

HTML5

HTML



JavaScript APIs

HTML5

- > Selectors
- > Drag & Drop
- > File reader API
- > Canvas
- > Storage
- > Offline Apps
- > Workers

Selector APIs

Methods for selecting and testing elements based on match of a given selector.

Two types Selector methods available

- > `querySelector()`: the first element which matches the specified DOM string
- > `querySelectorAll()`: returns all the elements which match the specified DOM string

Drag & Drop

- > HTML5 Drag & Drop lets you do basic drag & drop operations with a lot less JavaScript code.
- > `draggable = true`
- > events: `dragstart`, `drop`, etc
- > `event.transferData`.

Drag & Drop

- > Other events: dragenter, dragover, dragleave, dragend

Canvas

- > Container for graphics
- > Dynamic, scriptable rendering of 2D images
- > Uses JavaScript to draw
- > Resolution-dependent bitmap.

Demo - Canvas

```
<canvas id="example" width="200" height="200">
```

This text is displayed if your browser does not support HTML5 Canvas.

```
</canvas>
```

```
var example = document.getElementById('example');  
var context = example.getContext('2d');  
context.fillStyle = "rgb(255,0,0)";  
context.fillRect(30, 30, 50, 50);
```



Storage

- > Client Side Storage
 - > Cookies is the method used
 - > They are small
 - > Only 4 kilobytes of storage each
 - > Less for data-rich web apps.

Storage

- > Web Storage
 - > Most stable solution
 - > Stores 5 megabytes of data
 - > Two types of storage methods available
 - > Local storage : stores data with no expiration date
 - > Session storage: stores data for one session.

Demo

"detectionSpan"

Your browser supports both local and session storage

Enter key:

Enter value:

Set Key

setKey()

Local Storage

"localTable"

Key	Value	
1234	12345	<input type="button" value="Remove"/>
123	456	<input type="button" value="Remove"/>
lost	in tokyo	<input type="button" value="Remove"/>
hcl	world	<input type="button" value="Remove"/>

clearLocalKeys()

Session Storage

Key	Value	
lost	in tokyo	<input type="button" value="Remove"/>

"sessionTable"

clearSessionKeys()

Offline Applications

> Enable

<html manifest="my. appcache">

> my.manifest

**CACHE MANIFEST
2012-02-21 v1.0.0**

**index.html
images/shade.jpg
images/bin.jpg
style.css**

Offline Applications

- > Cache
 - > On load will hit my.manifest
 - > Change manifest: trigger reload
 - > Force through javascript

```
function onUpdateReady() {  
    alert('found new version!');  
}
```

Offline Applications

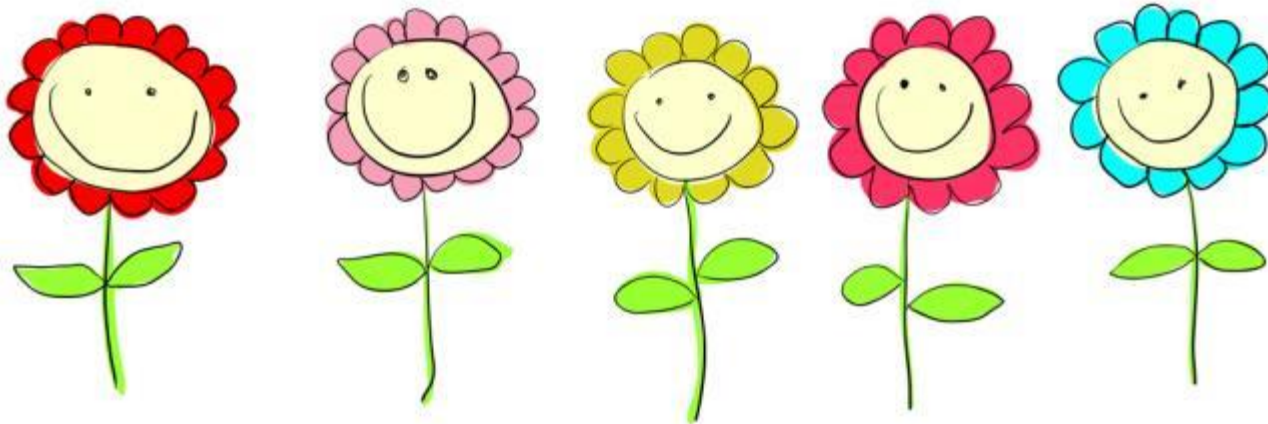
- > Updating the Cache
 - > The user clears browser data
 - > The manifest file is modified
 - > The app cache is programmatically updated.

Web Workers

- > JavaScript is a single thread process
- > Web worker creates a new thread
- > Runs without interrupting browser processing.

HTML5 – Questions





Thank you!