CS 641, Haik Sahakian

Mobile Web Development

Mobile JavaScript

Part 7

Mobile JavaScript

Overview of Features

Differences with Desktop

- 5 times slower
- Still very capable (face recognition)
- But slow (Blackboard would not work well)
- Beware of desktop library assumptions on mobile. Nested containers can trigger too many resize events for mobile, while OK for desktop.

Testing on Mobile

- Test early on mobile, rather than at the end of a project
- Use your laptop's IP address on your phone to hit your laptop web server
- Cool new features tend to have spotty browser support :)

Available Language Features

- Check the connection type with navigator.connection. Navigator.onLine tells you if the device is connected.
- Camera API with the input tag and createObjectURL()
- Vibration with window.navigator.vibrate()
- Low support for vibrate(). https://developer.mozilla.org/en-US/docs/Web/API/Navigator/vibrate.

Mobile Events

- Touch events: touchstart, touchmove, touchend, touchcancel.
- Gesture events: gesturestart, gesturechange, gestureend.
- Orientation events: orientationchange, deviceorientation.
- Motion events: devicemotion.

Available HTML Features

- The tel: and sms: protocols
- Disabling automatic telephone linking with the format-detection meta tag.
- Specify icons at different sizes with the apple-touch-icon meta tag.
- Turn off autocomplete, autocorrect, and autocapitalize with these attributes on an input tag.
- Camera control with the input tag with type set to “file”.

Available CSS Features

- Add momentum to scrolling with the “-webkit-overflow-scrolling: touch” style.
- Combines with the Overflow CSS class, this allows easy creation of scrollable regions on mobile.

Latest Features

- Language changes: Shadow DOM, promises, and Object.observe.
- Configure zooming with touch-action CSS style

Hands-On

The Window Object

- Built-in objects add the web into JavaScript. Otherwise it's just a language.
- The window and document objects are the most important built-in objects.
- Window handles the window itself, document handles what's inside it.
- Let's try using window.open().

The Document Object

- First let's get a reference to an existing tag in a page with getElementById().
- Then let's create a new DIV inside an existing one programmatically.

Basic Events

- Let's move the box we created around the browser window, using a JavaScript timer to trigger the movement.

Keyboard Events

- Let's move the box we created when the user hits the arrow keys.

The Canvas Object

- Let's draw a line inside a canvas:

```
c.beginPath();

c.moveTo(0, 0);

c.lineTo(666, 666);

c.stroke();
```

Questions?

hsahakian@pace.edu