window对象：

是js访问浏览器窗口的一个接口，

是一个全局对象，定义在全局作用域的中的变量。

常见事件：

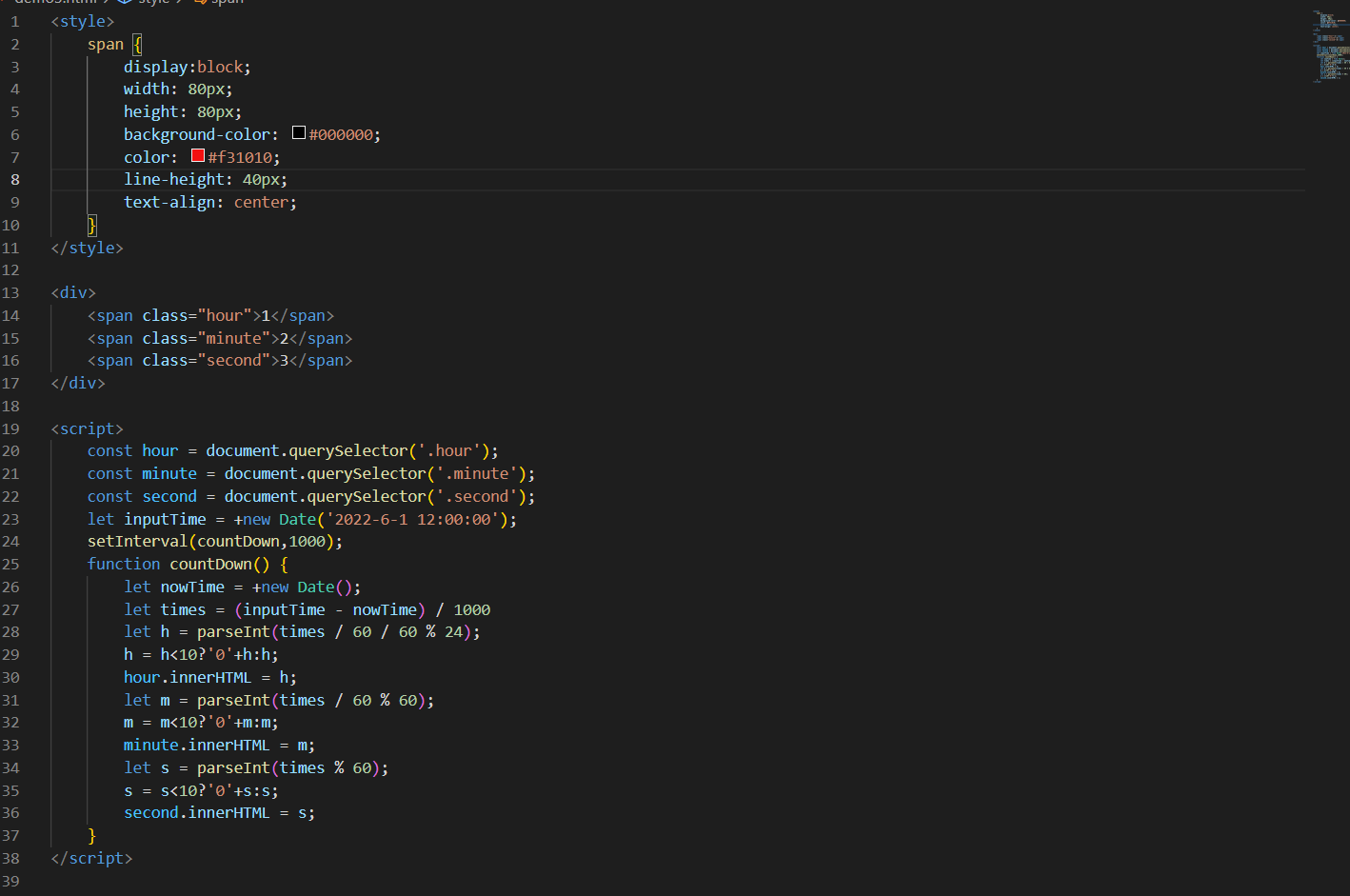
窗口加载事件 window.onload=function(){}/

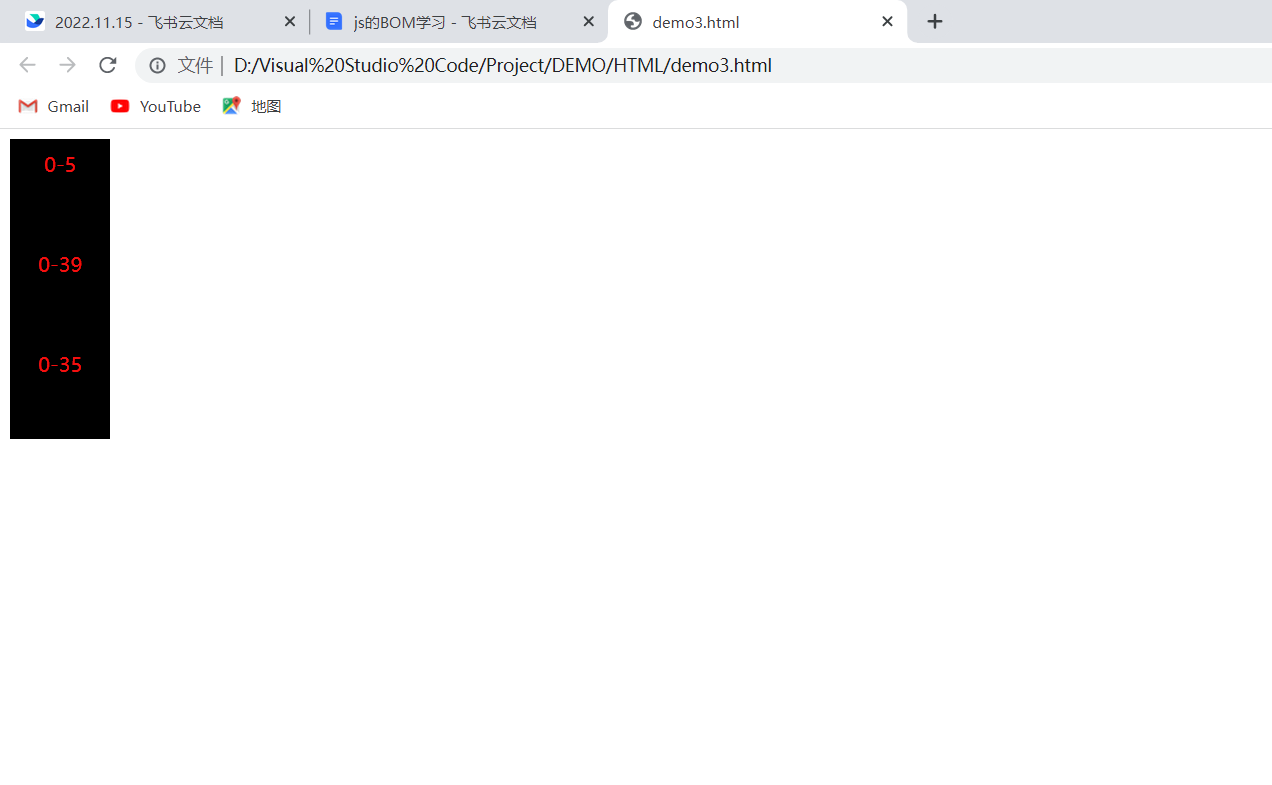
window.addEventListener("load",function(){})

调整窗口大小事件 window.onresize

计时器：

东京倒计时效果





# ****同步与异步****

HTML5 提出 Web Worker 标准，允许 JavaScript 脚本创建多个线程。

同步：前一个任务完后执行下一个

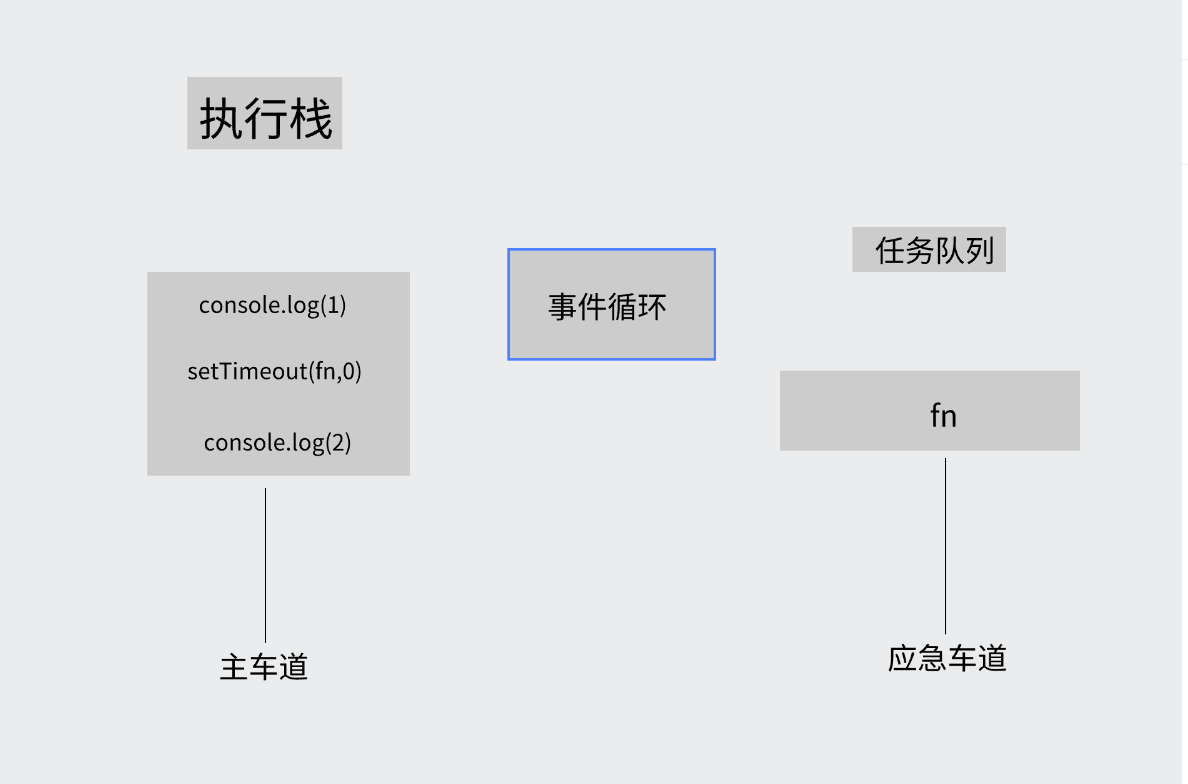
异步：在这个时间段中还可以做其它的

同步：同步任务都在主线程上执行

异步：js中的异步是通过回调函数实现的



异步任务相关回调函数添加到任务队列中



1.先执行执行栈中的同步任务 2.异步任务放入任务队列中 3.当执行栈中的所有同步任务执行完毕，系统就会按次序读取任务队列中的异步任务，于是被读取的异步任务结束等待状态，进入执行栈，开始执行

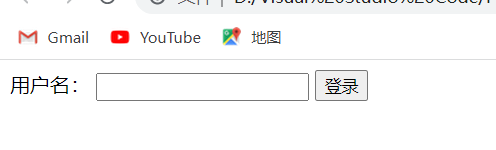
案例：获取URL数据

1.获取登陆页面

2.第二个页面使用第一个页面的参数实现不同页面的传递效果

3.利用URL中的location.search参数

4.在第二个页面需要把这个参数提取



window对象：

是js访问浏览器窗口的一个接口，

是一个全局对象，定义在全局作用域的中的变量。

常见事件：

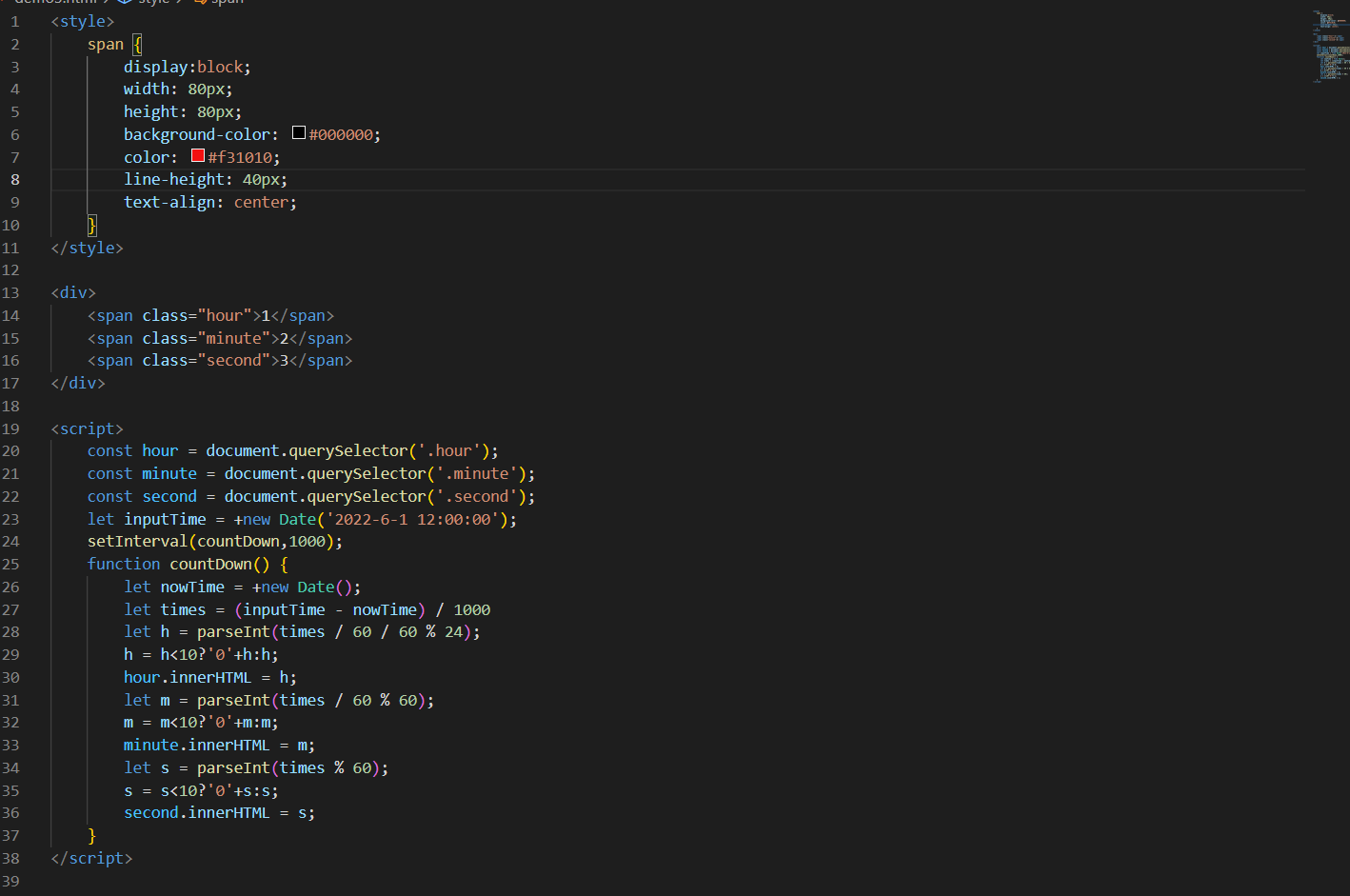
窗口加载事件 window.onload=function(){}/

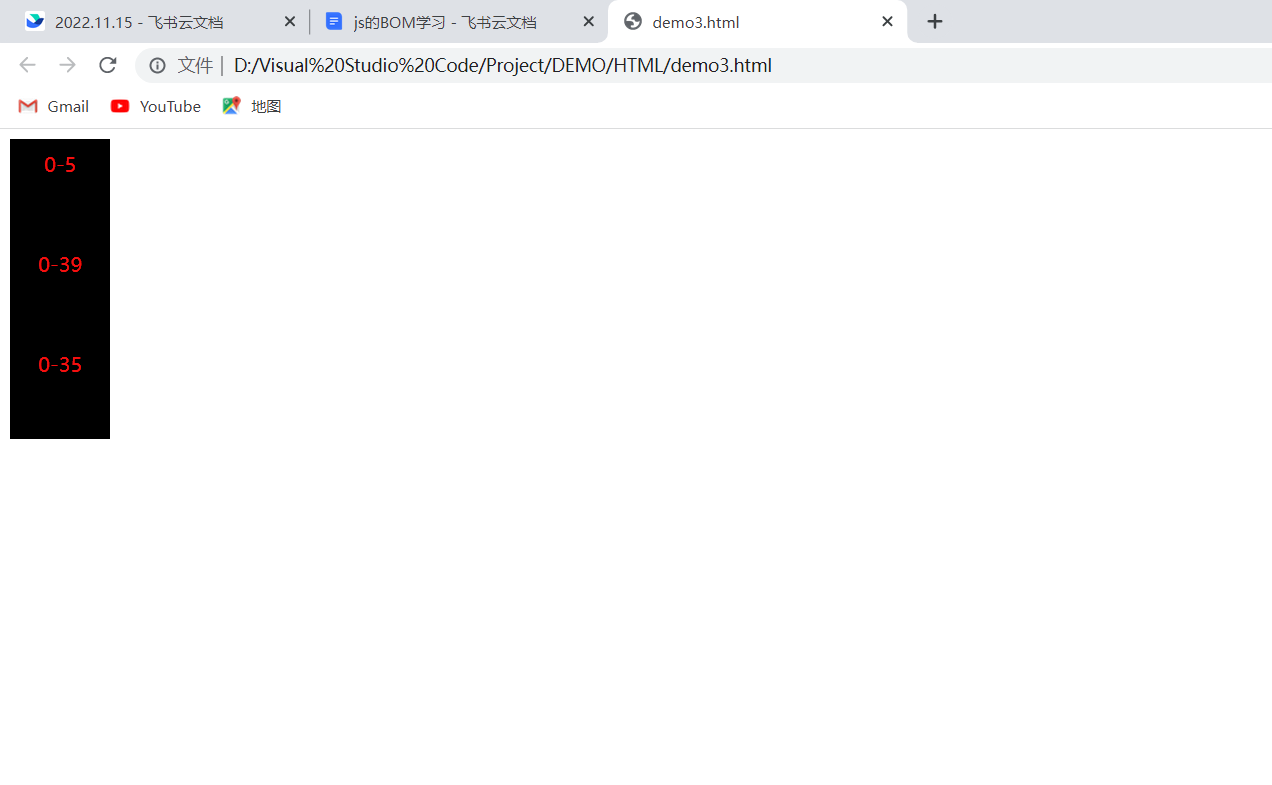
window.addEventListener("load",function(){})

调整窗口大小事件 window.onresize

计时器：

东京倒计时效果





# ****同步与异步****

HTML5 提出 Web Worker 标准，允许 JavaScript 脚本创建多个线程。

同步：前一个任务完后执行下一个

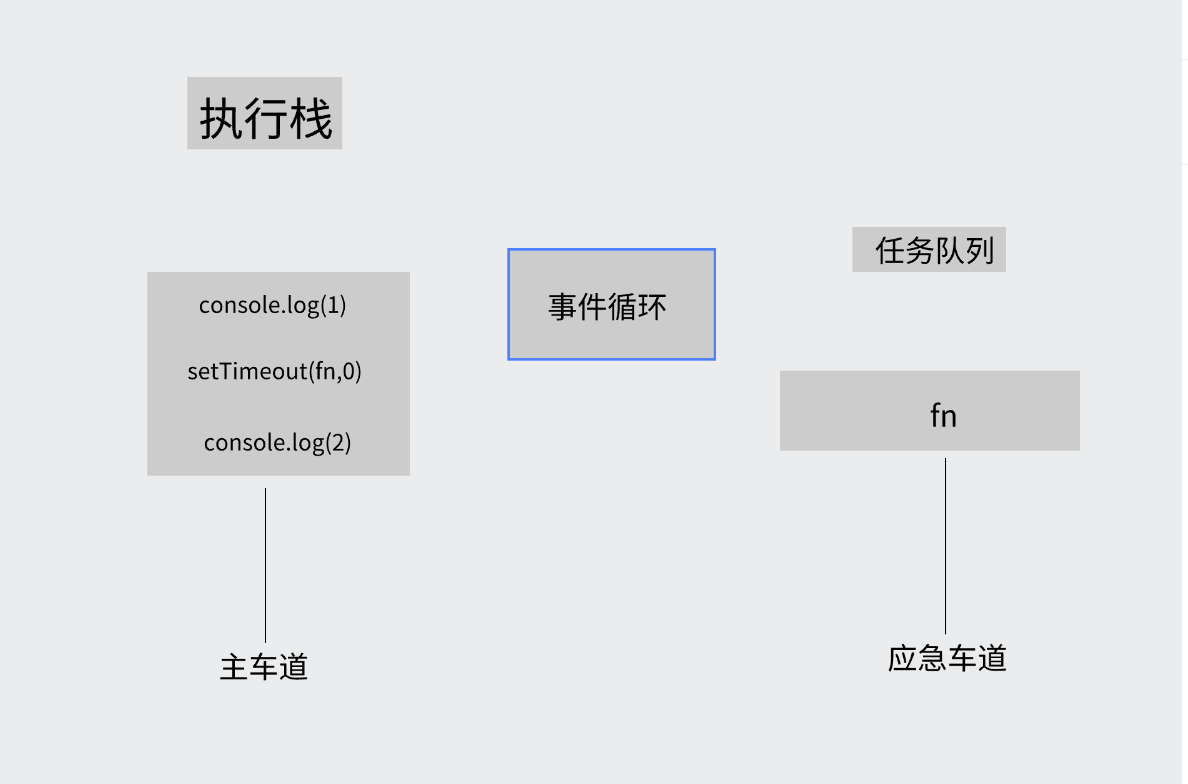
异步：在这个时间段中还可以做其它的

同步：同步任务都在主线程上执行

异步：js中的异步是通过回调函数实现的



异步任务相关回调函数添加到任务队列中



1.先执行执行栈中的同步任务 2.异步任务放入任务队列中 3.当执行栈中的所有同步任务执行完毕，系统就会按次序读取任务队列中的异步任务，于是被读取的异步任务结束等待状态，进入执行栈，开始执行

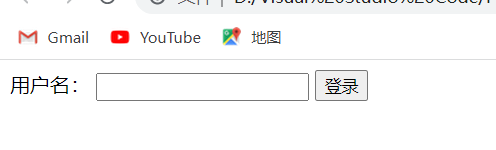
案例：获取URL数据

1.获取登陆页面

2.第二个页面使用第一个页面的参数实现不同页面的传递效果

3.利用URL中的location.search参数

4.在第二个页面需要把这个参数提取



window对象：

是js访问浏览器窗口的一个接口，

是一个全局对象，定义在全局作用域的中的变量。

常见事件：

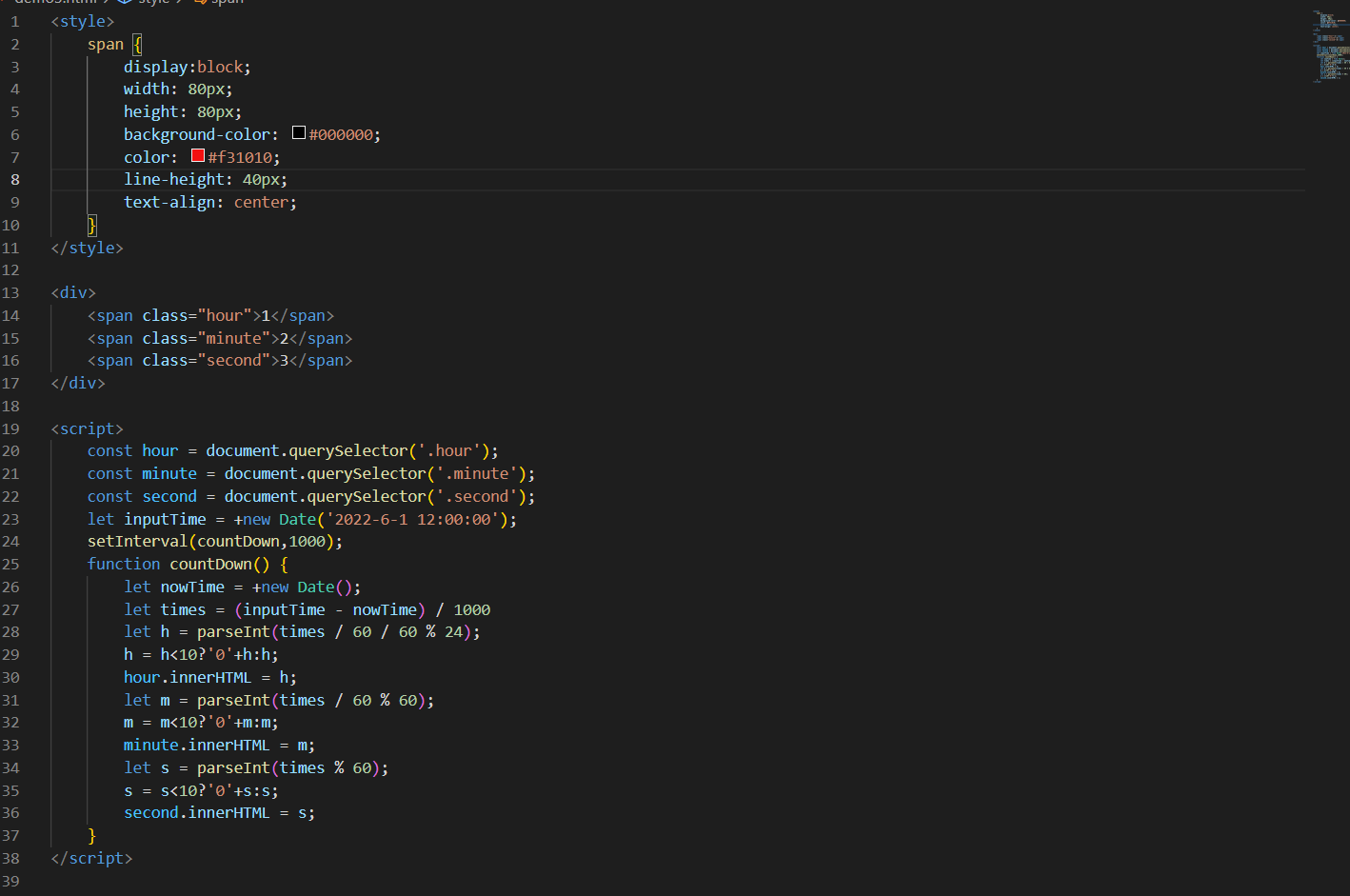
窗口加载事件 window.onload=function(){}/

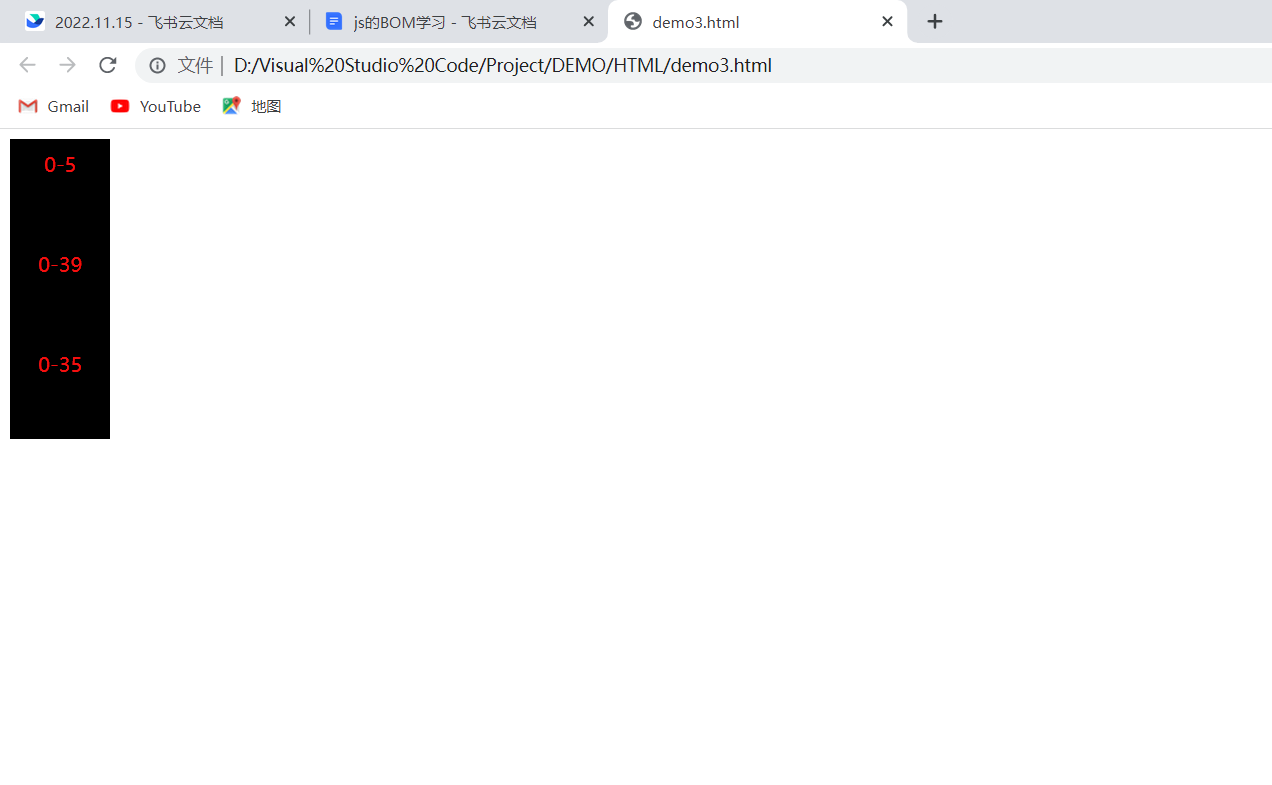
window.addEventListener("load",function(){})

调整窗口大小事件 window.onresize

计时器：

东京倒计时效果





# ****同步与异步****

HTML5 提出 Web Worker 标准，允许 JavaScript 脚本创建多个线程。

同步：前一个任务完后执行下一个

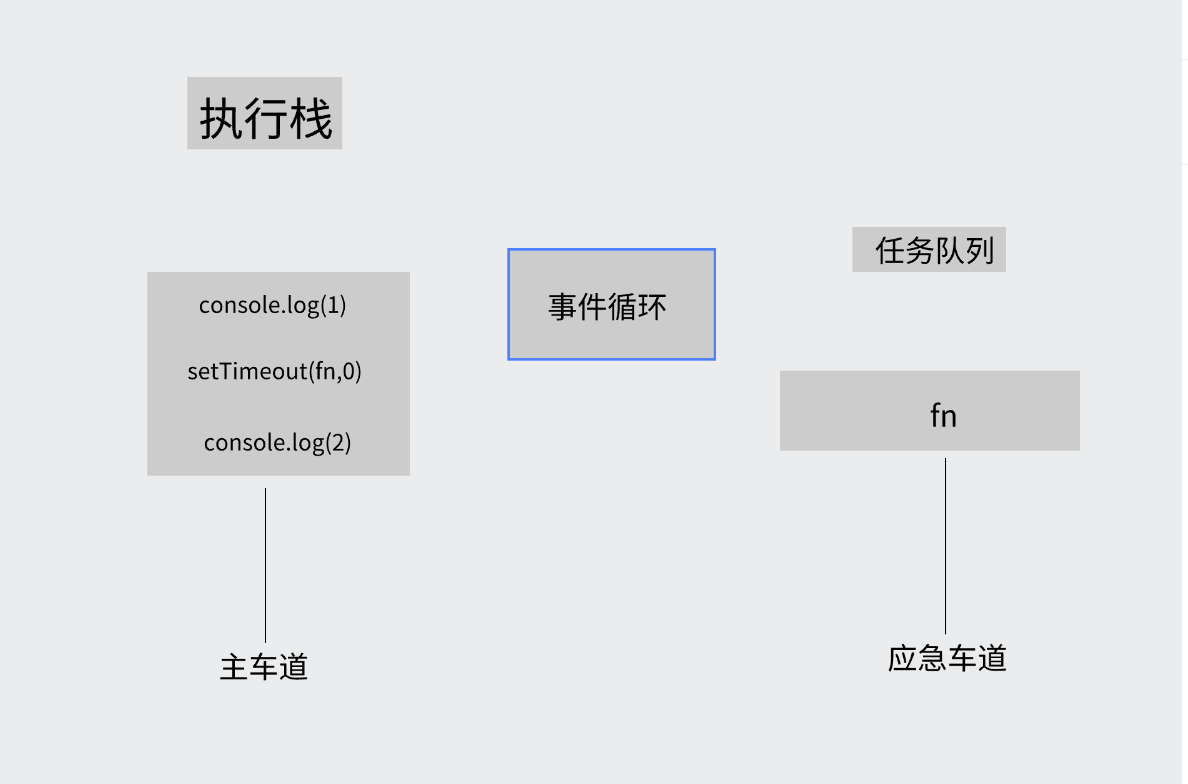
异步：在这个时间段中还可以做其它的

同步：同步任务都在主线程上执行

异步：js中的异步是通过回调函数实现的



异步任务相关回调函数添加到任务队列中



1.先执行执行栈中的同步任务 2.异步任务放入任务队列中 3.当执行栈中的所有同步任务执行完毕，系统就会按次序读取任务队列中的异步任务，于是被读取的异步任务结束等待状态，进入执行栈，开始执行

案例：获取URL数据

1.获取登陆页面

2.第二个页面使用第一个页面的参数实现不同页面的传递效果

3.利用URL中的location.search参数

4.在第二个页面需要把这个参数提取

