Webvanta

Designing for the Mobile Web
Lesson 3: HTML5 Web Apps

Michael Slater, CEO
Andrew DesChenes, Dir. Services
course-support@webvanta.com
888.670.6793

www.webvanta.com

Welcome!

- Four sessions
  - 1: The Mobile Landscape
  - 2: Device Constraints and Opportunities
  - **3: HTML5 Web Apps**
  - 4: Native Apps

Questions

- Ask questions via chat during the live course
- Ask in the discussion group
- Email us
  - *Please use the discussion group* for non-private questions
- Course Portal
  - www.online-web-courses.com

Michael Slater
Cofounder & CEO

Andrew DesChenes
Director of Services

Webvanta

Topics for This Lesson

- (Sort of) a cross-platform solution
- Responsive design
- Using jQuery Mobile to quickly build app-like sites
- HTML5 web app features
- Geolocation, camera access, and more

HTML5 Web Apps

- There is a (partial) way out of the multi-platform nightmare!
  - *Choose the web as the platform*
  - HTML5, CSS3, JavaScript
- Limited access to devices and sensors
- Deliver through browser, or use PhoneGap to turn it into a "native" app
- No icon on phone, unless user creates one

Only a Partial Way Out...

- iOS devices are very consistent and frequently updated
- Android devices are diverse
  - Users can install their own browsers, and they often don't get updated
    - Can't reliably predict what features the browser will have
  - Default browser is buggy
  - Need to deal with variety of screen resolutions and pixel densities

Responsive Design

- Scale to any screen size
- Use CSS media queries and JavaScript to adapt to different screen sizes
- Examples:
  - www.bostonglobe.com
  - http://mediaqueri.es
- *Responsive Web Design* from A Book Apart
  - www.abookapart.com/products/responsive-web-design

Responsive Design Basics

- One HTML page serves all devices
- Use percentage widths for grid and images
  - Flexible grid scales to screen
  - Browser scales images
- Use media queries to deliver different CSS depending on screen width, to selectively hide or change elements
- Frameworks available to help you get started
  - http://www.getskeleton.com
    http://stuffandnonsense.co.uk/projects/320andup/

Media Queries

- Use within your CSS files

```
@media only screen and (max-device-width: 480px) {
   (css rules here to be used for phones only)
}
```

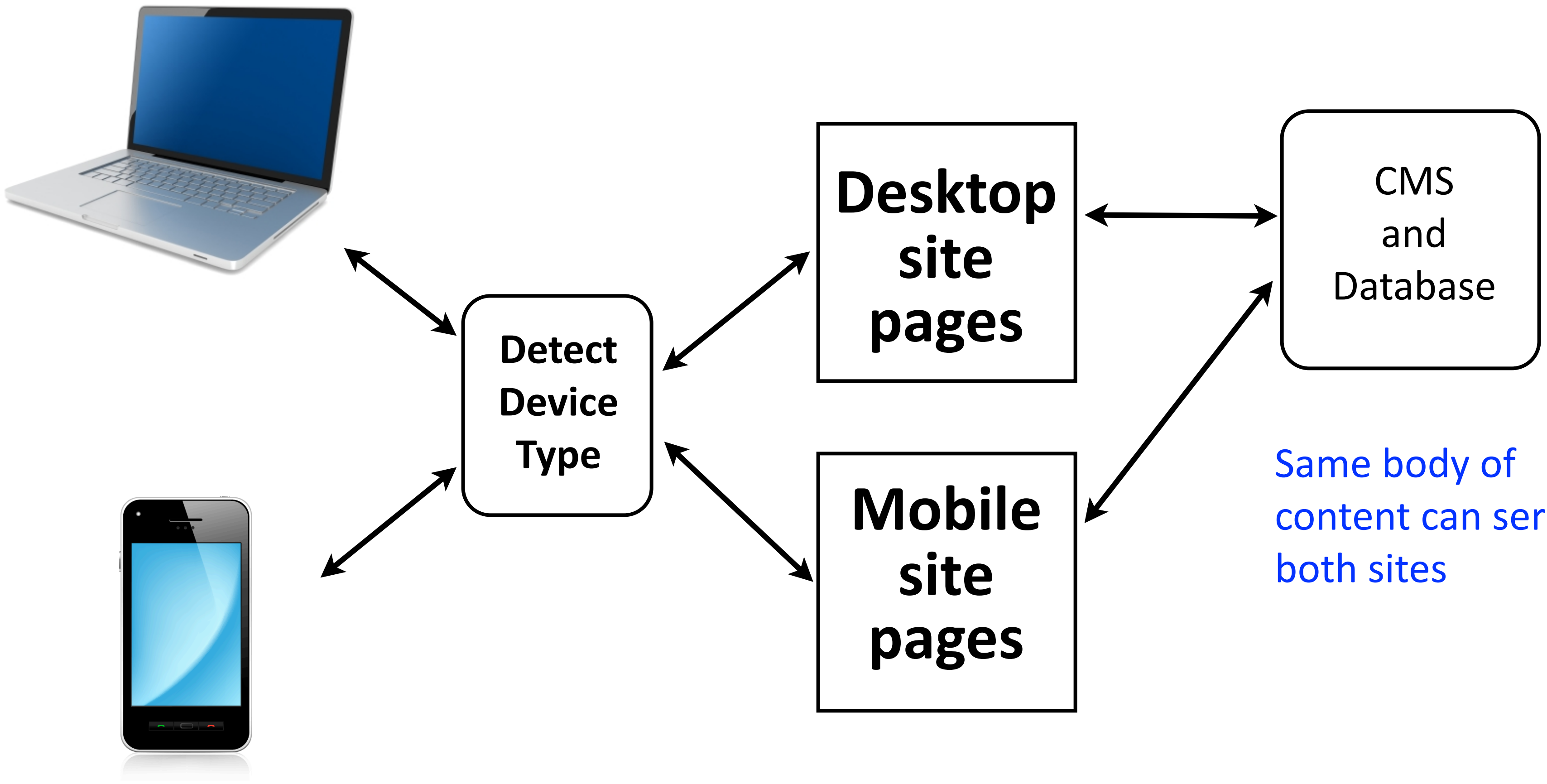
- Use to selectively load CSS files

```
<link rel="stylesheet" type="text/css"
media="only screen and (max-device-width: 480px)"
href="phone.css" />
```

Responsive Design Code Example

- Download the code files for this lesson
- Folder "responsive"
  - responsive.html is the HTML file
  - responsive.css is the basic CSS for the page
  - media-queries.css has the media queries
- Start with a browser at least 1000 px wide, and gradually reduce the width to see the changes

The Alternative: Separate Pages



Same body of content can serve both sites

Detecting Mobile Devices

- Responsive design allows a single set of pages to serve all devices
- If you *are* using separate pages, then you need a way to direct mobile devices to different pages (and maybe a different server)
  - Server-side detection can use request headers to detect mobile devices and serve different pages to them
  - JavaScript can examine the User Agent string to detect mobile devices
    - See http://detectmobilebrowsers.com/
  - Note: reliably separating phones from tablets is challenging
    - See http://www.webvanta.com/post/789079

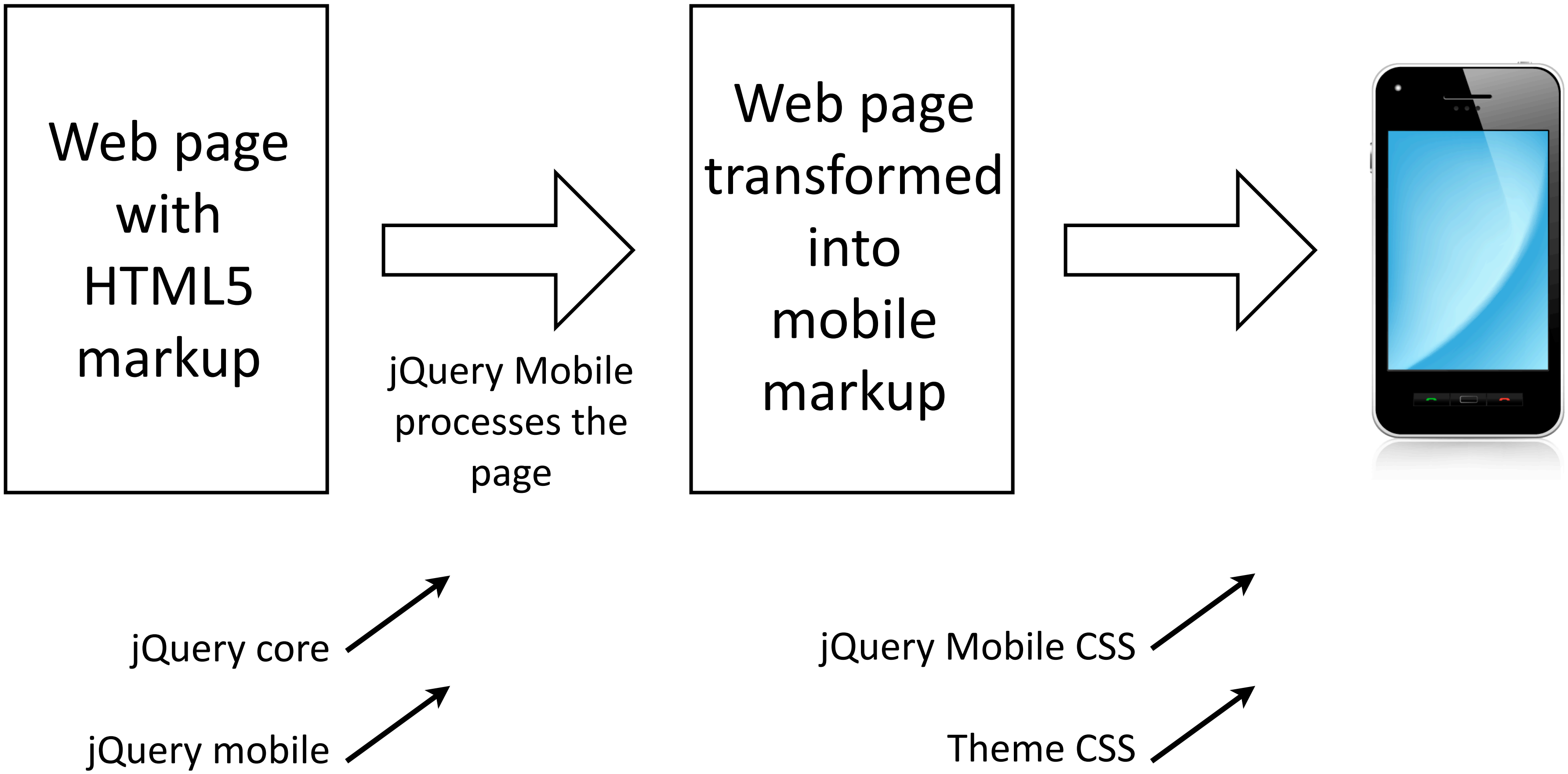
Building Mobile Web Pages

- They're just web pages
  - Use all the usual HTML, CSS and JavaScript
- A few mobile-specific issues
  - Viewports, as discussed last lesson
- Can depend on advanced HTML5 and CSS3 features
- Mobile web frameworks provide mobile-optimized, app-like interactions without having to design and code everything yourself

jQuery Mobile Framework

- Builds on top of regular jQuery library
- Not just interactive effects, but a complete framework including CSS and images
- Uses HTML5 data attributes to annotate markup
- JavaScript runs after HTML is loaded and transforms it into a mobile page
- Polished visual design, easily customized within limits
  - http://jquerymobile.com/themeroller/

Displaying a Page with JQM



Web page with HTML5 markup

jQuery Mobile processes the page

Web page transformed into mobile markup

jQuery core

jQuery mobile

jQuery Mobile CSS

Theme CSS

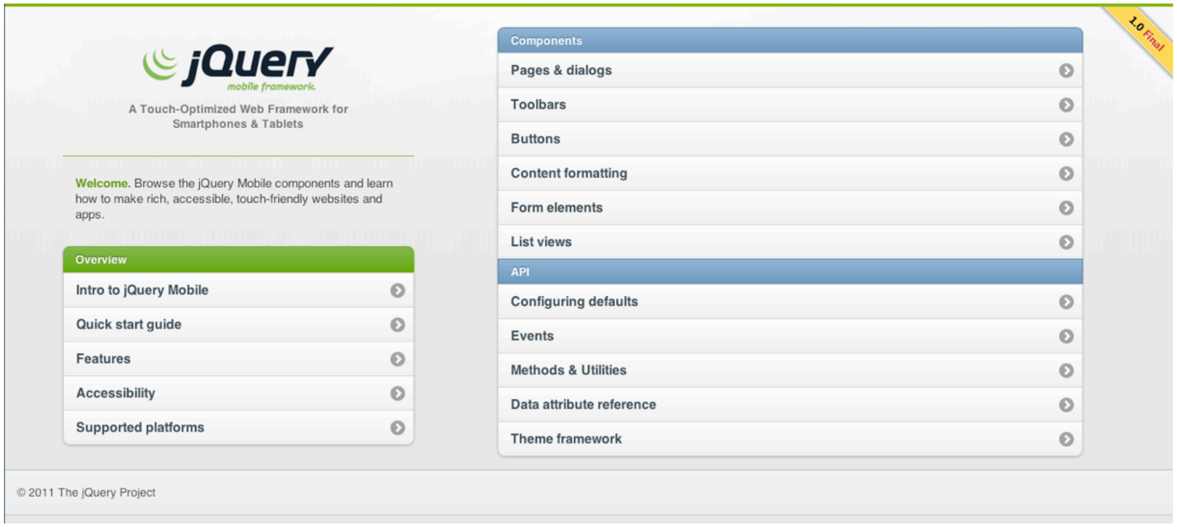
jQuery Mobile vs. HTML/CSS

- jQuery Mobile Advantages
  - Enables quick creation of polished sites
  - Interactions are mobile-optimized without additional effort and have app-like behavior
- jQuery Mobile Disadvantages
  - New paradigms take some getting used to
  - Harder to provide a completely customized visual design than with standard HTML/CSS

jQuery Mobile Docs

www.jquerymobile.com

- Books tend to be either out-of-date or not-yet-released



A Touch-Optimized Web Framework for Smartphones & Tablets

Welcome. Browse the jQuery Mobile components and learn how to make rich, accessible, touch-friendly websites and apps.

Overview
- Intro to jQuery Mobile
- Quick start guide
- Features
- Accessibility
- Supported platforms

Components
- Pages & dialogs
- Toolbars
- Buttons
- Content formatting
- Form elements
- List views

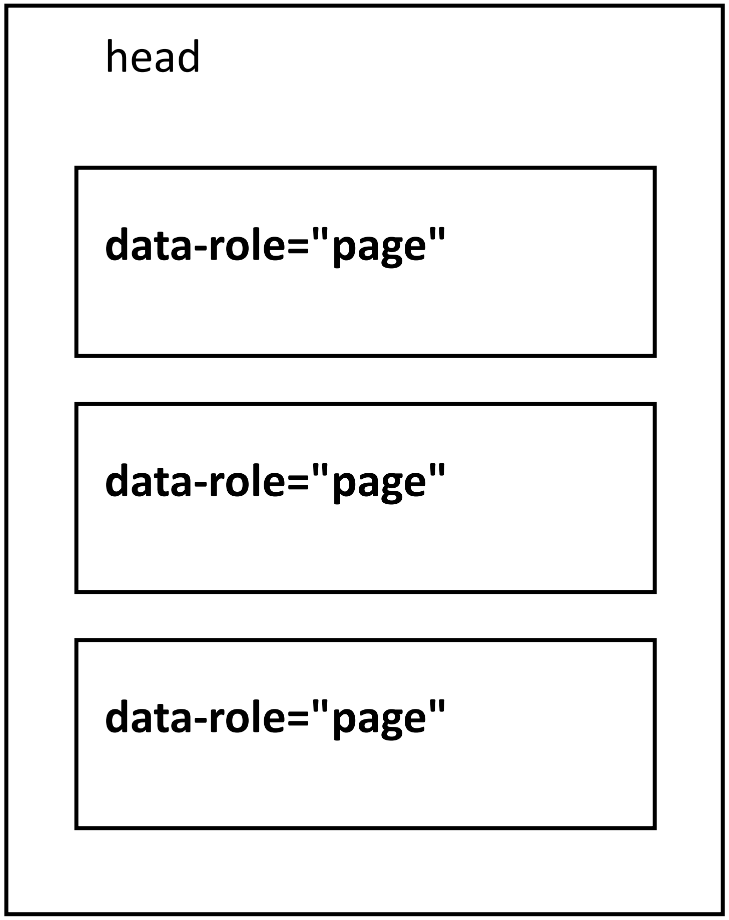
API
- Configuring defaults
- Events
- Methods & Utilities
- Data attribute reference
- Theme framework

© 2011 The jQuery Project

- Our free webinar on jQuery Mobile
  - www.webvanta.com/jqm-webinar
- Our list of references
  - www.webvanta.com/jquery-mobile

A Page is Not Always a Page

HTML Page



head

data-role="page"

data-role="page"

data-role="page"

- One HTML document can include multiple conceptual pages
- A "page" is just an HTML element with data-role="page"
- When a page link is followed, new page elements added to DOM

The Simplest Page

```
<div data-role="page">

    <div data-role="header">
        <h1>My Title</h1>
    </div>

    <section data-role="content">
        <p>Hello world</p>
    </section>

 </div>
```

jQuery Mobile Code Example

- Download the code files for this lesson
- Folder "jquery-mobile"
  - Home page is jqm-demo.html
  - Libraries are loaded from CDNs
  - Uses default theme
    - A few CSS overrides in jqm-overrides.css
- Note: main page runs fine locally, but loads of map and contact page will fail because Ajax requests aren't allowed on local files

Other Mobile Frameworks

- Sencha Touch
  - www.sencha.com/products/touch
- jQTouch
  - www.jqtouch.com
- jqMobi
  - http://jqmobi.com
- iUI
  - http://code.google.com/p/iui
- Zepto.js (smaller replacement for jQuery)
  - www.zeptojs.com

HTML5 Features for Mobile Web Apps

- Local Storage
  - Store string data easily
- Web SQL Database
  - Local database in the browser
- Application Cache
  - Allows semi-permanent app storage in browser
- Geo-location API
  - Get the visitor's geographic location

HTML5 Local Storage

- Provides local storage as key/value pairs
- Allows up to 5 Mbytes of storage, vs. 4K for cookies
- Simple JavaScript API
  - `localStorage.setItem("thisClass", "Design for Mobile");`
  - `localStorage.getItem("thisClass");`
- sessionStorage is similar but data goes away when browser is closed
- Stores strings only, but you can convert binary data to strings easily

HTML5 Web SQL Database

- SQLite implemented in the browser
- Great for being able to search data locally
- Works even if offline
- May be dropped from HTML5 spec, and not widely supported on desktop
  - Available on all recent iOS devices
  - Available in default Android browser, even old ones
  - May not be available in other Android browsers

HTML5 Offline Mode

- Application Cache stores files in browser cache, more permanently than normal cache
- Manifest file, provided by server along with normal site, provides list of all assets to be stored offline
- Cache is updated whenever the manifest changes (assuming connectivity, of course)
- Limited to 5 Mbytes normally
  - For Google App Store (Play), size can be overridden

Geolocation API

- Simple JavaScript API available in all recent SmartPhones
  - `getCurrentPosition()`
  - `watchPosition()`
- User is asked for permission before position is provided
- Underlying mechanism is up to the device
  - GPS, cell tower triangulation, WiFi

Accessing other Devices

- Device Motion API provided by Mobile Safari
  - iOS 4.2 and later
- Vibration API spec in development but not yet implemented
- Camera access not yet possible from browser

Triggering Other Apps

- Use URL scheme for handing off to other apps
  - A link to maps.google.com will open the Maps app
  - A link to youtube.com will open the YouTube app
  - mailto: opens the email client, as you'd expect
  - tel: asks user if they want to dial the phone number
  - sms: launches the messaging app
  - other apps can define their own URL scheme
    - e.g., href="fb://feed" opens the Facebook app to the news feed

Distributing Web Apps

- They are just web sites!
- Use "save to home screen" prompt as described in previous lesson
- Google provides HTML5 Web App store for Android
  - http://openappmkt.com/
  - Lacks Apple's App Store restrictions
    - Also lacks quality control

Mobile Emulators

- Testing mobile apps on a range of devices a real challenge
- Emulators are helpful but not the full solution
  - iOS simulator (iPhone and iPad)
    - Included in XCode, free download for Mac users (not available for Windows)
    - Available in App Store; requires recent Mac OS
  - Android emulator
    - http://developer.android.com/sdk/installing.html
    - Slow, relatively technical to install, but available on all platforms
  - Windows Phone emulator
    - http://msdn.microsoft.com/en-us/library/ff402530(v=vs.92).aspx

Homework

- Create a simple website with jQuery Mobile
  - Explore the listview, buttons, collapsibles, header, and footer options
- Build a simple web page using responsive design
  - Create separate phone, tablet, and desktop optimized views