个class

.blue { background: #ccf; color: #339; }

.pink { background: #f3b; color: #916; }

.dark { background: #111; color: #eee; }

提供一个下拉选择框，“—请选择您喜欢的页面主题—”，“湛蓝天空”、“芭比公主”、“杀马特”，选择某项之后，当前页面的body就使用对应的className。

跳转到usercenter.html，默认也呈现出index.html中选中的页面主题，即使重启浏览器，仍可以呈现出之前选中的主题色。提示：即使重新打开index.html，也要显示出之前选中的主题色。

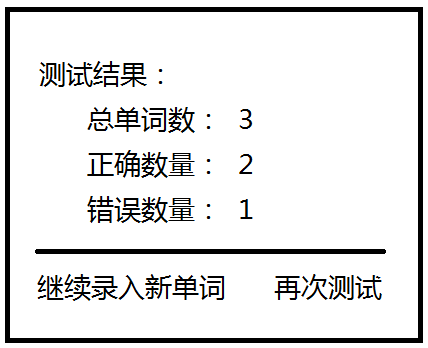
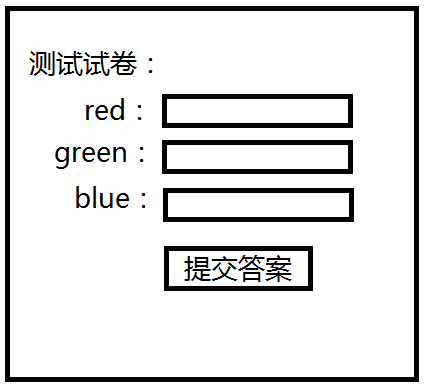
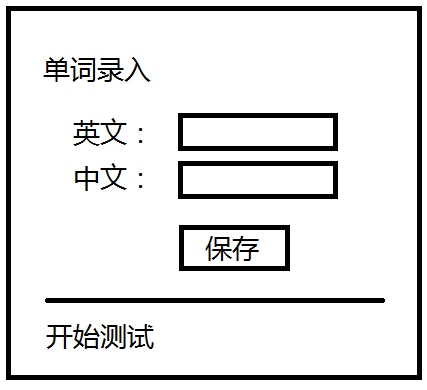
课后练习： 单词测试系统

(1)用户可以在save.html中不停的录入新单词；

(2)进入test.html开始测试，需要对之前录入过的所有单词进行测试；

(3)提交答案后，在result.html中显示出测试成绩。

提示：录入的单词需要永久保存；而此次测试结果只需要在当前会话中保存。



//遍历客户端存储的数据

for(var i=0; i<localStorage.length; i++){

var key = localStorage.key(i) //获取第i个key

var value = localStorage[key]; //获取第i个value

}