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 width
- 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 one 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:+4761135184">Call Rune</a>
```

CSS Media Queries

- The CSS `media` attribute is not sufficient for selecting a stylesheet based on devices
- CSS3 defines media queries for finer granularity control, 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 may be included in standard CSS
- 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 stroke

Platform Detection Using JavaScript

- Using JavaScript on the device:
  - Interacting with the navigator objects and its properties, e.g.,:
    - 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		OS 4.6-7	OS 5.x	OS 6.0+			
ACCELEROMETER	✓	✓	✓	✗	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	✗	✓	✓	✗	✗	✓
COMPASS	✗	✓	✓	✗	✗	✗	✗	✗	✗
CONTACTS	✓	✓	△	✗	✓	✓	✗	✓	✓
FILE	✗	✗	✓	✗	✓	✓	△	✗	✗
GEO LOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA (AUDIO RECORDING)	△	△	✓	✗	✗	✗	✗	△	✗
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓	✗
NOTIFICATION (VIBRATION)	✓	✓	✓	✓	✓	✓	✗	✓	✓
STORAGE	✓	✓	△	✗	△	✓	✓	✗	✗

The W3C Geolocation API

- Currently a candidate W3C recommendation
- Getting the position:

```
navigator.geolocation.getCurrentPosition(
                    successCBF, errorCBF);
```

- Tracking the position:

```
wId = navigator.geolocation.watchPosition(
                    successCBF, errorCBF);
// …
navigator.geolocation.clearWatch(wId);
```

- Where `successCBF` and `errorCBF` are callback functions

The Google Maps APIs

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

```
new google.mapsMap(mapElRef, options);
```

The getUserMedia API (1)

- Direct access to a media stream:

```
if (navigator.getUserMedia) {
  navigator.getUserMedia("video", successCallback,
                         errorCallback);
  function successCallback(stream) { video.src = stream; }
  function errorCallback() { ... }
}
```

- 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 (2)

- Currently a W3C Editor's Draft only
- Experimental implementation in the Opera Labs browser
- Augmented reality demo:

http://people.opera.com/brucel/articles/magic-html5-moustache.html

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