

geminnoTM 高博应诺

InnoDB的应用

上节课回顾

- 索引
- 视图
- 函数和存储guoc
- 事务

课程目标

- 一：创建数据库
- 二：插入数据
- 三：删除数据
- 四：查询数据
- 五：更新数据

html5之indexeddb

indexdb

是HTML5-WebStorage的重要一环，是一种轻量级NOSQL数据库。相比web sql(sqlite)更加高效，包括索引、事务处理和健壮的查询功能。。

indexeddb

indexedDB特点：

- 一个网站可能有一个或多个 IndexedDB 数据库，每个数据库必须具有唯一的名称。
- 一个数据库可包含一个或多个对象存储。一个对象存储（由一个名称惟一标识）是一个记录集合。每个记录有一个键 和一个值。该值是一个对象，可拥有一个或多个属性。键可能基于某个键生成器，从一个键路径衍生出来，或者是显式设置。一个键生成器自动生成惟一的连续正整数。键路径定义了键值的路径。它可以是单个 JavaScript 标识符或多个由句点分隔的标识符。(有点像列数据库的特点)

indexdb

- IndexedDB中，几乎所有的操作都是采用了command->request->result的方式。比如查询一条记录，返回一个request，在request的result中得到查询结果。又比如打开数据库，返回一个request，在request的result中得到返回的数据库引用。

创建数据库

1、判断浏览器是否支持indexedDB

```
if (!window.indexedDB) {  
    alert('不支持');  
}
```

2、新建/删除数据库

```
var request=indexedDB.open('db_name');
```

```
var deleteDbRequest = indexedDB.deleteDatabase('test');
```

更新数据库

indexedDB标准建议，在初始化的时候创建表。以后每次打开浏览器，只需要check版本号，不需要第二次创建。(只有当版本变化或者第一次时采调用onupgradeneeded方法)

默认情况下，IndexedDB 数据库是没有版本号的。新版本的 IndexedDB 规范不支持调用setVersion()方法为数据库指定一个版本号（传入一个表示版本号的字符串）；代替的方法是在 open() 的时候传入表示版本号的字符串，同时使用onupgradeneeded 事件代替了 setVersion() 方法。onupgradeneeded 事件会在onsuccess 之前被调用。

创建数据库_demo

```
var request=indexedDB.open('LZHAN-2',v);
request.onerror=function(event){
    alert('构建数据库失败！'+event.target.errorCode);
};
request.onupgradeneeded=function(ev){ //onupgradeneeded
    alert('更新、创建数据库'+ev.oldVersion);
    db=request.result;
    if(db.objectStoreNames.contains('students')){
        db.deleteObjectStore('students');
    }
    var store = db.createObjectStore("student", {keyPath: "isbn"});
    var titleIndex = store.createIndex("by_sid", "sid", {unique: true});
    var authorIndex = store.createIndex("by_sno", "sno");

    // Populate with initial data.
    store.put({sid: "1", sno: "001",sname:'tom', isbn: 123456,other:"ceshi"});
    store.put({sid: "2", sno: "002",sname:'jack', isbn: 234567});
    store.put({sid: "3", sno: "003",sname:'rose', isbn: 34567800000000000000});
```

插入数据

IndexedDB	事务模式
模式	描述
readonly	提供对某个对象存储的只读访问，在查询对象存储时使用。
readwrite	提供对某个对象存储的读取和写入访问权。
versionchange	提供读取和写入访问权来修改对象存储定义，或者创建一个新的对象存储。

```
function insertData(){  
    var tx = db.transaction("students", "readwrite");  
    var store = tx.objectStore("students");  
    for (var i in userData) {  
        store.add(userData[i]);  
    }  
}
```

//注意：不能出现重复的数据；会结束事务进程

删除数据

//删除制定数据

```
function delData(){
    var tx = db.transaction("students", "readwrite");
    var store = tx.objectStore("students");
    var delRequest=store.delete('2'); //key
    delRequest.onsuccess=function(){
        alert('del success!!');
    }
}
```

//删除数据必须为根据主键删除

//删除所有数据

```
function clearAll(){
    var tx = db.transaction("students", "readwrite");
    var store = tx.objectStore("students");
    var clsRequest=store.clear();
    clsRequest.onsuccess= function () {
        alert('clear成功');
    }
}
```

更新数据

```
function updateData(){  
    var tx = db.transaction("students", "readwrite");  
    var store = tx.objectStore("students");  
    store.get('13').onsuccess=function(e){  
        var person= e.target.result;  
        person.sname='popup';  
        store.put(person).onsuccess=function(){  
            alert('更新成功');  
        }  
    }  
}
```

查询数据

```
function getData(){  
    var tx = db.transaction("students", "readwrite");  
    var store = tx.objectStore("students");  
    //使用事务查找  
    store.get('13').onsuccess=function(ev){  
        // '13' 为SID  
  
        //使用索引查找  
        //store.index('by_sid').get('13').onsuccess=function(ev){  
            var person=ev.target.result;  
            $('#div01').html(JSON.stringify(person));  
        };  
    }  
}
```

遍历数据

类似于关系数据库中游标的工作方式，IndexedDB 中的游标使您能够迭代一个对象存储中的记录。您还可以使用对象存储的索引来迭代记录。IndexedDB 中的游标是双向的，所以您可向前和向后迭代记录，还可以跳过非惟一索引中的重复记录。openCursor 方法可以打开一个游标。它接受两个可选的参数，其中包括范围和方向。

```
function getAllData(){
    var tx = db.transaction("students", "readwrite");
    var store = tx.objectStore("students");
    var req=store.openCursor();
    var s="";
    req.onsuccess=function(e){
        var cursor=e.target.result;
        if(cursor){
            var stu=cursor.value;
            var key=cursor.key;
            s=s+JSON.stringify(stu);
            $('#div01').html(s);
            cursor.continue();
        }
    }
}
```


Html5之manifest

离线应用

HTML5 引入了应用程序缓存，这意味着 web 应用可进行缓存，并可在没有因特网连接时进行访问。

应用程序缓存为应用带来三个优势：

- 离线浏览 - 用户可在应用离线时使用它们
- 速度 - 已缓存资源加载得更快
- 减少服务器负载 - 浏览器将只从服务器下载更新过或更改过的资源。

新建MANIFEST文件

新建文件

*.appcache

文件内容

```
CACHE MANIFEST
# 2015-02-21 v1.0.0
CACHE:
ad-01.jpg
demo.css
jquery.min.js
```

新建MANIFEST文件

manifest 文件可分为三个部分：

- CACHE MANIFEST - 在此标题下列出的文件将在首次下载后进行缓存
- NETWORK - 在此标题下列出的文件需要与服务器的连接，且不会被缓存
- FALLBACK - 在此标题下列出的文件规定当页面无法访问时的回退页面（比如 404 页面）

引入MANIFEST文件

引入文件

```
<html lang="en" manifest="index.appcache">
```

服务器配置

配置

Apache服务器进行配置的时候，需要找到{apache_home}/conf/mime.type这个文件，并在文件最后添加如下代码：

```
text/cache-manifest manifest
```

本节概要

ActionScript的应用

- 一：动作脚本入门
- 二：时间轴控制函数-gotoAndStop
- 三：使用行为-转到某帧停止
- 四：使用ActionScript实现鼠标跟随

ActionScript可以实现对动画播放的各种控制。通过为影片中的元件添加脚本，还可以实现更多丰富的动画效果。

课后任务

根据所学内容，制作一个小学网页加载界面，内容样式如下：

<http://vip.520wawa.com/schools/school86/welcome.html>

制作如上类似的Flash（整个welcome里面，定格在“Enter”）

大小为1366.768

质量最低需达到课堂上老师制作的水准

课后预习

网页设计基础知识
认识互联网
域名与空间
静态网页与动态网页
网站的类型
网页设计的原则
Dreamweaver介绍

Thanks