


<b>MTA-HTML5</b> App-Entwicklung mit HTML5		<b>Milestone 5</b> Moderne HTML5-Apis einsetzen		<b>OSZ</b>  <b>IMT</b>	
Name:		Datum: 13.12.2015		Klasse:	Blatt Nr.: 1/2
					Lfd. Nr.:

## Touch API / Safari Multitouch API

The touch events can be used [with various mobile browsers](#). However, gesture events are currently only supported in Webkit browsers (Safari). They occur any time a user touches the screen with two fingers.

- **touchstart:** triggered every time a finger is placed on the screen
- **touchend:** triggered every time a finger is removed from the screen
- **touchmove:** triggered as a finger already placed on the screen is moved across screen

A **touch event** contains three lists of points:

- **touches** every finger currently touching the screen.
- **changedTouches** information for every finger involved in the event. The **targetTouches** touch points currently in contact with the screen and whose **touchstart** event occurred within the same node (inside same target element as the current target element)

The three **gesture events** are triggered on every two-finger gesture. The sequence is quite simple:

**gesturestart** (touching the screen) → **gesturechange** (moving around) → **gesturechange** → ... → **gestureend** (lifting the fingers).

A gesture event has to properties:

- **scale:** indicates the amount of two-finger pinch zooming that occurred
- **rotation:** indicates the amount of two-finger rotation that occurred

## Geolocation API

The Geolocation API defines an interface that provides a device's location, usually using latitude and longitude coordinates. It exposes the latitude and longitude to JavaScript in a Web page using the **geo-location** object by the following methods, the first two accepting a callback function as parameter:

- **getCurrentPosition** get the device's current geographic position
- **watchPosition:** watches the device's position as it changes over time and generate an event if a change occurs
- **clearWatch** stops the watch

*Geodetic* data provides raw location data, such as longitude and latitude, or meters. *Civic data* is location data that's more easily understood by humans, such as a map or an address like 637 Park Street.

## Web Storage API


The WebStorage API allows client-side method for saving session information locally within the memory, using the following to methods:

- **localStorage** saves larger amounts of data from session to session (persistent data)
- **sessionStorage** keeps data only for one session (until the browser is closed).

The data stored in key/value pairs. The **sessionStorage** is isolated to a specific window or browser tab and stores temporary data during an HTTP session that occurs in a single window or tab. Multiple windows or tabs can maintain their own session data.

Values are added using **setItem()** with a named key. Existing keys will silently be overwritten. Calling **getItem()** returns a value. Called with a non-existent key, it will return null rather than throw an exception. Like other JavaScript objects, you can treat the **localStorage** and **sessionStorage** object as an associative array, e.g. `int foo = localStorage["foo"]` or `sessionStorage["bar"] = 42`.



<b>MTA-HTML5</b> App-Entwicklung mit HTML5	<b>Milestone 5</b> Moderne HTML5-Apis einsetzen	
Name:	Datum: 13.12.2015	Klasse:
		Blatt Nr.: 2/2
		Lfd. Nr.:

## DeviceOrientation API

The DeviceOrientation API uses a device's accelerometer. It specifies three events

- **deviceorientation** fired when a significant orientation occurs
- **compassneeds\_calibration** fired if the compass needs calibration
- **devicemotion** fired regularly with information about the motion of the device

The **DeviceOrientation** Event is based around three pieces of data, **alpha** (direction the device is facing according to the compass), **beta** (angle in degrees the device is tilted front-to-back) and **gamma** (angle in degrees the device is tilted left-to-right).

The **DeviceMotion** Event uses the properties **acceleration** (in m/s<sup>2</sup> for each of the x, y, and z axes), **accelerationIncludingGravity** (see acceleration) and **rotationRate** (in deg/s around each axes, as a, b, g) and interval (sampling time in milliseconds)

## Other APIs

**Web Workers** are scripts that run in the background, performing calculations or other actions that allow for a more responsive user interface. They objects run in isolated threads and cannot access the DOM. Communication is performed via messages with the HTML document's JavaScript code.

**WebSockets** offer full-duplex communication through a single socket over the Internet. Main events: **onopen** (socket opens), **onmessage** (message has been received) and **onclose**: (socket closes). The **send()** method can send a **String** or a **Blob** (binary large object) value.

**File API** uploads files from local storage to a remote server without a plug-in. Interfaces are **File** (read-only informational attributes, reads in the file as a URL), **FileList** (array-like sequence of **File** objects; also if folders are selected), **Blob** (binary data) and **FileReader** (methods to read and display a file)

**Camera API** uses a capture attribute with the input element to capture data from cameras, camcorders, webcams, microphones, and so on. Generic code that uploads an image from a device's camera:

```
<input type="file" accept="image/*" capture="camera" id="capture" /
```

## Sources

- <http://mobiforge.com/design-development/html5-mobile-web-device-orientation-events>
- <http://diveintohtml5.info/storage.html>
- Lesson 10 Slides: JavaScript Coding for the Touch Interface, Device and Operating System Resources, and More

