Outline

- The history of web on the mobile device
- HTML on mobile devices
  - HTML features of special interest
- CSS on mobile devices
  - Media queries
  - WebKit extensions
  - …
- JavaScript on mobile devices
  - Platform detection
  - Focus and scroll management
  - Libraries and platform extension
  - Access to sensor data

Web on mobile devices

- The first generation
  - Wireless Access Protocol (WAP1)
  - HDML/WML as markup languages (non-HTML)
  - WAP push
- Second generation
  - WAP 2, 3G, WiFi support, AJAX
  - HTML 4/5
  - Geolocation
  - Offline apps

The viewport

- The area of the browser in which the page fits
- Browsers may assume a standard page with
- Controlling the page scaling by defining the viewport in a meta tag, e.g.

```
<meta name="viewport"
      content="width=device-width,
      user scalable=no" />
```

Inline images

- Page loading time can be reduced by keeping all data in a single file
- The data URI is a mechanism for embedding image data directly in the `src` attribute of the img `tag`, e.g.

```
Data:img/png;base64,iVBOR…YII=
```

Linking to phone features

- Many mobile devices are also phones
- These may be able to set up phone calls when the user clicks a link (if the user approves), e.g.

```
<a href="tel:+4761135218">Call Øivind</a>
```

CSS media queries

- The CSS media attribute is not sufficient for selection a stylesheet based on devices
- CSS3 defines media queries for finer granularity controll, e.g.

```
media="only screen and
      (max-device-width: 480px)"
```

Text overflow

- Long titles may be broken over several lines because of the small screen width
- A CSS trick will clean this up

```
#header h1 {
   /* Other style settings */
   max-width: 160px;
   overflow: hidden;
   white-space: nowrap;
   text-overflow: ellipsis;
}
```

The rgba() function

- Information may be stacked on small screens
- Transparency may make the design more appealing
- The a in the CSS rgba() function defines the opacity
  - a=0 => fully transparent
  - a=1 => opaque color
  - 0 < a < 1 partly transparent
  - e.g.

```
background: rgba(0,0,0,0.3);
```

WebKit extensions

- WebKit added many extensions to CSS
- Several are already included in the standard, more may follow
- Compatibility is not perfect across platforms
- Sample extensions
  `-webkit-border-radius`: rounded-corner box
  `-webkit-box-shadow`: shadow for block element
  `-webkit-columns`: width and count of columns
  `-webkit-border-image`: border image
  `-webkit-text-stroke`: color for text outline
  `-webkit-text-fill`: color for text inside

Platform detection using JavaScript

- Using JavaScript on the device
  - Interacting with the navigator object and its properties
    - appName
    - appVersion
    - mimeTypes
    - platform

Focus and scroll management

- Scrolling can be inconvenient on many mobile devices but page code may prescroll
  - `window.scrollTo()`
- Similarly, focus can be set calling
  - `focus()` on the given DOM element

JavaScript libraries

- JavaScript – and DOM – implementations may vary in different browsers/browser versions
  - Painful to get the code to work on every browser
- JavaScript libraries hide browser differences and simplifies the development of web apps
  - **jQuery** – a much used but large JavaScript library
  - **jQTouch** – an open source jQuery plugin for mobile web development

Platform extensions

- JavaScript API extensions required for accessing some device features, e.g.
  - Messaging
  - Address book management
  - Camera
  - Gallery
  - Compass
  - Accelerometer
- PhoneGap is a cross-platform extension
  - Open source
  - Based on HTML5

PhoneGap feature matrix

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 5.x	Blackberry OS 6.0+	WebOS	Windows Phone 7	Symbian	Bada
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	X	✓	✓	X	✓
Contacts	✓	✓	✓	✓	✓	X	✓	✓	✓
File	✓	✓	✓	✓	✓	X	✓	X	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	X	X	✓	X	X
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓	X

The W3C geolocation API

- Currently a candidate W3C recommendation
- Getting a position

```
navigator.geolocation.getCurrentPosition(
        onSuccess, onError);
```

- Tracking the position

```
wId = navigator.geolocation.watchPosition(
        onSuccess, onError);
   // …
navigator.geolocation.clearWatch (wId);
```

- Where `onSuccess` and `onError` are callback functions

The google maps API

- Google maps API v3
  - Rich and dynamic API
  - Currently only on iPhone and Android devices
  - Sample code

```
new google.maps.Map (mapElRef, options);
```

The getUserMedia API

- Direct access to a media stream

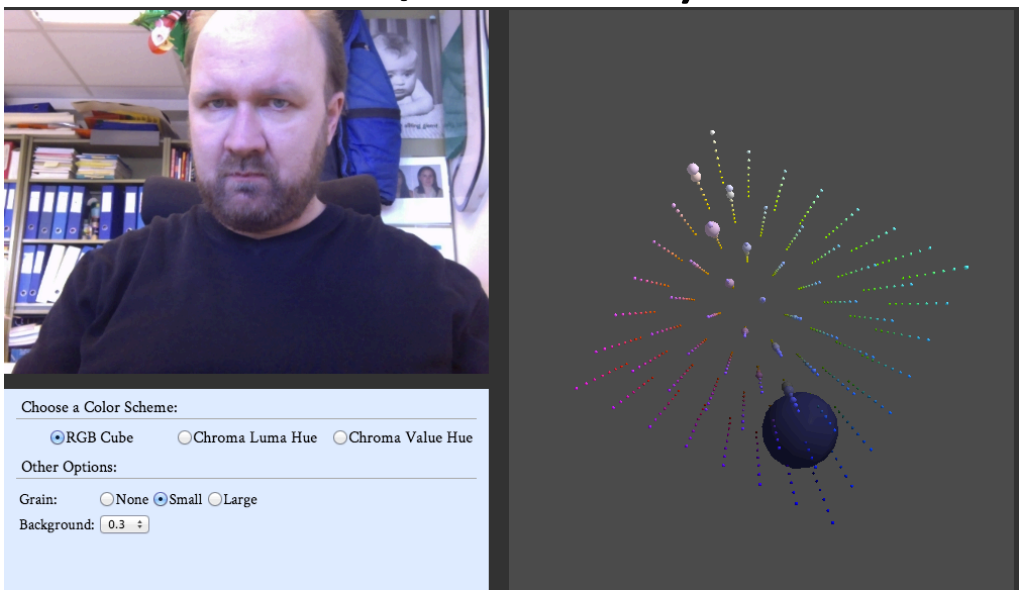
```
function onSuccess (stream) { video.src = stream; }

if (navigator.getUserMedia) {
    navigator.getUserMedia (“video”, onSuccess,
                            onError);
}
```

- Can be used for augmented reality apps

```
video = document.getElementById(“video”);
canvas = document.getElementById(“output”);
ctx = canvas.getContext(“2d”);
ctx.drawImage(video, 0, 0, video.width,
video.height,
                  0, 0, canvas.width,
canvas.height);
```

The getUserMedia API

- Currently a W3C working draft
- Implemented in Firefox, Chrome and Opera (caniuse.com/stream)



http://shinydemos.com/getusermedia/

Conclusion

- HTML and CSS constructs from desktop apps work mostly well in mobile browsers also
- Some HTML and CSS constructs are especially useful in mobile browsing
- JavaScript: Libraries, libraries, libraries
  - jQuery
  - jQTouch
  - PhoneGap