

Advanced web technology

# 高级Web技术

*RIA技术—HTML5*

# HTML5关键技术

---

## ■ Web Worker

### – 特征

- 允许JavaScript创建多个线程，子线程完全受主线程控制
- 不得操作DOM，不能访问全局变量（window、document之类的浏览器全局变量）或是全局函数，不能调用alert()之类函数

### – 两种web worker

- 专用线程dedicated web worker: 只能被创建它的页面访问,随当前页面的关闭而结束;
- Shared web worker可以被多个页面访问

# HTML5关键技术

## ■ Web Worker

### – dedicated web worker

- 可以通过注册监听器和onMessage（onError）两种方式进行主线程和子线程之间的通信。

```
var worker = new Worker("script/lenthytask.js");
worker.addEventListener("message", function (evt) {
    alert(evt.data);
}, false);
worker.postMessage(10000)
```

主线程

```
addEventListener("message", function (evt) {
    var date = new Date();
    var currentDate = null;
    do {
        currentDate = new Date();
    } while (currentDate - date < evt.data);
    postMessage(currentDate);
}, false);
```

子线程

# HTML5关键技术

## ■ Web Worker

### – dedicated web worker

- **postMessage(data)**: 用于子线程与主线程之间互相通信
- **terminate()**: 主线程中终止worker
- **onMessage**: 有消息时触发该事件, 消息内容可通过事件对象的数据来获取
- **error**: 出错处理。错误信息: e.message, e.filename, e.lineno

```
//创建一个Worker对象, 并向它传递将在新线程中执行的脚本url
var worker = new Worker('worker.js');
//接收worker传递过来的数据
worker.onmessage = function(event){
    document.getElementById('result').innerHTML+=event.data+"<br/>" ;
};
```

```
//worker = new Worker('url');
//worker.postMessage传递给子线程数据, 对象
worker.postMessage({first:1,second:2});

//子线程中也可以使用postMessage, 如传递字符串
postMessage('test');
```

# HTML5关键技术

---

## ■ Web Worker

### – web worker的Ajax调用

- 可以使用XMLHttpRequest与服务端通信
- 适合MVVM模式

```
addEventListener("message", function (evt) {  
    var xhr = new XMLHttpRequest();  
    //访问天气预报的API  
    xhr.open("GET", "http://www.help.bj.cn/apis/weather?id=101060101");  
    xhr.onload = function () {  
        | postMessage(xhr.responseText);  
    };  
    xhr.send();  
}, false);
```