**Overarching goals of the guidance:**“Helping people understand how to think”.

**Guidance topics**

* Project introduction
  + Goals, technologies etc
  + Exploring the reference application
* Why mobile web?
  + Choosing between native, web and PhoneGap
    - Flow chart
    - Links to other articles
    - 1-2 page narrative
* Mobile web development approaches
  + Using a server-side proxy
  + Using responsive design
  + Building a standalone site
* Architectural choices
  + Single page applications
  + Applications with page refresh
  + Combined approach
    - Progressive enhancement
    - Maintaining flow
    - Thematic consistency
* Deciding which devices to support
  + Determining level of support
    - Diagram
  + Choosing test devices
    - Links to suitable references
* Detecting features and capabilities
  + Features and capabilities vs devices
  + Detection methods (Pros/cons)
    - UA string
    - Device databases
    - Client-side detection
    - Third party tools (Modernizr)
* Spectrums of reuse when converting a desktop app to mobile
  + Server-side business logic
  + HTML
  + CSS
  + JavaScript (client-side functionality)
    - Desktop toolkits and micro-frameworks
  + Content/data
    - XML, JSON
    - Graphics
    - Media (video, audio)
* Managing media assets
  + Basic problem set
  + Overview of approaches
  + Lots of links to community discussions
* Taking advantage of new technologies
  + Manifest, DBs, app cache
  + Location
* Testing guidance

**Patterns**

* Location detection
* Collection of patterns TBD
* Login

**Reuse**

Portions of the HTML can be reused but, where possible the following should be adjusted:

* Remove redundant div elements. Where possible, select the actual element that will be styled. This will also often enable you to reduce the number of classes being used.
* Convert div elements to new HTML5 document level elements.
* Ensure all content is enclosed in a semantic element such as a paragraph tag. Content should not float freely within div elements.
* Check HTML syntax. Ensure markup is properly nested. This is good practice but will also ensure older browsers can parse the markup as expected.

Changes made to desktop app markup

* Redundant divs removed around form labels.
* Will replace certain structural divs with HTML5 elements.
* Changed h1s to h2s in certain contexts to improve semantics and reusability.
* Removed image within button element

\* Project introduction

\* Goals, technologies etc

\* Exploring the reference application

\* Why mobile web?

\* Choosing between native, web and PhoneGap

\* Flow chart

\* Links to other articles

\* 1-2 page narrative

\* Mobile web development approaches

\* Using a server-side proxy

\* Using responsive design

\* Building a standalone site

\* Architectural choices

\* Single page applications

\* Applications with page refresh

\* Combined approach

\* Progressive enhancement

\* Maintaining flow

\* Thematic consistency

\* Deciding which devices to support

\* Determining level of support

\* Diagram

\* Choosing test devices

\* Links to suitable references

\* Detecting features and capabilities

\* Features and capabilities vs devices

\* Detection methods (Pros/cons)

\* UA string

\* Device databases

\* Client-side detection

\* Third party tools (Modernizr)

\* Spectrums of reuse when converting a desktop app to mobile

\* Server-side business logic

\* HTML

\* CSS

\* JavaScript (client-side functionality)

\* Desktop toolkits and micro-frameworks

\* Content/data

\* XML, JSON

\* Graphics

\* Media (video, audio)

\* Managing media assets

\* Basic problem space

\* Overview of approaches

\* Lots of links to community discussions

\* Taking advantage of new technologies

\* Manifest, DBs, app cache

\* Location

\* Testing guidance