# HTML5 Web 存储

## 什么是 HTML5 Web 存储?

使用HTML5可以在本地存储用户的浏览数据。

## 浏览器支持

Internet ExplorerFirefoxOperaGoogle ChromeSafari

Internet Explorer 8+, Firefox, Opera, Chrome, 和 Safari支持Web 存储。

**注意:** Internet Explorer 7 及更早IE版本不支持web 存储.

## localStorage 和 sessionStorage

* localStorage - 没有时间限制的数据存储
* sessionStorage - 针对一个 session 的数据存储

在使用 web 存储前,应检查浏览器是否支持 localStorage 和sessionStorage:

if(typeof(Storage)!=="undefined")  
  {  
  // Yes! localStorage and sessionStorage support!  
  // *Some code.....*  
  }  
else  
  {  
  // Sorry! No web storage support..  
  }

# HTML5 Web SQL 数据库

以下是规范中定义的三个核心方法：

1. **openDatabase**：这个方法使用现有的数据库或者新建的数据库创建一个数据库对象。
2. **transaction**：这个方法让我们能够控制一个事务，以及基于这种情况执行提交或者回滚。
3. **executeSql**：这个方法用于执行实际的 SQL 查询。

## 打开数据库

我们可以使用 openDatabase() 方法来打开已存在的数据库，如果数据库不存在，则会创建一个新的数据库，使用代码如下：

var db = openDatabase('mydb', '1.0', 'Test DB', 2 \* 1024 \* 1024);

openDatabase() 方法对应的五个参数说明：

1. 数据库名称
2. 版本号
3. 描述文本
4. 数据库大小
5. 创建回调

第五个参数，创建回调会在创建数据库后被调用。

## 执行查询操作

执行操作使用 database.transaction() 函数：

var db = openDatabase('mydb', '1.0', 'Test DB', 2 \* 1024 \* 1024);

db.transaction(function (tx) {

tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');});

上面的语句执行后会在 'mydb' 数据库中创建一个名为 LOGS 的表。

## 插入数据

在执行上面的创建表语句后，我们可以插入一些数据：

var db = openDatabase('mydb', '1.0', 'Test DB', 2 \* 1024 \* 1024);

db.transaction(function (tx) {

tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (1, "菜鸟教程")');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (2, "www.runoob.com")');});

我们也可以使用动态值来插入数据：

var db = openDatabase('mydb', '1.0', 'Test DB', 2 \* 1024 \* 1024);

db.transaction(function (tx) {

tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');

tx.executeSql('INSERT INTO LOGS

(id,log) VALUES (?, ?'), [e\_id, e\_log];});

实例中的 e\_id 和 e\_log 是外部变量，executeSql 会映射数组参数中的每个条目给 "?"。

## 读取数据

以下实例演示了如何读取数据库中已经存在的数据：

var db = openDatabase('mydb', '1.0', 'Test DB', 2 \* 1024 \* 1024);

db.transaction(function (tx) {

tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (1, "菜鸟教程")');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (2, "www.runoob.com")');});

db.transaction(function (tx) {

tx.executeSql('SELECT \* FROM LOGS', [], function (tx, results) {

var len = results.rows.length, i;

msg = "<p>查询记录条数: " + len + "</p>";

document.querySelector('#status').innerHTML += msg;

for (i = 0; i < len; i++){

alert(results.rows.item(i).log );

}

}, null);});

## 删除记录

删除记录使用的格式如下：

db.transaction(function (tx) {

tx.executeSql('DELETE FROM LOGS WHERE id=1');});

删除指定的数据id也可以是动态的：

db.transaction(function(tx) {

tx.executeSql('DELETE FROM LOGS WHERE id=?', [id]);});

## 更新记录

更新记录使用的格式如下：

db.transaction(function (tx) {

tx.executeSql('UPDATE LOGS SET log=\'www.w3cschool.cc\' WHERE id=2');});

更新指定的数据id也可以是动态的：

db.transaction(function(tx) {

tx.executeSql('UPDATE LOGS SET log=\'www.w3cschool.cc\' WHERE id=?', [id]);});