



NextGen



Web



Bài 18

Lưu trữ Web trong HTML5

Mục tiêu

- Giải thích về lưu trữ web trong HTML5
- Giải thích về lưu trữ session
- Giải thích về lưu trữ cục bộ
- Giải thích về việc đánh chỉ mục DB API
- Mô tả về ứng dụng gốc
- Giải thích sự khác nhau giữa các ứng dụng gốc và ứng dụng HTML5
- Mô tả về ưu thế của ứng dụng gốc so với ứng dụng HTML5
- Các bước chuyển từ ứng dụng HTML5 sang ứng dụng gốc.

Giới thiệu

Theo truyền thống, trong vài thập kỷ qua, các ứng dụng web đã được sử dụng cookie để lưu trữ một lượng nhỏ thông tin trên máy tính của người dùng.

Cookie là một tập tin lưu trữ thông tin người dùng có liên quan có thể là tạm thời hoặc vĩnh viễn.

Một cookie có thể được tạo ra cho các chi tiết đăng nhập có thể được lưu trong một thời gian quy định trên máy tính của người dùng.

- Những hạn chế của cookie như sau:
 - Cookie làm chậm hiệu suất của ứng dụng Web, như chúng đi kèm với mọi yêu cầu HTTP
 - Cookie không thể được coi là phương tiện an toàn để truyền dữ liệu nhạy cảm
 - Cookie không thể lưu trữ lượng thông tin lớn vì chúng có một giới hạn kích thước của 4 KB
- W3C đã thiết kế một đặc điểm kỹ thuật có tên là Web lưu trữ API trong đó cung cấp một giải pháp lưu trữ dữ liệu trên máy khách phía

Lưu trữ Web trong HTML5

Là một đặc điểm kỹ thuật W3C và trình duyệt nhất định gọi nó là 'DOM storage'.

Cung cấp chức năng cho lưu trữ dữ liệu trên máy khách.

Lưu trữ dữ liệu có thể phục vụ cho cả nhu cầu thường xuyên tạm thời cũng như.

Cung cấp kiểm soát nhiều hơn các tập tin cookie truyền thống, và rất dễ dàng để làm việc với chúng.

Ban đầu là một phần của HTML5, nhưng bây giờ nó hiện diện trong một đặc điểm kỹ thuật riêng biệt và lưu trữ tối đa là 5 MB thông tin cho mỗi tên miền.

Lưu trữ Web so với Cookies

- Một số khác biệt quan trọng giữa các tập tin cookie và lưu trữ web như sau:

Cookie nhằm để đọc được trên phía máy chủ, trong khi lưu trữ web chỉ có sẵn trên phía máy khách.

Cookie được gửi cùng với mỗi yêu cầu HTTP đến máy chủ, trong khi dữ liệu lưu trữ web không được chuyển đến máy chủ.

Cookie dẫn đến chi phí băng thông và do đó dẫn đến chi phí cao, khi chúng được gửi đi với mỗi yêu cầu HTTP. Lưu trữ web được lưu trữ trên ổ cứng của người dùng, do đó không tốn kém gì để sử dụng.

Với các tập tin cookie, thông tin có thể được lưu trữ là 4 KB, trong khi với lưu trữ web, các dữ liệu có thể được lưu trữ tối đa 5 MB.

Lưu trữ Web với các trình duyệt

Lưu trữ web phụ thuộc vào trình duyệt cụ thể và vị trí nơi lưu trữ dữ liệu web được lưu trữ phụ thuộc vào trình duyệt.

Lưu trữ của mỗi trình duyệt là riêng biệt và độc lập, ngay cả khi nó có mặt trên cùng một máy.

HTML5 lưu trữ web được thực hiện tự nhiên trong hầu hết các trình duyệt web, vì vậy người ta có thể sử dụng nó ngay cả khi trình duyệt của bên thứ ba plug-in không có sẵn.

- Bảng sau liệt kê sự hỗ trợ của các trình duyệt khác nhau cho HTML5 lưu trữ web:

Trình duyệt	Phiên bản
IE	8.0+
Firefox	3.6+
Safari	4.0+
Chrome	5.0+
Opera	10.5+

Khám phá lưu trữ Web 1-2

Hai loại HTML5 lưu trữ web là cụ thể là, lưu trữ session và lưu trữ cục bộ.

Cả hai loại đều cho phép lưu trữ khoảng 5 MB dữ liệu cho mỗi tên miền.

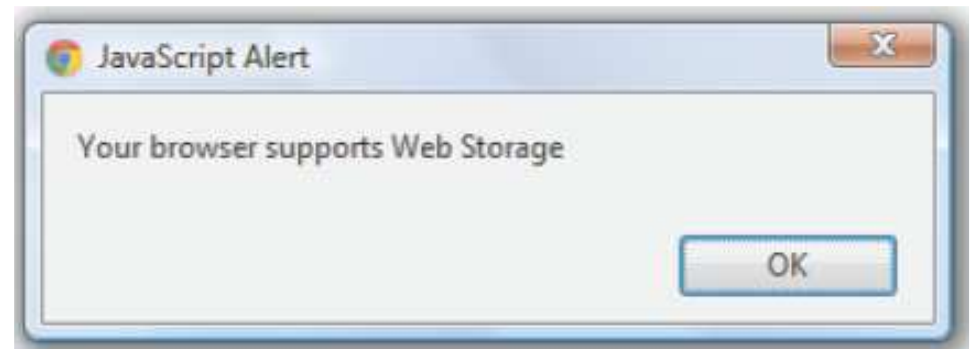
Để kiểm tra trình duyệt hỗ trợ HTML5 lưu trữ web, một thuộc tính có tên `localStorage` hoặc `sessionStorage` có sẵn như là một biến toàn cục cho các đối tượng cửa sổ.

Nếu không có hỗ trợ, thuộc tính `localStorage` hoặc `sessionStorage` sẽ không xác định.

Khám phá lưu trữ Web 2-2

- Ví dụ.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Support for Web Storage</title>
    <script>
      function checkSupport() {
        if (('localStorage' in window) && window['localStorage']
          !== null)
        {
          alert("Your browser supports Web Storage");
          return;
        }
        alert("Your browser does not support Web Storage");
      }
    </script>
  </head>
  <body onload="checkSupport();" >
  </body>
</html>
```



Lưu trữ Session 1-6

Theo dõi dữ liệu cụ thể cho một cửa sổ hoặc tab và loại bỏ nó ngay người dùng đóng tab (hoặc cửa sổ) mà anh / cô ấy đã làm việc.

Kéo dài toàn bộ thời gian của phiên giao dịch và do đó, không phải là liên tục.

Sử dụng các cặp khóa / giá trị được kèm theo trong dấu ngoặc kép.

Lưu trữ dữ liệu sử dụng key, trong khi dữ liệu được lấy bằng cách tham chiếu đến key.

Key là một chuỗi, trong khi giá trị được lưu trữ trong khóa có thể chứa bất kỳ loại dữ liệu nào như chuỗi, boolean, số nguyên, hoặc số thập phân. Không phụ thuộc vào loại dữ liệu được lưu trữ, nó thực sự được lưu trữ cục bộ như một chuỗi.

Lưu trữ và lấy dữ liệu của các loại cần sử dụng các chức năng để chuyển đổi chúng thành các kiểu dữ liệu thích hợp.

Lưu trữ Session 2-6

- Bảng dưới đây liệt kê một số ví dụ về cặp tên khóa / giá trị thuộc kiểu dữ liệu khác nhau.

Key	Value
Name	Sarah
book	C Programming
Email	info@me.com
car	Toyota Innova
age	28
uservalid	true

Lưu trữ Session 3-6

- Một số thao tác có thể được thực hiện với các đối tượng sessionStorage như sau:

Lưu trữ và lấy dữ liệu được sử dụng phương thức - SetItem () và getItem () để lưu trữ và lấy dữ liệu từ phiên lưu trữ tương ứng.

- **Để gán dữ liệu**

```
sessionStorage.setItem(key, value);
```

Trong đó,

key: Là khóa tham chiếu đến dữ liệu.

value: Là dữ liệu được lưu trữ.

Lưu trữ Session 4-6

- **Để nhận dữ liệu**

```
var item = sessionStorage.getItem(key);
```

Trong đó,

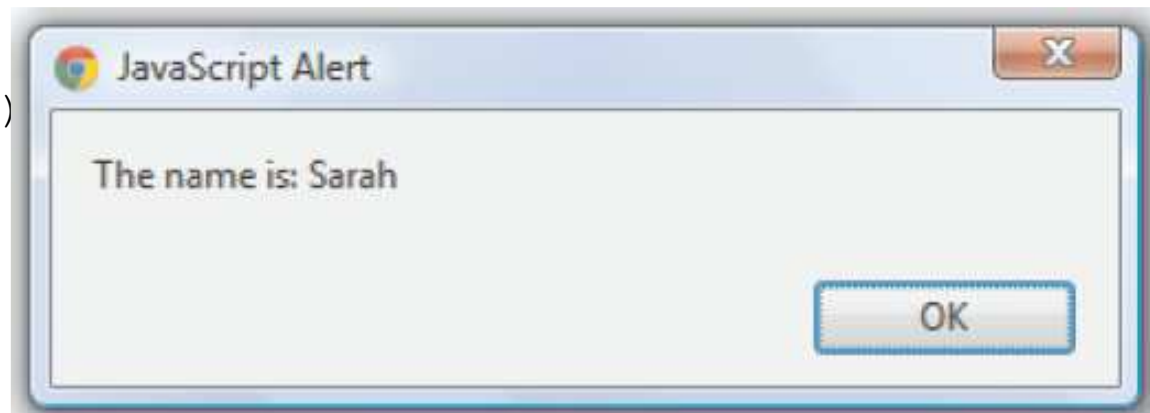
`item`: là biến mà sẽ lưu trữ giá trị.

`Key`: là khóa để nhận dữ liệu.

Lưu trữ Session 5-6

- Ví dụ

```
<!DOCTYPE html>
<html>
  <head>
    <title>Working with Session Storage</title>
    <script>
      function testStorage() {
        if (('sessionStorage' in window) && window['sessionStorage'] !==
null)
        {
          sessionStorage.setItem('name', 'Sarah');
          alert('The name is: ' + sessionStorage.getItem('name'));
        }
      }
    </script>
  </head>
  <body onload=" testStorage()"
  </body>
</html>
```



Lưu trữ Session 6-6

- **Xóa dữ liệu**

```
sessionStorage.removeItem(key);
```

- **Xoá tất cả dữ liệu**

```
sessionStorage.clear();
```

Lưu trữ cục bộ 1-4

Cho phép lưu dữ liệu trong thời gian dài trên máy tính của người dùng, thông qua trình duyệt.

Dữ liệu là liên tục và có thể được lấy khi người dùng truy cập các trang web một lần nữa.

Được sử dụng, nếu dữ liệu cần phải được lưu trữ trong hơn một phiên duy nhất.

Làm việc trong một cách thức tương tự như phiên lưu trữ.

Sử dụng các chức năng tương tự, chẳng hạn như `SetItem()`, `getItem()`, `RemoveItem()`, và `clear()`.

Lưu trữ cục bộ 2-4

- Ví dụ

```
<!DOCTYPE html>
<html>
  <title> Local Storage </title>
  <script>
    function store() {
      if (('localStorage' in window) && window['localStorage'] !== null) {
        var username = document.getElementById('username').value;
        localStorage.setItem('username', username);
      } else {
        alert ('your browser does not support storage');
      }
    }
    function cancel_store() {
      if (('localStorage' in window) && window['localStorage'] !== null) {
        localStorage.removeItem('username');
      } else {
        alert ('your browser does not support storage');
      }
    }
  </script>
</html>
```


Lưu trữ cục bộ 3-4

- ```
</script>
</head>
<body>
 <form method="get" action="success.html">
 Username: <input type="text" id="username" value="" size="20"
onblur="store()" />
 <input type="submit" value="Submit" />
 <input type="reset" Value="Cancel" onclick="cancel_store()" />
 </body>
</html>
```

# Lưu trữ cục bộ 4-4

- Ví dụ

```
<!DOCTYPE html>
<head>
 <title> Local Storage </title>
 <script>
 function print() {
 var username = localStorage.getItem('username');
 document.getElementById('lblMsg').innerHTML = 'Username: is '+
username+'';
 }
 </script>
</head>
<body onload="print()">
 <label id="lblMsg"></label>

</body>
</html>
```



# Lập chỉ mục Database API 1-3

Một cơ sở dữ liệu là một tập hợp có tổ chức của dữ liệu.

Cơ sở dữ liệu quan hệ lưu trữ dữ liệu dưới dạng bảng.

Một bảng bao gồm các hàng và cột được sử dụng để lưu trữ dữ liệu.

Sự hiện diện của dữ liệu từ một bảng ở dạng các bản ghi.

HTML5 đã giới thiệu một Web Storage API mới mà có thể lưu trữ Web databases cục bộ dưới máy của người dùng.

Web databases không giống như cơ sở dữ liệu quan hệ về chức năng.

# Lập chỉ mục Database API 2-3

Indexed Database API là một đặc điểm kỹ thuật còn được gọi là IndexedDB.

Nó là cơ bản một đối tượng lưu trữ có thể được sử dụng để lưu trữ và thao tác dữ liệu trên phía máy khách.

Đối tượng lưu trữ là cơ chế lưu trữ chính mà đối tượng lưu trữ trong cơ sở dữ liệu cục bộ quản lý bởi trình duyệt.

Nó cho phép để tạo ra một đối tượng lưu trữ của một loại hình cụ thể, trong đó đối tượng có thể được tiếp tục lưu trữ sử dụng JavaScript.

IndexedDB cho phép để tạo ra các ứng dụng web với khả năng truy vấn phong phú và có thể làm việc cả online và offline.

IndexedDB hỗ trợ hai loại API cụ thể là, đồng bộ và không đồng bộ.

API đồng bộ có thể được sử dụng với WebWorkers, trong khi API không đồng bộ có thể được sử dụng cho các ứng dụng Web.

# Lập chỉ mục Database API 3-3

IndexedDB API được thực hiện bằng cách sử dụng window.indexedDB đối tượng.

Trình duyệt thực hiện các đối tượng IndexedDB với tiền tố của mình. Ví dụ, trình duyệt Chrome sử dụng tiền tố webkit, trong khi Mozilla hỗ trợ-moz tiền tố.

- Bảng dưới đây liệt kê các trình duyệt hỗ trợ cho các API IndexedDB.

IE	Firefox	Chrome	Safari	Opera	iOS Safari
6.0	-	-	-	-	3.2
7.0	8.0moz	-	-	-	4.0-4.1
8.0	9.0moz	16.0webkit	5.0	-	4.2-4.3
9.0	10.0moz	17.0webkit	5.1	11.6	5.0
10.0	11.0moz	18.0webkit	6.0	12.0	-
-	12.0moz	19.0webkit	-	-	-

# IndexedDB API 1-2

- Một số các cấu trúc cơ bản của IndexedDB API như sau:

**Database** - Cơ sở dữ liệu IndexedDB gồm nhiều đối tượng lưu trữ. Mỗi cơ sở dữ liệu có chứa một tên xác định nguồn gốc của cơ sở dữ liệu và một số phiên bản trong đó xác định thời gian tồn tại của cơ sở dữ liệu.

**Object Store** - Là cơ chế chính để lưu trữ dữ liệu trong một cơ sở dữ liệu. Chúng được lưu trữ trong cơ sở dữ liệu dưới dạng bản ghi.

**Keys** - Mỗi bản ghi được lưu trữ trong cơ sở dữ liệu được xác định bởi một khóa duy nhất.

**Values** - Được các dữ liệu được lưu trữ trong bản ghi

**Key Path** - Là một chuỗi định nghĩa chỉ cho trình duyệt nên lấy khóa từ một giá trị. Khóa từ một giá trị có thể được lấy trong các đối tượng lưu trữ hoặc chỉ số.

**Index** - Được sử dụng khi các dữ liệu từ các đối tượng lưu trữ được lấy dựa trên một số giá trị khác mà không phải từ key.

**Transaction** – Hơn nữa, nhận dữ liệu trong một cơ sở dữ liệu được thực hiện bằng cách sử dụng giao dịch. Mỗi giao dịch có một chế độ, phạm vi, và một danh sách yêu cầu

# IndexedDB API 2-2

**Requests** - Hoạt động, chẳng hạn như đọc hoặc viết dữ liệu được thực hiện bằng cách sử dụng một yêu cầu. Mỗi yêu cầu chứa các thuộc tính, chẳng hạn như flag, đối tượng nguồn, kết quả, và lỗi.

**Cursor** - Là một cơ chế sử dụng để lấy nhiều bản ghi từ một cơ sở dữ liệu....

**Key Range** – Các bản ghi từ các đối tượng lưu trữ và các chỉ mục được lấy ra bằng cách sử dụng khóa hoặc dãy khóa. Một loạt khóa chính dùng để hạn dữ liệu trong miền giới hạn quy định dựa trên các khóa.

# Thực thi IndexedDB API 1-6

- Các bước để thực hiện các API IndexedDB trong một ứng dụng Web như sau:

Mở một cơ sở dữ liệu

Tạo một đối tượng lưu trữ

Bắt đầu một giao dịch

Thực hiện một số hoạt động cơ sở dữ liệu, chẳng hạn như thêm và lấy

Làm việc với các kết quả nhận được



# Thực thi IndexedDB API 2-6

- **Mở CSDL**

```
var indexedDB = window.indexedDB || window.webkitIndexedDB ||
window.mozIndexedDB || window.msIndexedDB;
var request = indexedDB.open("CompanyDB", 1);
request.onsuccess = function (event) {
 . . .
};
request.onerror = function (event) {
 console.log("IndexedDB error: " + event.target.errorCode);
};
```

# Thực thi IndexedDB API 3-6

- Cập nhật phiên bản CSDL

Sau khi cơ sở dữ liệu được mở ra, nó có thể được cấu trúc bằng cách cung cấp một số phiên bản giúp để thiết lập cơ sở dữ liệu.

- Ví dụ

```
var setVrequest = db.setVersion("1.99");
setVrequest.onsuccess = function(event) {
 . . .
}
```

# Thực thi IndexedDB API 4-6

- **Tạo đối tượng lưu trữ**

Cấu trúc của cơ sở dữ liệu IndexedDB tạo điều kiện cho việc lưu trữ nhiều đối tượng lưu trữ.

Đối tượng lưu trữ được tạo ra bằng cách sử dụng phương thức `createObjectStore()` mà chấp nhận hai đối số cụ thể là, tên lưu trữ và một đối tượng tham số.

```
var employeeData = [
 { name: "John Smith", email: "john@company.com" },
 { name: "Jill Patrick", email: "jill@company.com" },
 { name: "Rock Ethan", email: "rock@company.com" },
 { name: "Daniel Andrew", email: "daniel@company.com" }
];
var objectStore = db.createObjectStore("employee", {
 keyPath: "id", autoIncrement: true });
for (i in employeeData) {
 objectStore.put(employeeData[i]);
 alert("Record added");
}
```

# Thực thi IndexedDB API 5-6

- **Tạo một giao dịch**

Để thực hiện thao tác cơ sở dữ liệu, chẳng hạn như lấy dữ liệu từ đối tượng lưu trữ, IndexedDB cung cấp một đối tượng IDBTransaction.

Đối tượng này có thể được tạo ra trong ba chế độ cụ thể là, chỉ đọc, đọc-ghi, và ảnh chụp.

```
var trans = db.transaction(["employee"],
 IDBTransaction.READ_WRITE).objectStore("employee");
var request = trans.get(2);
request.onerror = function(event) {
 // Handle errors!
};
request.onsuccess = function(event) {
 // Do something with the request.result!
 alert("Name: " + request.result.name);
};
```

# Thực thi IndexedDB API 6-6

- **Mở một Cursor**

Cursor được sử dụng để lấy nhiều bản ghi từ một đối tượng lưu trữ.

Chúng có thể được sử dụng khi giá trị key path không hiểu. Chúng là một phần của một giao dịch và được mở cho một đối tượng lưu trữ cụ thể.

```
store = db.transaction("employee").objectStore("employee");
store.openCursor().onsuccess = function(event) {
 var cursor = event.target.result;
 if (cursor) {
 alert("Name for id " + cursor.key + " is " +
cursor.value.name);
 cursor.continue();
 }
};
```

# Những giới hạn của IndexedDB API

- Giới hạn thiết kế cho IndexedDB API được sử dụng để lưu trữ phía máy khách của dữ liệu như sau:

Quốc tế hóa phân loại giao dịch với sắp xếp dữ liệu chuỗi. Như cơ sở dữ liệu không tuân theo bất kỳ trật tự quốc tế để lưu trữ dữ liệu, sắp xếp quốc tế không được hỗ trợ bởi các API.

IndexedDB API không đồng bộ hóa cơ sở dữ liệu phía máy khách với cơ sở dữ liệu phía máy chủ.

IndexedDB API hỗ trợ truy vấn cơ sở dữ liệu phía máy khách, nhưng không hỗ trợ việc sử dụng các nhà điều hành, chẳng hạn như LIKE được sử dụng bởi Structured Query Language (SQL).

# Chuyển đổi HTML5 apps sang Native apps

Một ứng dụng nguyên bản còn được gọi là ứng dụng gốc là một chương trình ứng dụng được xây dựng để sử dụng nó trên một thiết bị cụ thể hoặc nền tảng.

Một ứng dụng gốc, khi so sánh với các ứng dụng web được cài đặt trên một thiết bị và có phản ứng nhanh hơn, bởi vì nó có một giao diện người dùng trực tiếp.

BlackBerry Messenger (BBM) là một ứng dụng gốc có sẵn trên các thiết bị di động BlackBerry.

# Sự khác nhau giữa Native Apps và HTML5 Apps

HTML5 ứng dụng web có thể truy cập và sử dụng trên bất kỳ thiết bị thông qua trình duyệt web tương tự như các trang web di động.

Các ứng dụng web có khả năng truy cập offline có nghĩa là người sử dụng không cần phải có kết nối mạng.

- Bảng sau liệt kê sự khác biệt giữa các ứng dụng gốc và các ứng dụng HTML5.

Native Apps	HTML5 Apps
Native apps chạy trên các thiết bị iOS và Android có thể được tải về hoặc mua từ các cửa hàng ứng dụng trực tuyến.	HTML5 Apps chạy trên một máy chủ Web, thường là trong một trình duyệt Web.
Native Apps sử dụng ngôn ngữ lập trình, chẳng hạn như Java cho các thiết bị Android và Objective C cho các thiết bị iOS.	Các nhà phát triển web sử dụng HTML, JavaScript và CSS. Họ cần phải có được các kỹ năng của Java và mục tiêu C để viết các ứng dụng bản địa.



# Ưu điểm của HTML5 Apps

- Ưu điểm chính của việc sử dụng HTML5 là để tạo ra các ứng dụng thực hiện trên một loạt các thiết bị dễ dàng. Một số lý do để phát triển các ứng dụng HTML5 như sau:

**Users cannot identify the differences** - Không thể xác định xem họ đang làm việc trên một ứng dụng HTML5 gốc lai hoặc một ứng dụng hoàn toàn có nguồn gốc hoặc một ứng dụng HTML5.

**Users adjust styles for devices** - Ứng dụng HTML5 có thể được xem trên bất kỳ thiết bị có chứa trình duyệt Web.

**Upcoming functionalities** - HTML5 không hỗ trợ tất cả các tính năng trên một thiết bị, nhưng nó đến với các chức năng mới.

**Improving Performance** – Nhiều nhà phát triển tìm hiểu phương pháp mới để cải thiện hiệu suất của web.

**Independent device** - Nếu các nhà phát triển muốn rằng ứng dụng của họ được sử dụng bởi một số lượng lớn người sử dụng, sau đó họ sẽ thiết kế và phát triển ứng dụng cho cả người dùng điện thoại di động cũng như người sử dụng máy tính để bàn.

**Developers are not locked in app stores** – Những nhà phát triển HTML5 không giới hạn cửa hàng ứng dụng. Thay vào đó, họ có thể tạo ra các ứng dụng và bán chúng như các trang web khác.

# Ưu điểm của Native Apps

- Lợi thế chính của các ứng dụng gốc trên các ứng dụng HTML5 là họ được nhanh hơn so với các ứng dụng HTML5. Các ứng dụng bản địa cung cấp nhiều lợi ích hơn so với các ứng dụng HTML5. Đây là như sau:

**Providing access to device hardware** - Không có sẵn API cho gia tốc, máy ảnh, hoặc bất kỳ thiết bị phần cứng khác cho các ứng dụng HTML5.

**Uploading Files** - Các ứng dụng gốc có thể truy cập hệ thống tập tin trong Android và một số tập tin trong iOS. Tuy nhiên, tập tin API HTML5 không hoạt động trên Android hay iOS.

**Push Notifications** - Các thông báo đẩy được gửi luôn luôn với một kết nối IP mở các ứng dụng trên thiết bị iOS.

**Accessing device files** - Các ứng dụng gốc giao tiếp với các tập tin trên các thiết bị, chẳng hạn như địa chỉ liên lạc và hình ảnh. Tuy nhiên, những tập tin này không thể được nhìn thấy từ HTML5 ứng dụng.

**Superior graphics than HTML5** - HTML5 có một yếu tố cavans, nhưng nó sẽ không tạo ra một trải nghiệm 3D đầy đủ.

**Offline access** - HTML5 cung cấp quyền truy cập vào các ứng dụng web offline. Tuy nhiên, một ứng dụng bản địa được lưu trữ trên máy tính địa phương, do đó người dùng không cần truy cập vào trang web để làm việc với các ứng dụng.

# Tổng kết

- Web lưu trữ là một đặc điểm kỹ thuật W3C cung cấp chức năng để lưu trữ dữ liệu trên phía máy khách cho cả hai nhu cầu thường xuyên.
- Các ứng dụng web HTML5 sử dụng lưu trữ web để thực hiện phía khách hàng lưu trữ liên : lưu trữ phiên và lưu trữ địa phương cũng như tạm thời.
- Lưu trữ phiên theo dõi dữ liệu cụ thể cho một cửa sổ hoặc tab.
- Các SetItem () và getItem () phương pháp được sử dụng để lưu trữ và lấy dữ liệu từ phiên lưu trữ.
- Lưu trữ địa phương cho phép để lưu dữ liệu trong thời gian dài trên máy tính của người dùng, thông qua trình duyệt.
- IndexedDB API về cơ bản là một đối tượng lưu trữ có thể được sử dụng để lưu trữ và thao tác dữ liệu trên phía máy khách.
- Một ứng dụng bản địa còn gọi là ứng dụng gốc là một chương trình ứng dụng được xây dựng cho một thiết bị cụ thể hay nền tảng.