



# Designing for the Mobile Web

## Lesson 2: Device Constraints and Opportunities

Michael Slater, CEO  
Andrew DesChenes, Dir. Services  
[course-support@webvanta.com](mailto:course-support@webvanta.com)  
888.670.6793

**[www.webvanta.com](http://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](http://www.online-web-courses.com)



Michael Slater  
Cofounder & CEO



Andrew DesChenes  
Director of Services

# Topics for This Lesson

- Phones vs. tablets: what do you need to do differently?
- Design for touchscreens
- What screen sizes do you target?
- Gestures and how to use them
- Being location-aware

# Homework Review

- Define an app and/or site strategy for the following situations:
  - Retail store wanting to attract new customers
  - Restaurant wanting to provide easy to-go ordering
  - Manufacturer wanting to provide product information to prospective purchasers
  - Publisher wanting to distribute content and monetize with advertising
  - Publisher wanting to distribute paid content

# Strategies for Designing for Tablets and Phones

- Tablets typically need a visual design much closer to desktop than to phone
- Several options
  - Use responsive design to make a single HTML page adapt to desktop, tablet and phone
  - Use one set of pages for desktop and tablet, and another set for phone
  - Create three sets of pages, for desktop, tablet, and phone

# Our Preferred Strategy

- Use one design for tablet and desktop
  - Accept that hover can't be used (no tooltips)
  - Keep click/touch targets large enough
  - If design dictates, use responsive techniques to provide modest modifications for tablet vs. desktop
- Use a different design for phones
- Potential challenge: big phones and small tablets
  - Breakpoints are becoming fuzzy



"Feature Phones"  
Ignore

**Webkit Browser**  
HTML5 + CSS3

**IE9 browser in  
recent phones**

Ignore older models  
with IE7 browser



# Amazing Browsers!

- When developing for SmartPhones, you can assume very capable HTML5/CSS3 browsers
  - CSS3 auto-hyphenation (iOS)
  - HTML5 form input types
  - HTML5 database and offline mode
- New JavaScript APIs
  - geocoding
  - device orientation
  - fullscreen



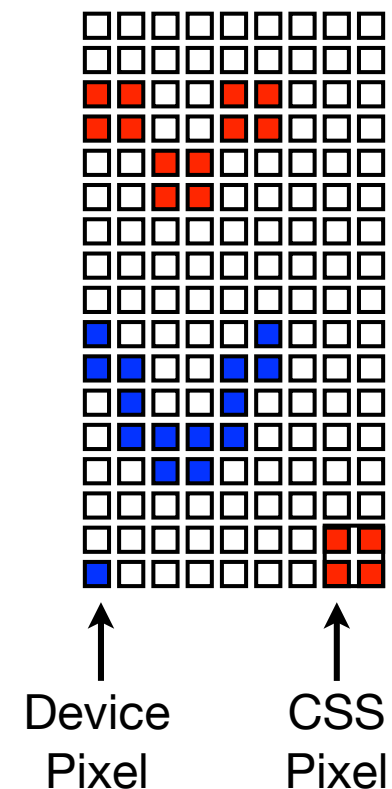
# Flash, Video & Animation

- No Flash on iPhone and iPad
  - No future Flash development for Android
- Use HTML5 audio and video players
- Create animation using JavaScript & CSS3

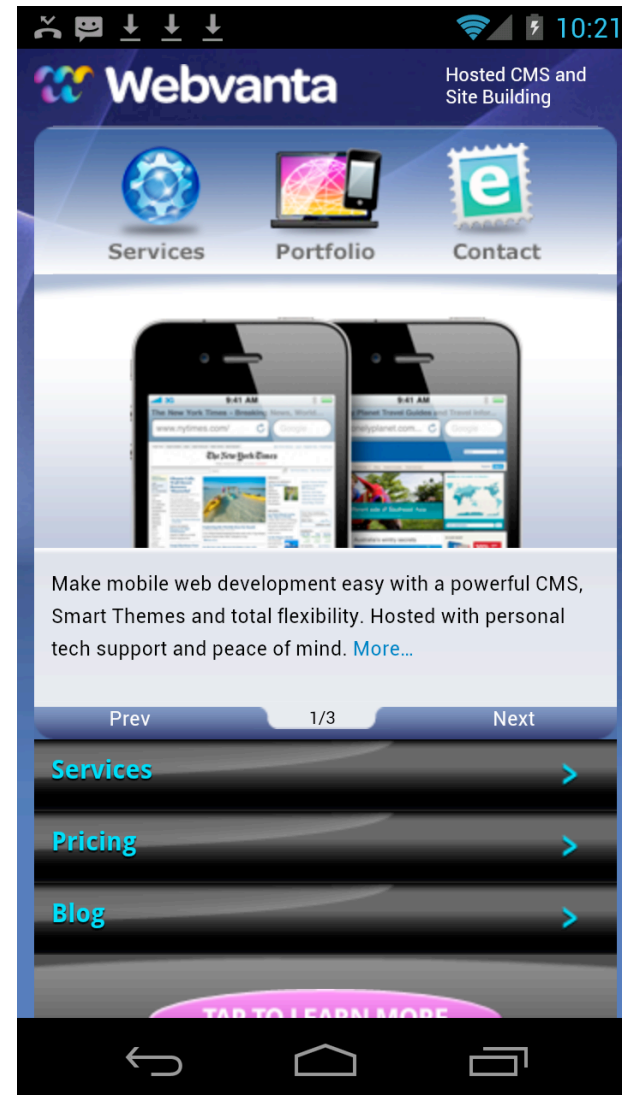
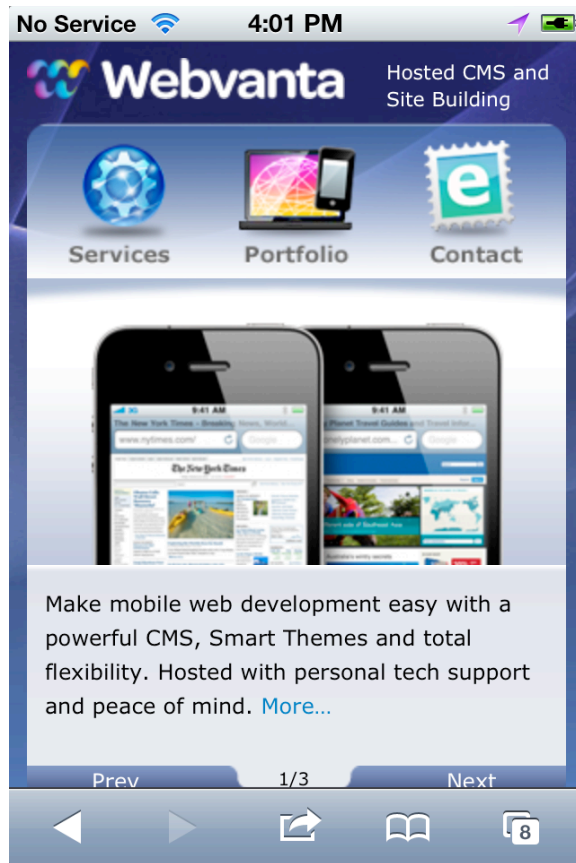


# Screen Sizes

- Physical size: 3.5 to 4.3 inch diagonal
- Density: number of pixels per inch
- Effective resolution for HTML content
  - "CSS resolution", or  
"density-independent resolution"
  - 320 x 480 to 360 x 640
- Device pixel resolution
  - 320 x 480 to 720 x 1280 (720p)



# Android Screen Can Radically Change the View



# Usable Screen Space

- iPhone (portrait)
  - 320 px wide
  - 480 px - 44 px (toolbar at bottom) - 20 (status bar at top)  
= 416 px high if URL bar is hidden (268 px in landscape)
- Android
  - Varies widely depending on the phone
  - Many different browsers
  - Up to 360 px wide (no doubt wider before long)
  - Up to 640 px tall
    - No fixed toolbar at the bottom

# The Browser's Viewport

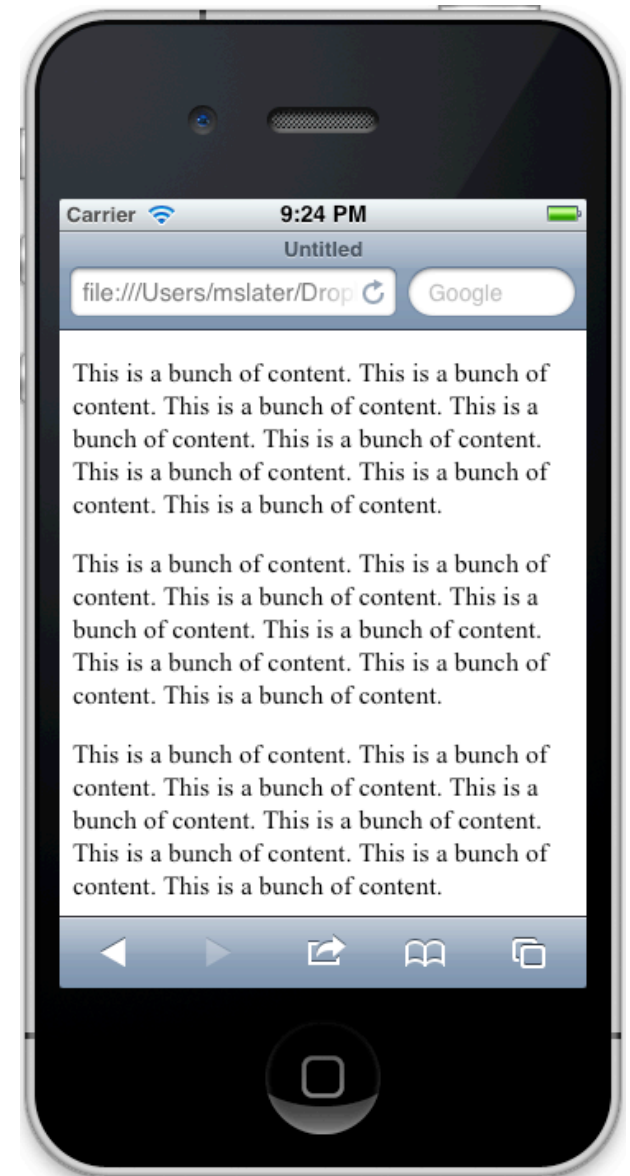
- The viewport is the virtual screen on which your content is displayed
- Default viewport is desktop-sized, even on mobile devices
  - iOS: 980px, Android: 800px, IE: 974px



# Setting the Mobile Viewport

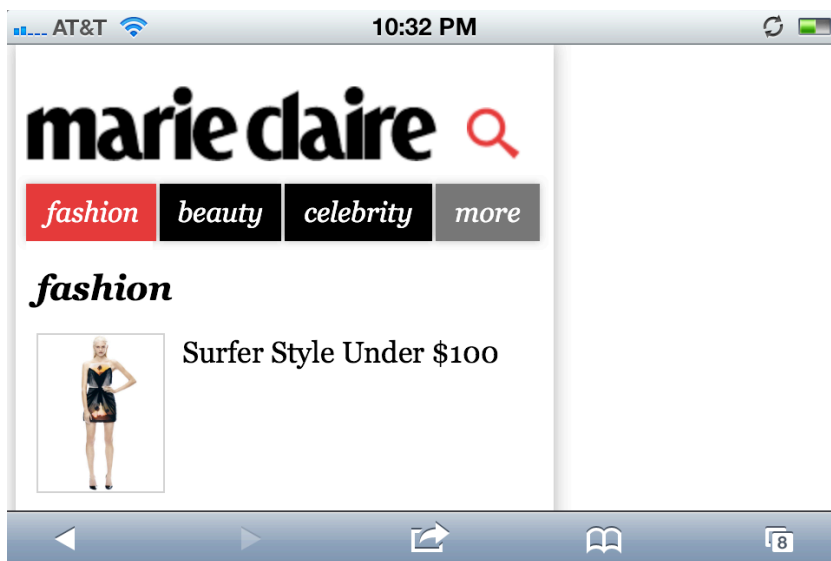
- Set the viewport size to the device's effective screen size

```
<meta name="viewport"  
content="width=device-  
width,  
initial-scale=1.0, user-  
scalable=no">
```



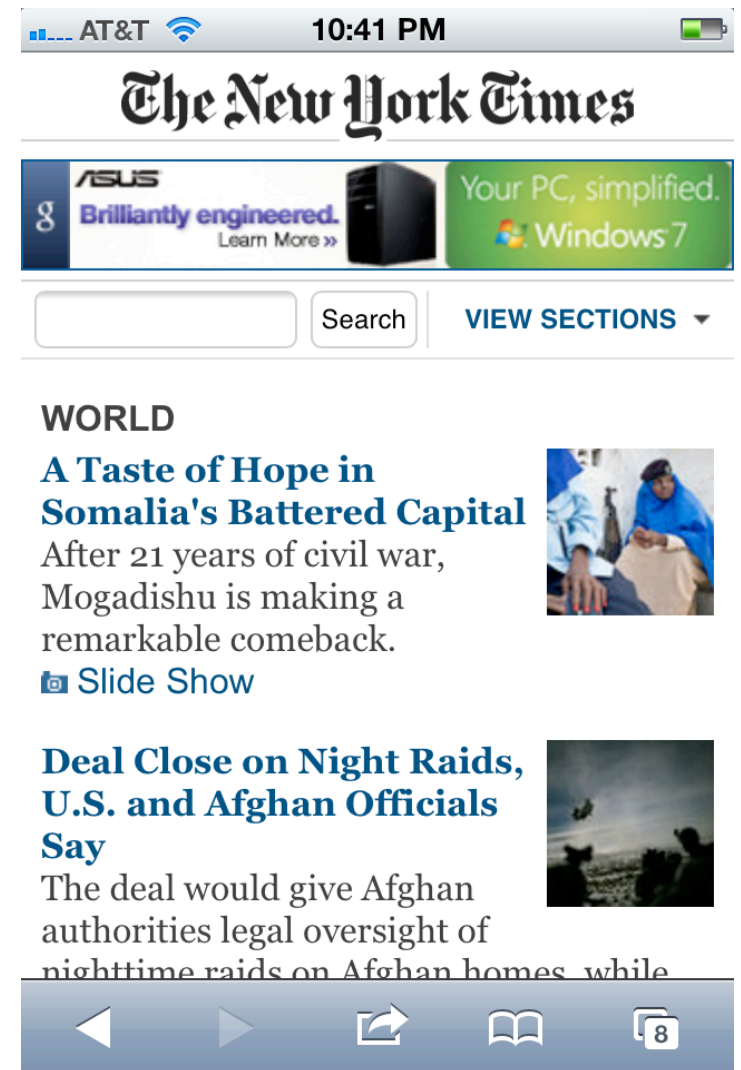
# Don't Forget Landscape

- Mobile sites and apps must adapt to either portrait or landscape orientation
- Portrait orientation is most widely used because it is the natural way to hold phone and height is small in landscape
- Landscape provides more room for text
- Mobile web pages must scale to any reasonable width



Don't do this!  
(fixed width designed for portrait)





# Adapting to Screen Density

- Phones and tablets have a range of pixel densities
  - Apple calls high-density "retina display"
  - Android defines a range of display resolutions
- HTML content uses *effective pixels*, so it appears at the same size regardless of density
- By default, images are scaled up on high-density screens to maintain size
  - If you provide a higher-resolution alternate image and display it without scaling, it will look sharper

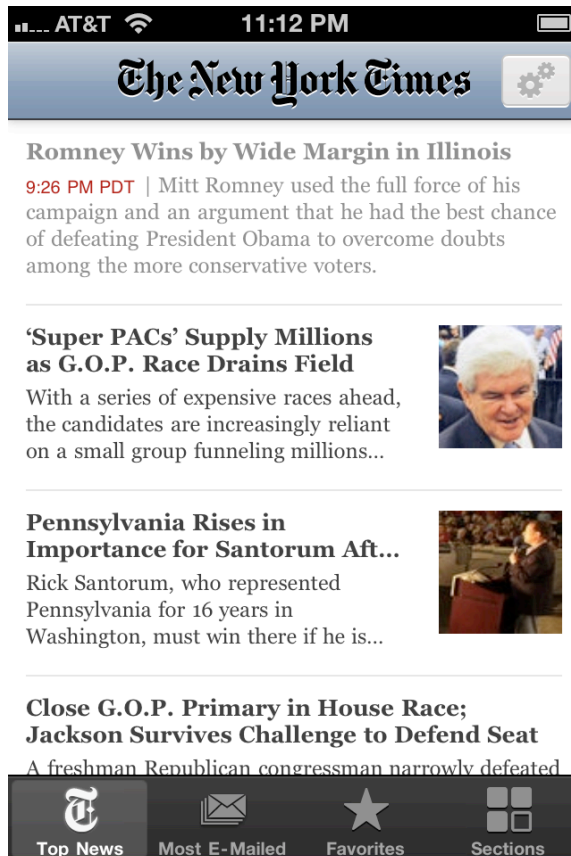
# Choosing Images Based on Screen Density

- Create images of twice the desired size for use with Retina display
- Use CSS media queries for background images
- Use JavaScript or server-side solutions for inline images
- For apps, you can directly select different resolution images for different resolution devices
- Like everything, this is more complicated for Android devices
- See the resources listed at [www.webvanta.com/mobile-design](http://www.webvanta.com/mobile-design)

# App vs. Website Design

## App

Status bar at top



Apps typically have toolbar at bottom

## Website

Status bar at top

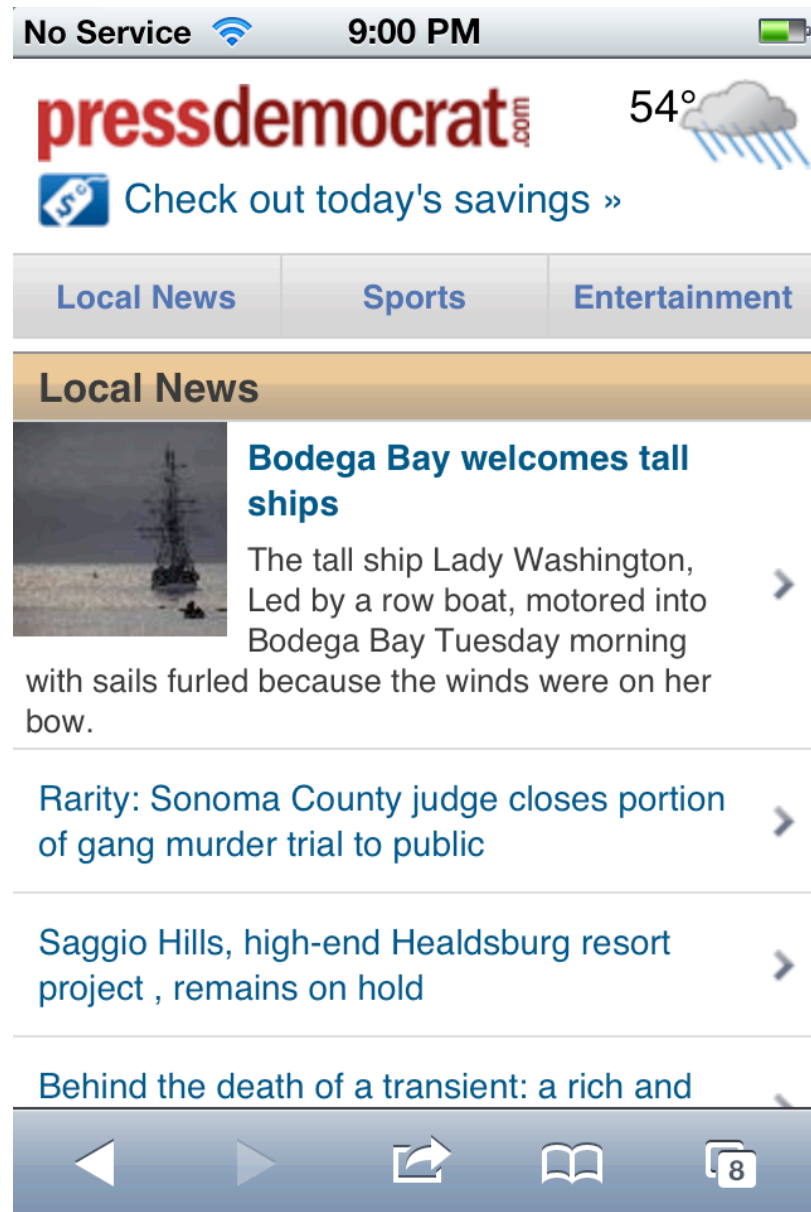


Browser has toolbar at bottom

# Hiding the URL Bar

- With a little JS, you can hide the space-consuming URL bar at the top of the browser
- See <http://mobile.tutsplus.com/tutorials/mobile-web-apps/remove-address-bar/>

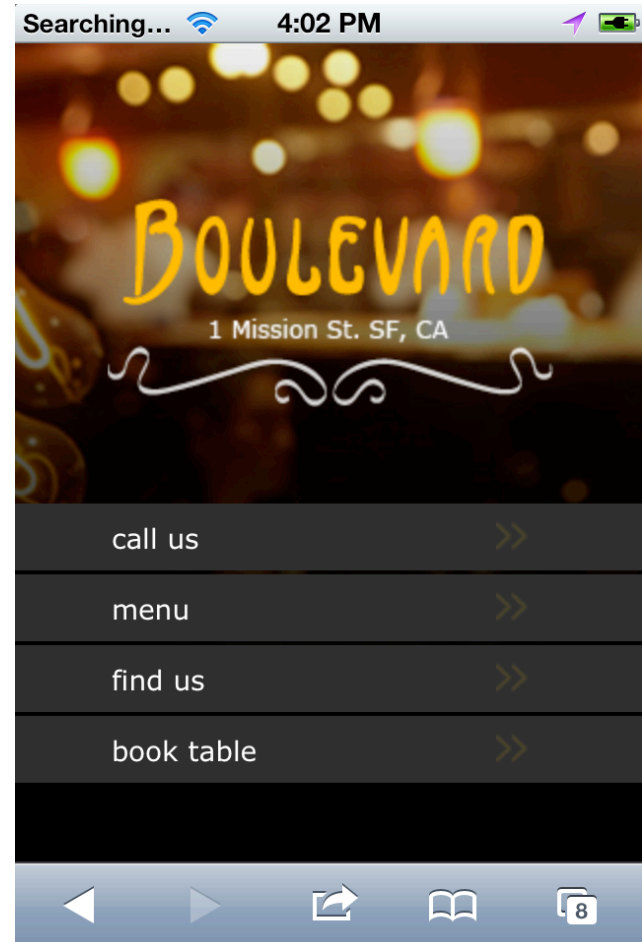
```
function hideAddressBar() {  
  if(!window.location.hash)  
  {  
    if(document.height < window.outerHeight)  
    {  
      document.body.style.height = (window.outerHeight + 50) + 'px';  
    }  
    setTimeout( function(){ window.scrollTo(0, 1); }, 50 );  
  }  
}  
window.addEventListener("load", function(){ if(!window.pageYOffset)  
{ hideAddressBar(); } } );  
window.addEventListener("orientationchange", hideAddressBar );
```



# Hiding the Button Bar in iOS

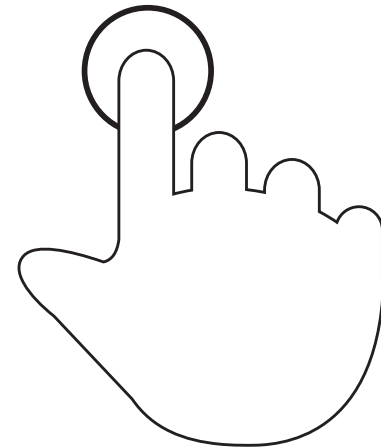
- iOS always displays a button bar at the bottom of the browser
- "Full-screen web apps" can hide the normal browser controls
  - Only works when launched from a shortcut

```
<meta name="apple-mobile-web-app-capable" content="yes" />
```



# Design for Touchscreens

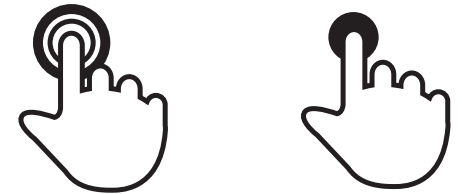
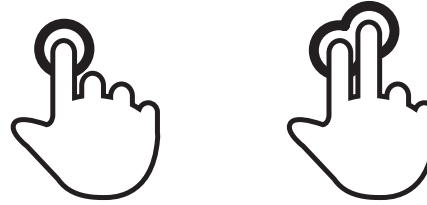
- No hover (tooltips, dropdowns)
- Use large touch targets
  - Min. 29 px wide x 44 px tall
- Position touch targets with hands and fingers in mind
- Support flexible layout
  - Varying screen sizes
  - Portrait and landscape orientations



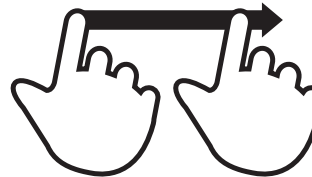


# Common Gestures

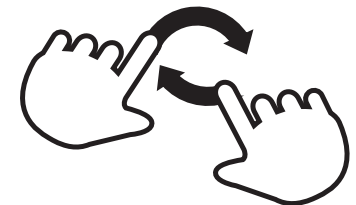
- Tap, two-finger tap, double tap, press



- Drag, flick



- Pinch, spread, rotate



# Choose the Right Keyboard with HTML5 Input Types

- Text: display a standard keyboard

```
<input type="text" />
```

- Telephone: display a telephone keypad

```
<input type="tel" />
```

- URL: display a URL keyboard

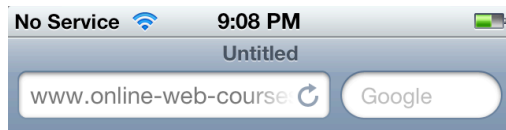
```
<input type="url" />
```

- Email: display an email keyboard

```
<input type="email" />
```

- Zip Code: display a numeric keyboard

```
<input type="text" pattern="[0-9]*" />
```



Plain Text Field

Email Field

Telephone Field



Plain Text Field

Email Field

Telephone Field

Number Field



Plain Text Field

Email Field

Telephone Field

Number Field

Date Field



Telephone Field

Number Field

Date Field



Telephone Field

Number Field

Date Field

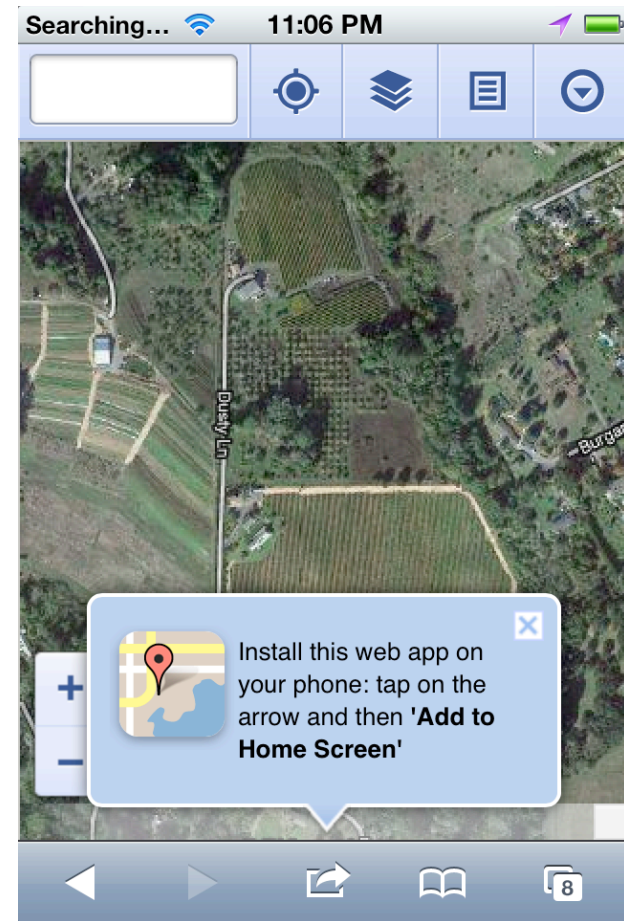


# What Makes an App Feel Like an App?

- Freedom from conventions of web layout
- Designed for a particular purpose
- Optimized for a phone-sized screen
- Quick screen changes, animation
- Toolbar at bottom of screen
- Works offline (sometimes)
- *None of this requires a native app!*

# Installing Apps

- Native app
  - Find in the app store
  - App is automatically installed with icon
- Web app
  - Find from web search
  - "Save bookmark to home screen" is equivalent of "installing"



# Phone "Desktop" Icons

- "Save to home screen" puts app icon on the phone, just like a native app

```
<link rel="apple-touch-icon" href="touch-icon-iphone.png" />
```

```
<link rel="apple-touch-icon" sizes="72x72" href="touch-icon-ipad.png" />
```

```
<link rel="apple-touch-icon" sizes="114x114" href="touch-icon-iphone4.png" />
```

- Android also supports the Apple icon tags

# Homework

- Explore a range of mobile websites and apps
  - What do you find that you like?
  - What do you find that you don't like?
  - What patterns do you see?
  - How do sites and apps differ?