

Mobile Web & Native Accessibility in iOS & Android Workshop

@pauljadam

Accessibility Evangelist, Deque Systems

#CSUNmobile Workshop Twitter Hashtag

<http://pauljadam.com/bootcamp/>

Introductions

- **via #CSUNmobile Twitter hashtag**
- **Your Name?**
- **Job Title or Accessibility Work?**
- **Something Specific You Want to Learn Today?**



Legal Requirements

Mobile Accessibility is Law

- **Section 508 (Amendment to the Rehabilitation Act of 1973)**
- **CVAA (21st Century Communications and Video Accessibility Act)**
- **ACAA (Air Carrier Access Act)**
- **ADA (Americans with Disabilities Act of 1990)**
- **AODA (Accessibility for Ontarians with Disabilities Act)**



HTML Accessibility Overview

- **Web Accessibility APIs**
- **HTML4 Accessibility Techniques**

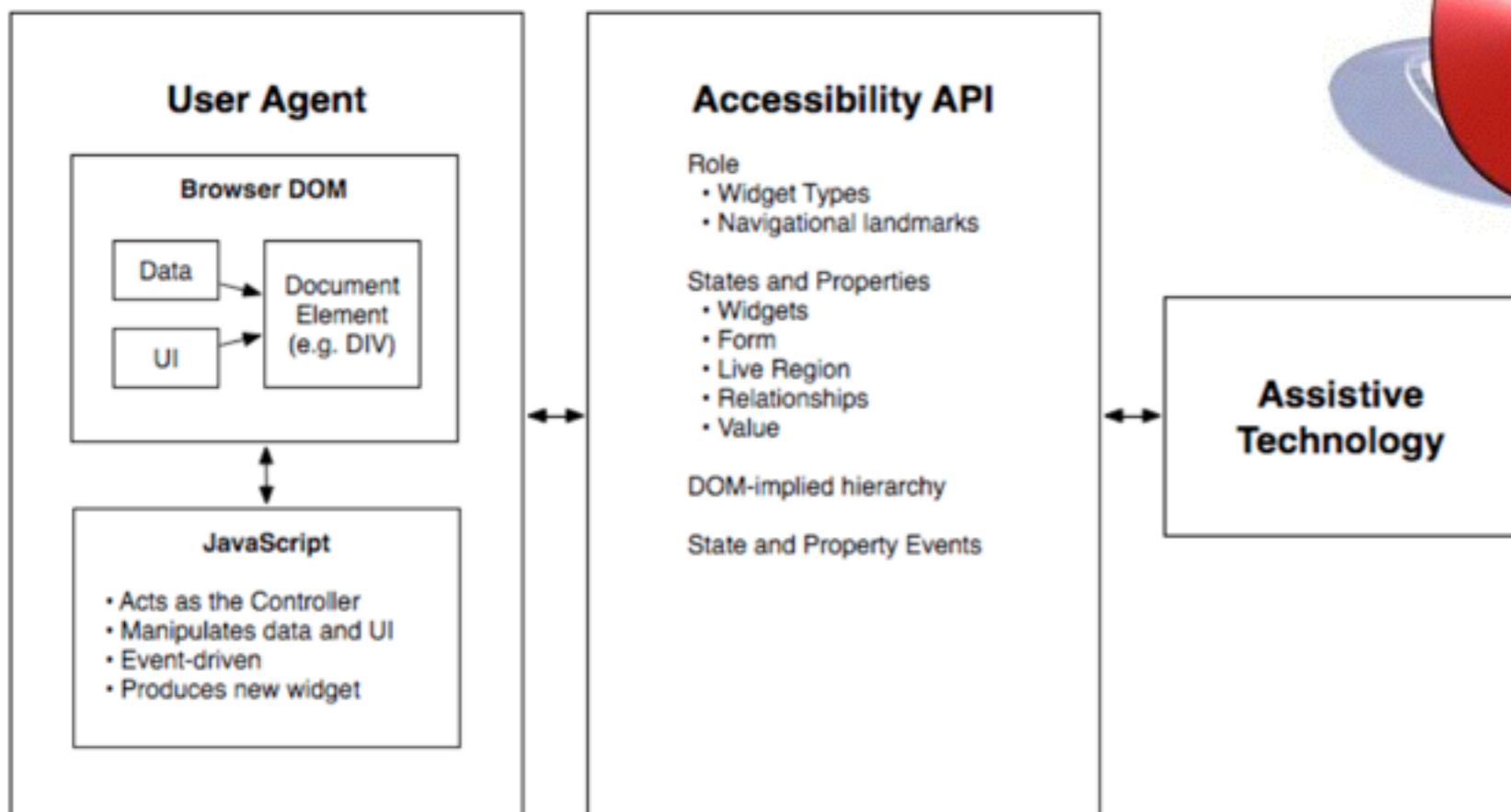
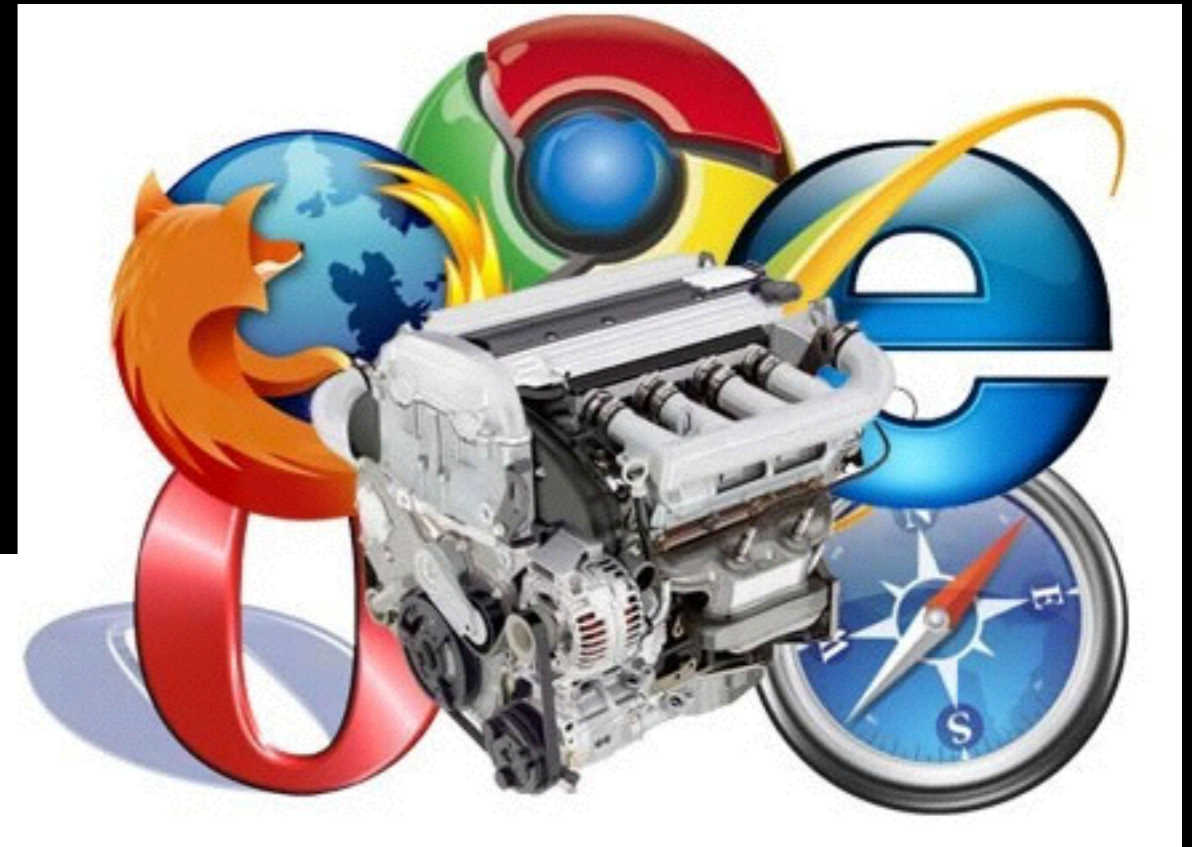
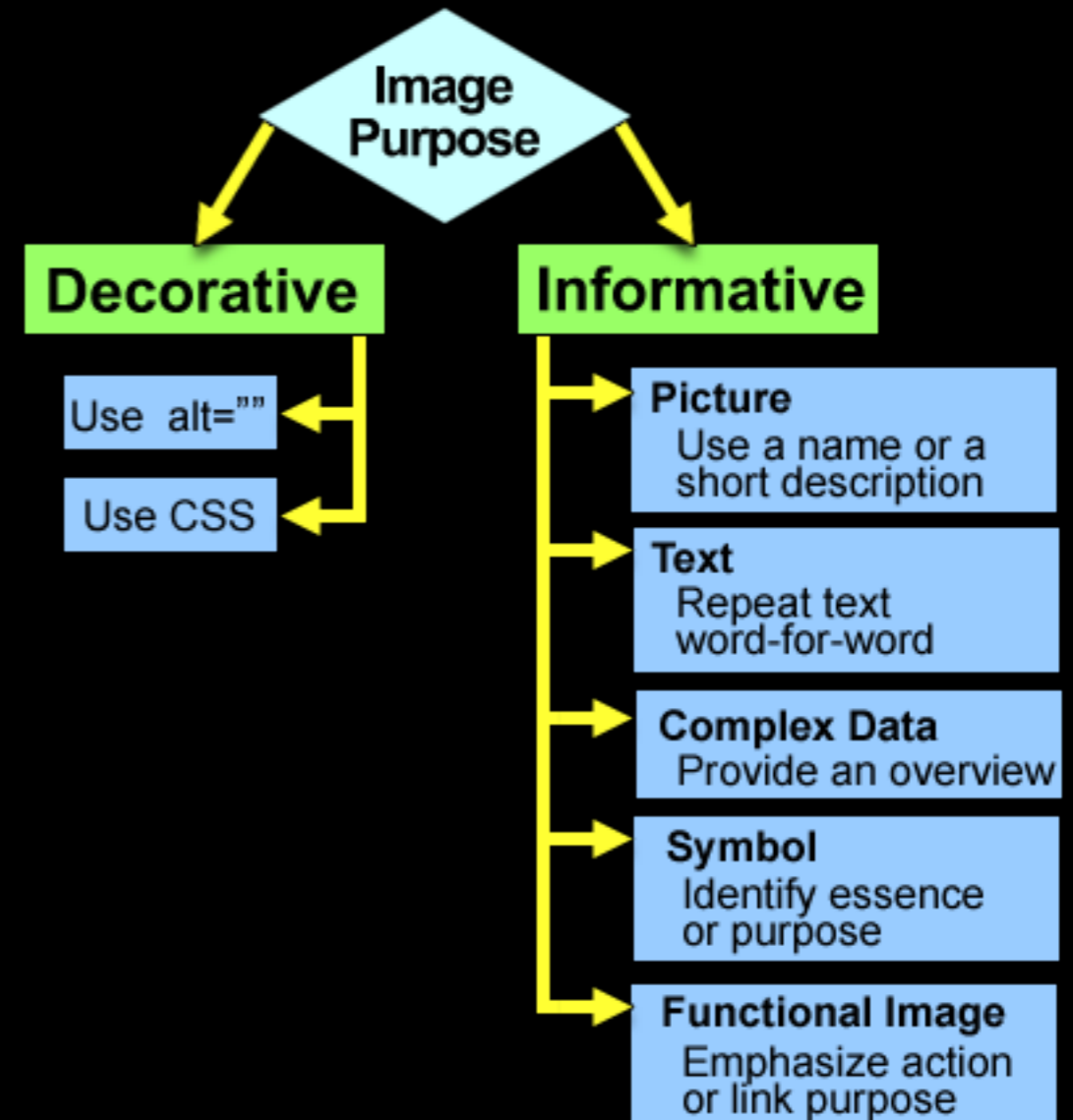


Figure 1: The contract model with accessibility APIs


Accessible Name & ALT Text

- Active (links/buttons), Decorative, Informational, CSS Icons, Background Images, etc
- <http://pauljadam.com/demos/accessiblenameimg.html>
 - aria-labelledby
 - aria-label
 - alt
 - title
 - no accessible name
- <http://www.w3.org/TR/html51/semantics.html#alt>
- http://www.d.umn.edu/~lcarlson/eclasses/104_images/1.3_lecture_short_how.html



Headings


Deque Blog



What iOS Traits Actually Do

This post was co-authored by Chris McMeeking and Alistair Barrell. A useful and important feature in the realm of iOS Accessibility. An Accessibility trait is used to describe for what an element in your application does. It is important to set it correctly so that users are not confused when clicking on [...]

EST PRACTICE Read more about [What iOS Traits Actually Do](#)



Accessible Text Input in Android

This post was co-authored by Chris McMeeking and Melinda Kothbauer. EditText objects are difficult because the common practices (such as using accessibility descriptions), do not work. Android provides the framework to make it easier to use what, when, and how to use the built-in tools. In [...]

EST PRACTICE Read more about [Accessible Text Input in Android](#)

Deque Announces Support for FIPS 201-2 PIV in its WorldSpace

WorldSpace accessibility testing software now complies with the new mandate for Secure Per

Headings

- 1: Deque Systems | Software, Training and Section 508 Compliance
- 2: "It's Easy to be Smart After the Fact." Deque University
- 3: That defense doesn't work here....knowledge is power. Get educated.
- 2: Common Website Accessibility Myths
- 3: Myth #1
- 3: Myth #2
- 3: Myth #3
- 2: Deque Blog
- 3: What iOS Traits Actually Do
- 3: Accessible Text Input in Android
- 3: Deque Announces Support for FIPS 201-2 PIV in its WorldSpace
- 2: Accessibility Stories
- 2: Deque Highlighted Partnership
- 3: We are the official jQuery accessibility partners and jquery partners in accessibility
- 2: Essential Guide to Digital Accessibility Infographic

Data Tables

2015 Sales Report

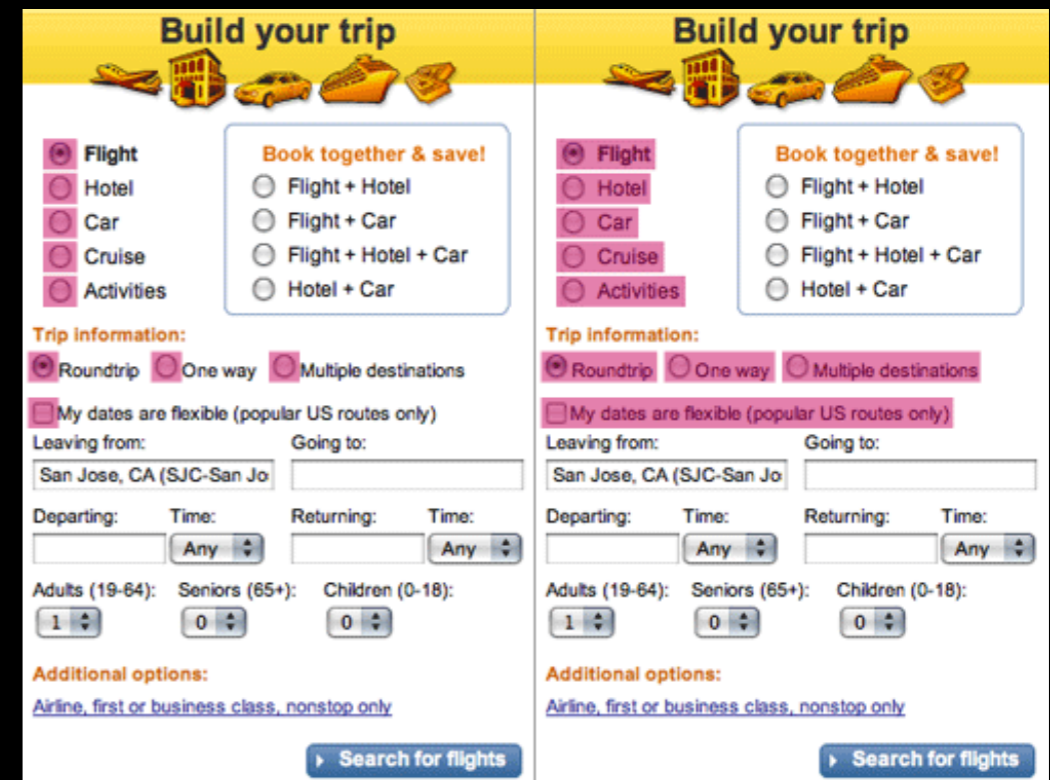
- <http://pauljadam.com/demos/tableA11yTest.html>
- `<th scope=row/col>`
- `<caption>`
- `<td headers="id1 id2 id3">`

Report	Quarter	Apples	Oranges	Bananas
print	Q1	100	200	250
print	Q2	120	180	230
print	Q3	110	120	210
print	Q4	140	150	300

Forms



- Labels, Fieldset/Legend, Error Messages briefly



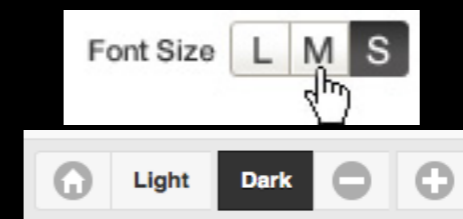
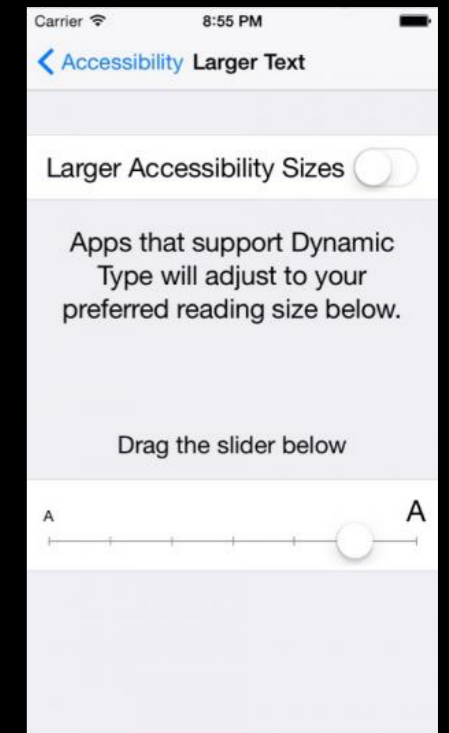
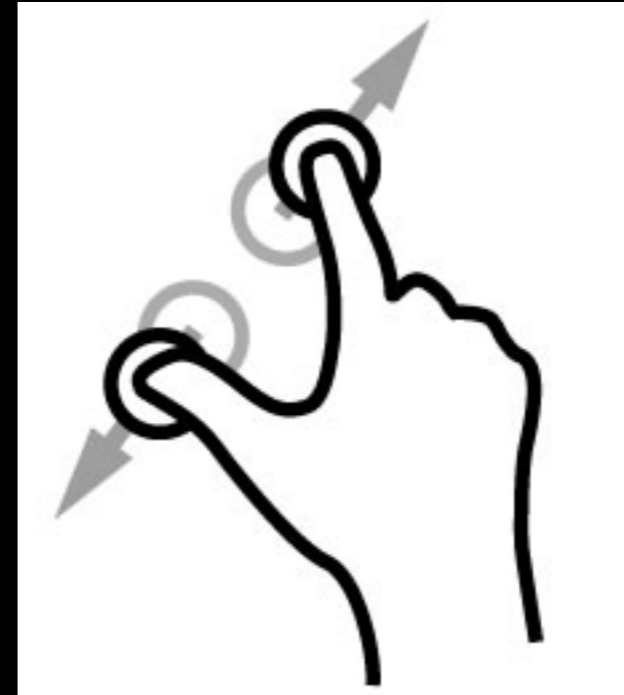
Keyboard Accessibility

- **Keyboard & Screen Reader Operable HTML Controls (`<input>` `<button>` `<a href>` `tabindex=0/-1`)**
- **Focusable UI Controls**
- **Focus Outline Visibility**
- **Match Focus with Hover**
- **Never use positive `tabindex` values!**
- **`<button>` vs. `<div role=button tabindex=0>`**
- **`<a href>` vs `<a>`**

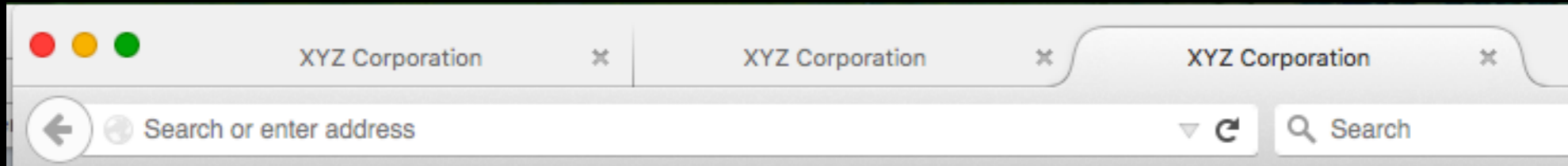


Text Size & Readability

- **Resize 200%+**
- **Don't disable pinch-to-zoom!**
- **Contrast differences for Large vs Small Text**
- **Responsive Design's Benefits for Text Resizing**
- **<http://pauljadam.com/demos/apple-system-css-font.html>**



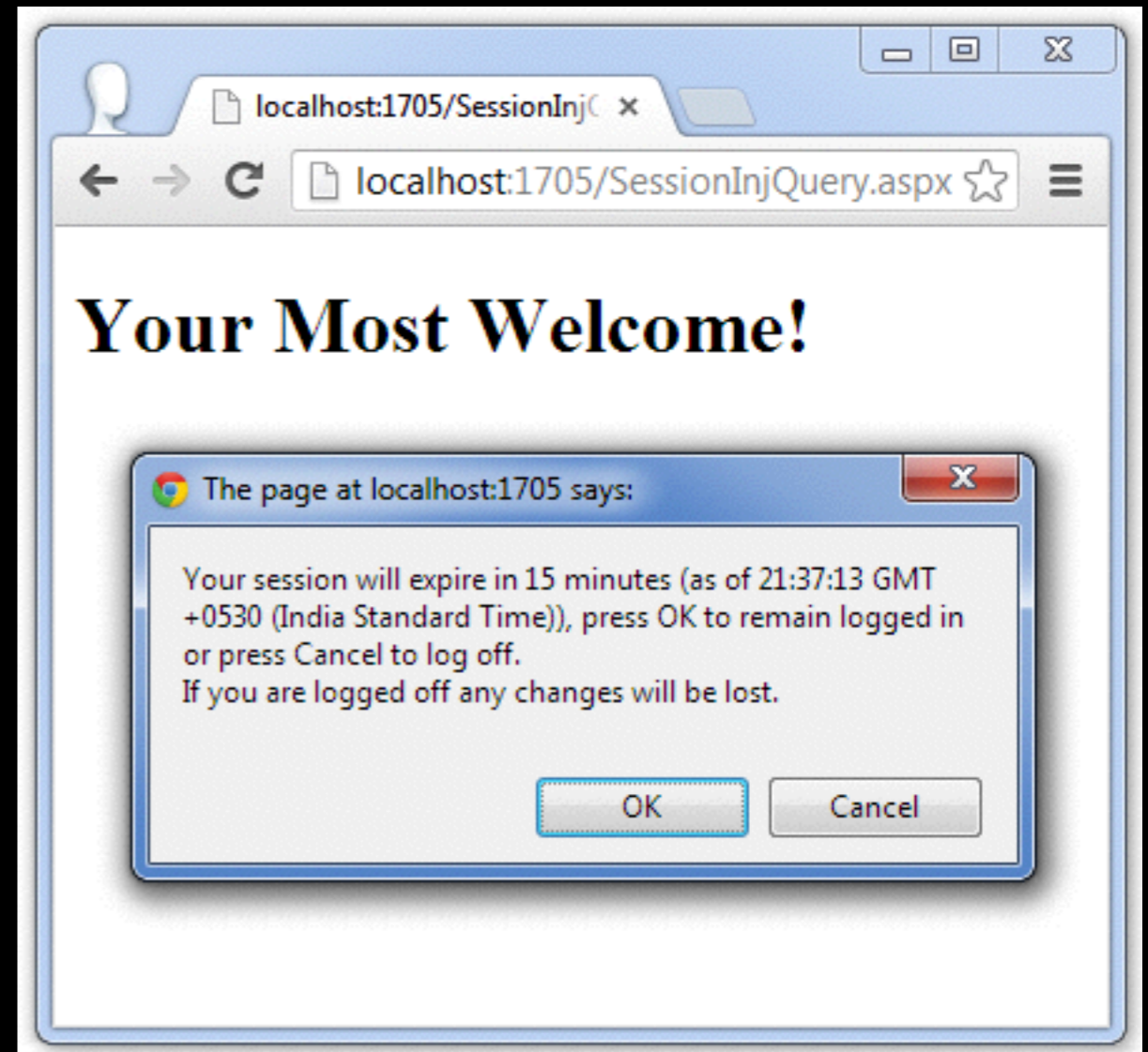
Page Titles



- **Common problem in single page apps**
- **Can modify page title dynamically using JavaScript.**
 - **`document.title = "New Page Title";`**

Session Timeouts

- Old fashioned JavaScript `alert()`'s work well to get the user's attention even if they've left the browser window.
- Modal dialogs may not get user's attention if they're working in another app.
- <http://pauljadam.com/demos/timeout.html>



Pause, Stop, Hide

- <http://wet-boew.github.io/wet-boew/demos/tabs/tabs-carousel-en.html>
- <http://www.w3.org/WAI/tutorials/carousels/animations/#finalized-carousel>
- <http://shouldiuseacarousel.com>



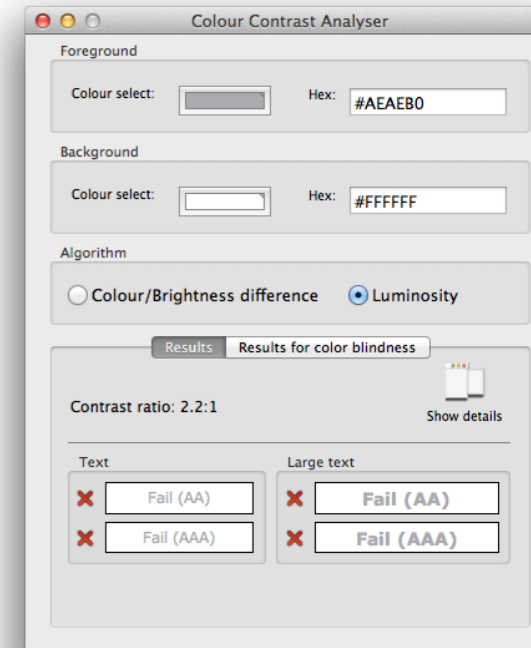
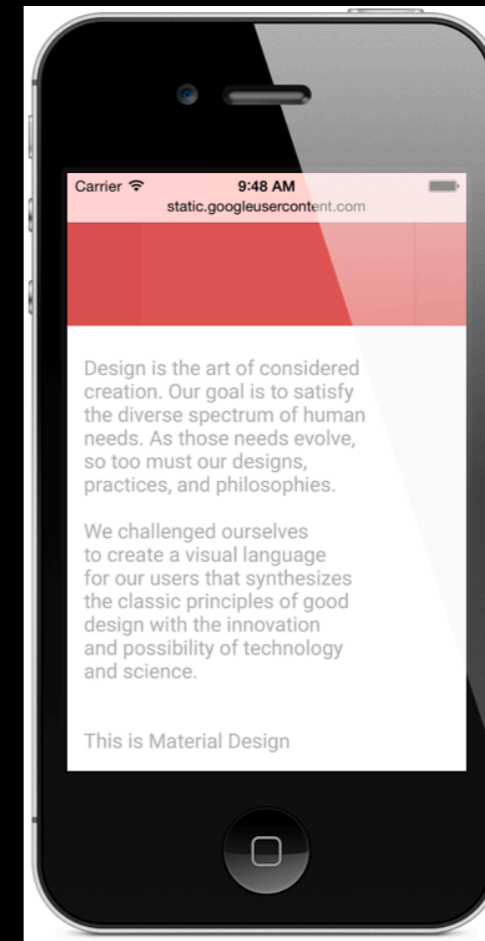
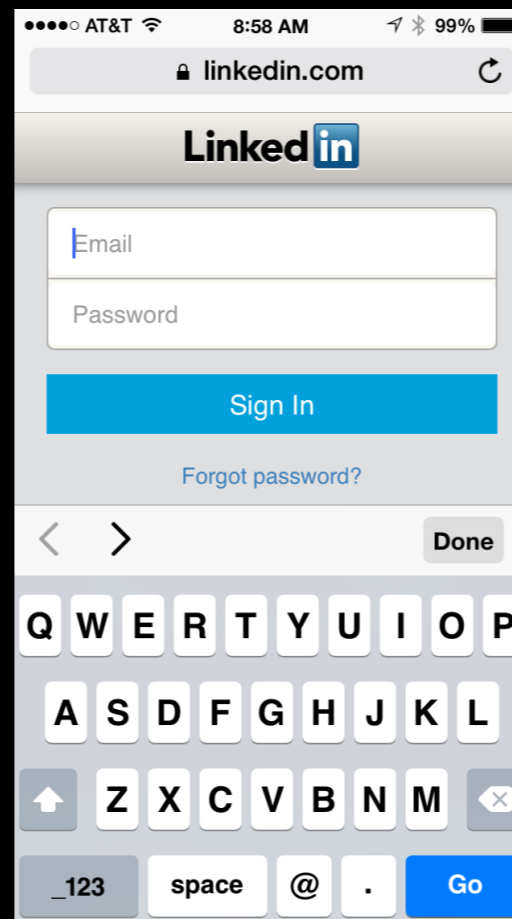
Focus Outline & Visibility

- WCAG requires all elements to be visible on focus
- Skip links must be visible when you TAB to them
- NEVER USE OUTLINE:NONE!
- <http://pauljadam.com/demos/focusvisible.html>



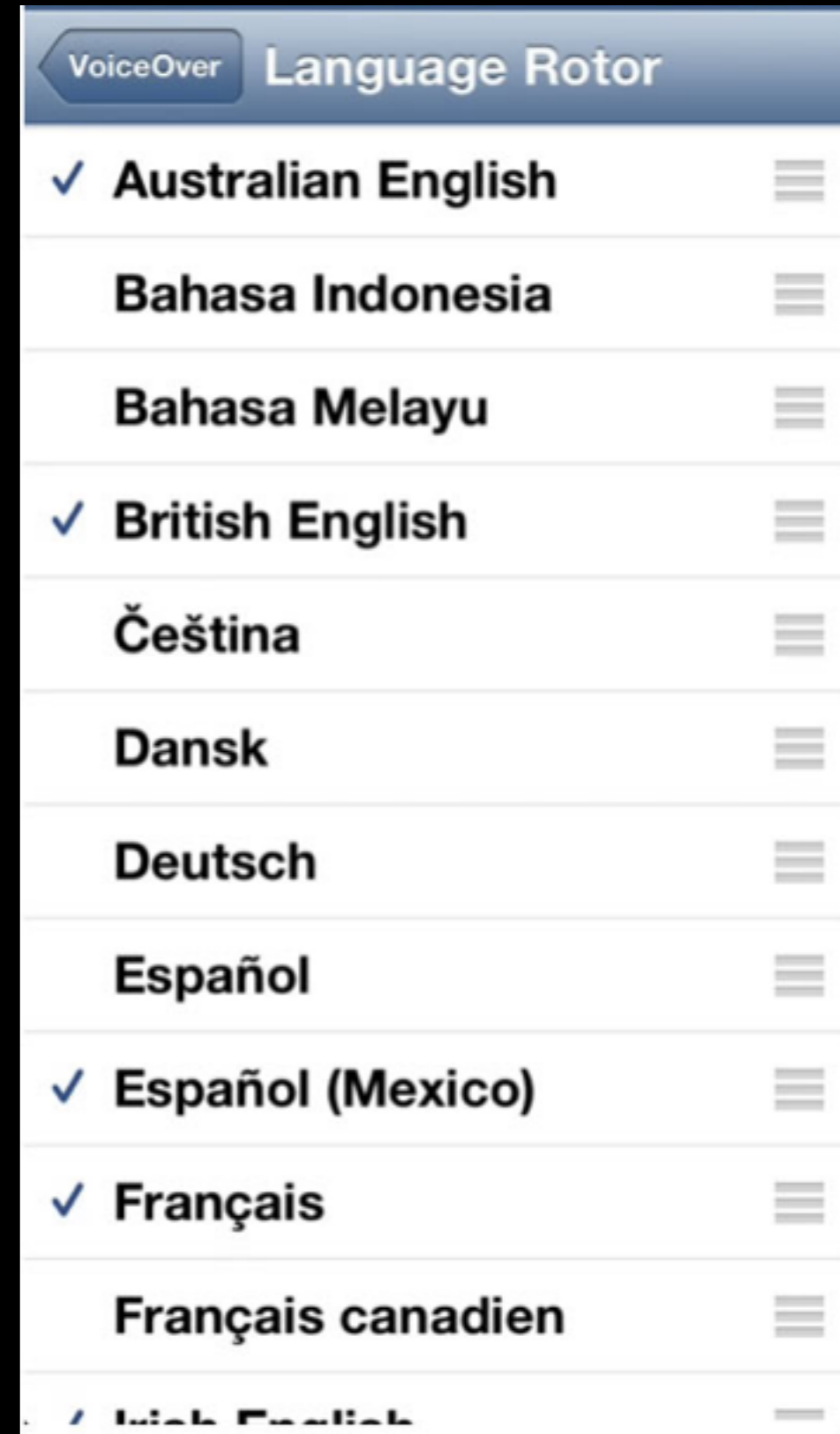
Use of Color & Contrast

- Placeholder text fails color contrast
- Only disabled controls can have bad contrast
- If error messages are Red then use `aria-invalid=true`
- Links (Hover & Focus)
- Keyboard Focus Outline
- Don't Let Design Trump Readability!

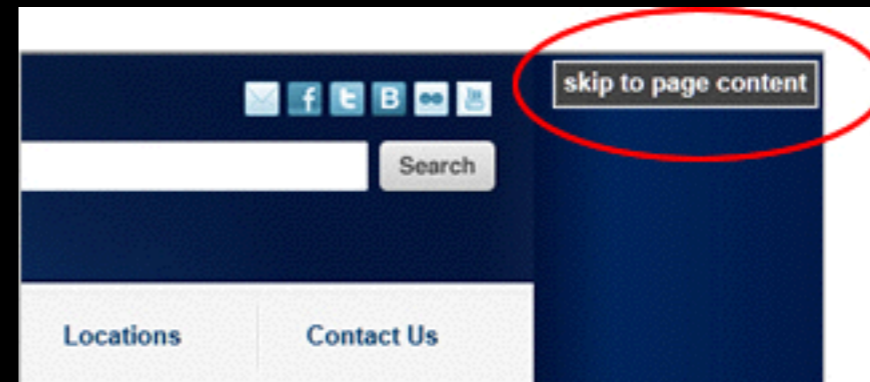


Language & i18n

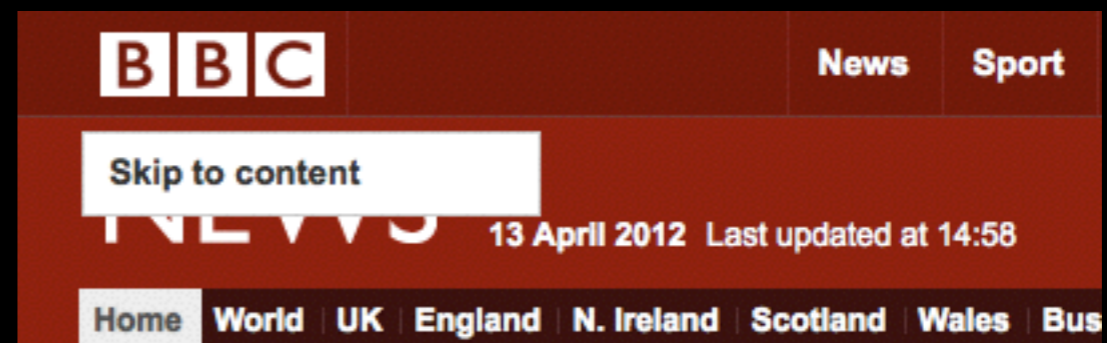
- **lang=en vs es**
- **<http://www.freedomscientific.com/training/Surfs-Up/Languages.htm>**
- **Desktop vs Mobile support**
- **Text-to-Speech API**
- **Controlling Verbosity with CSS**
- **<http://pauljadam.com/demos/css3speech.html>**



Skip Links



- In-Page Links
- Table of Contents
- Multiple Skip Links?
- <http://pauljadam.com/demos/inpagelinks.html>



Accessibility Testing with Mobile Screen Readers

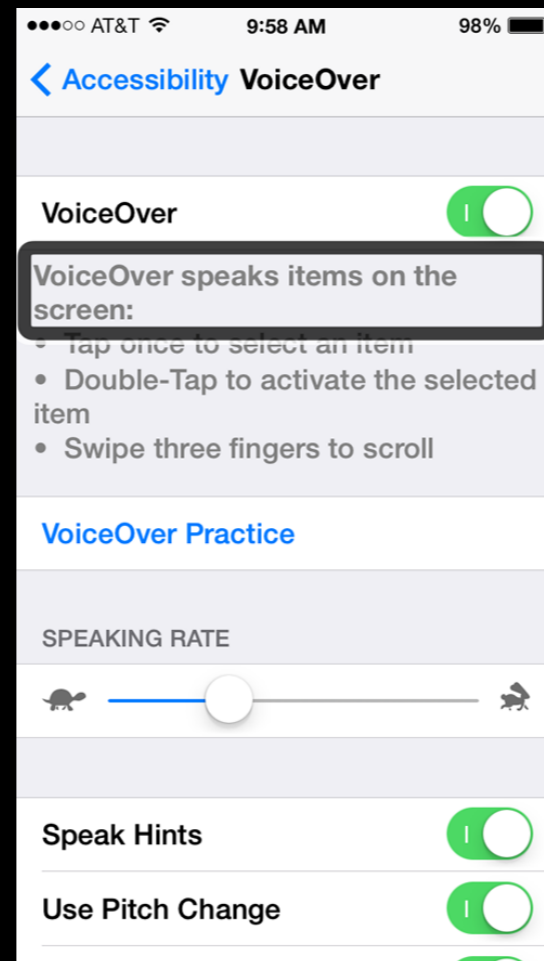


- **VoiceOver iOS Cheatsheet**
- **TalkBack Android Cheatsheet**
- **Mobile Accessibility QA Testing Checklist**



VoiceOver for iOS

- Gestures
- Rotor



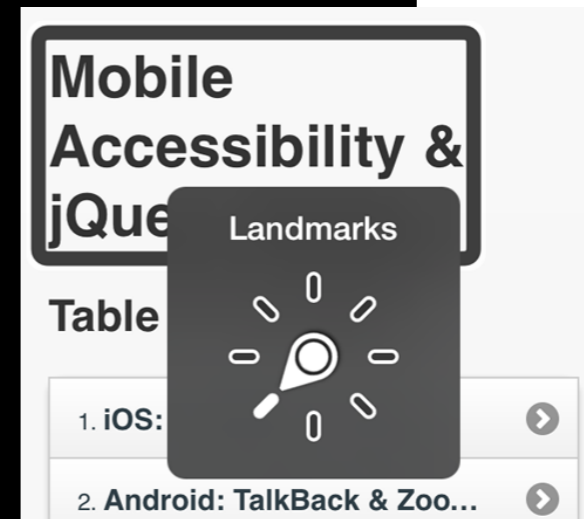
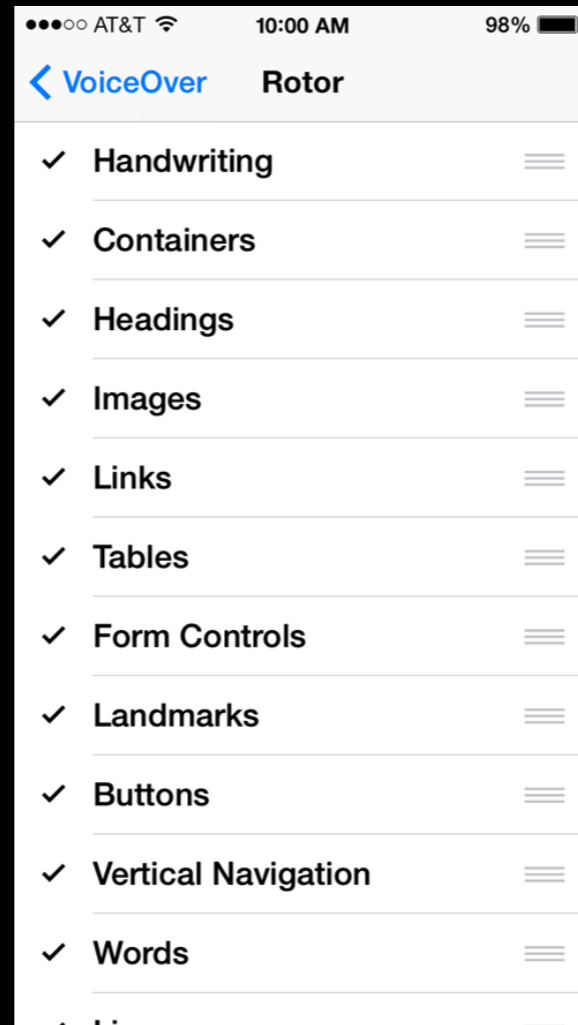
VoiceOver Keyboard Testing

- **VO Keys**
- **Quick Nav**
- **Single-Key Quick Nav**

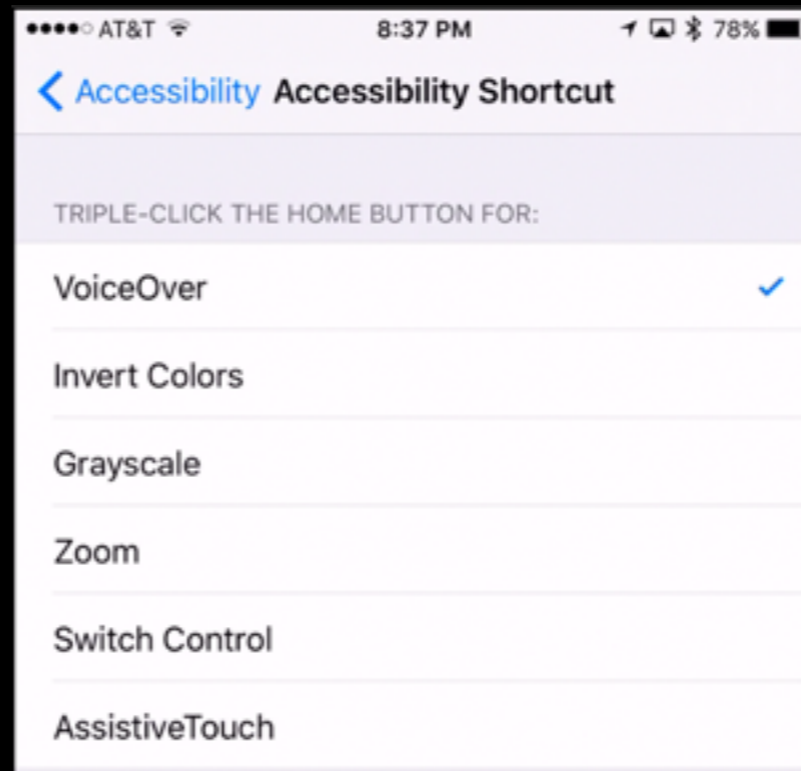


Data Table Testing on iOS

- **VoiceOver Rotor**
- **Rows**
- **Vertical Navigation**



Accessibility Shortcut (Triple-Click Home Button for VoiceOver)



- **Siri voice commands can also enable/disable VoiceOver.**

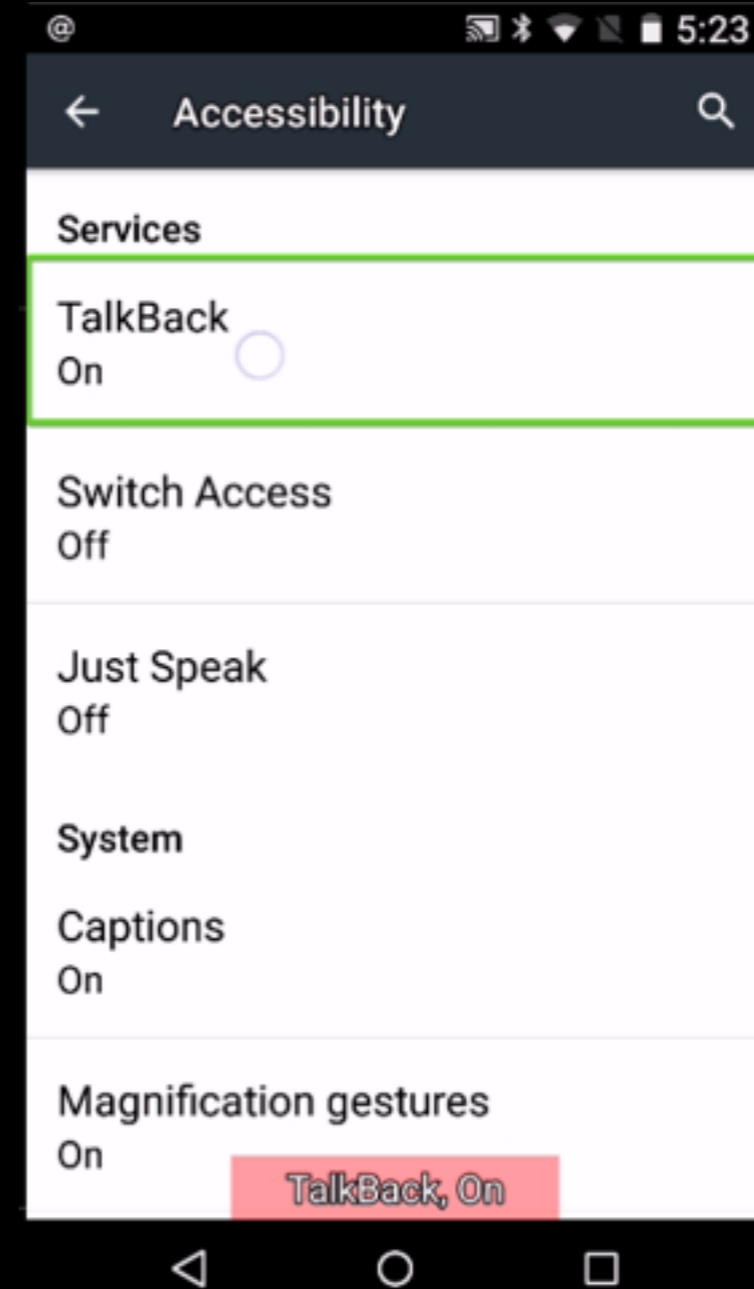
Virtual Keyboard Settings

- **Virtual Keyboard Settings
(Direct-Touch, Standard,
Touch Typing)**

TalkBack for Android

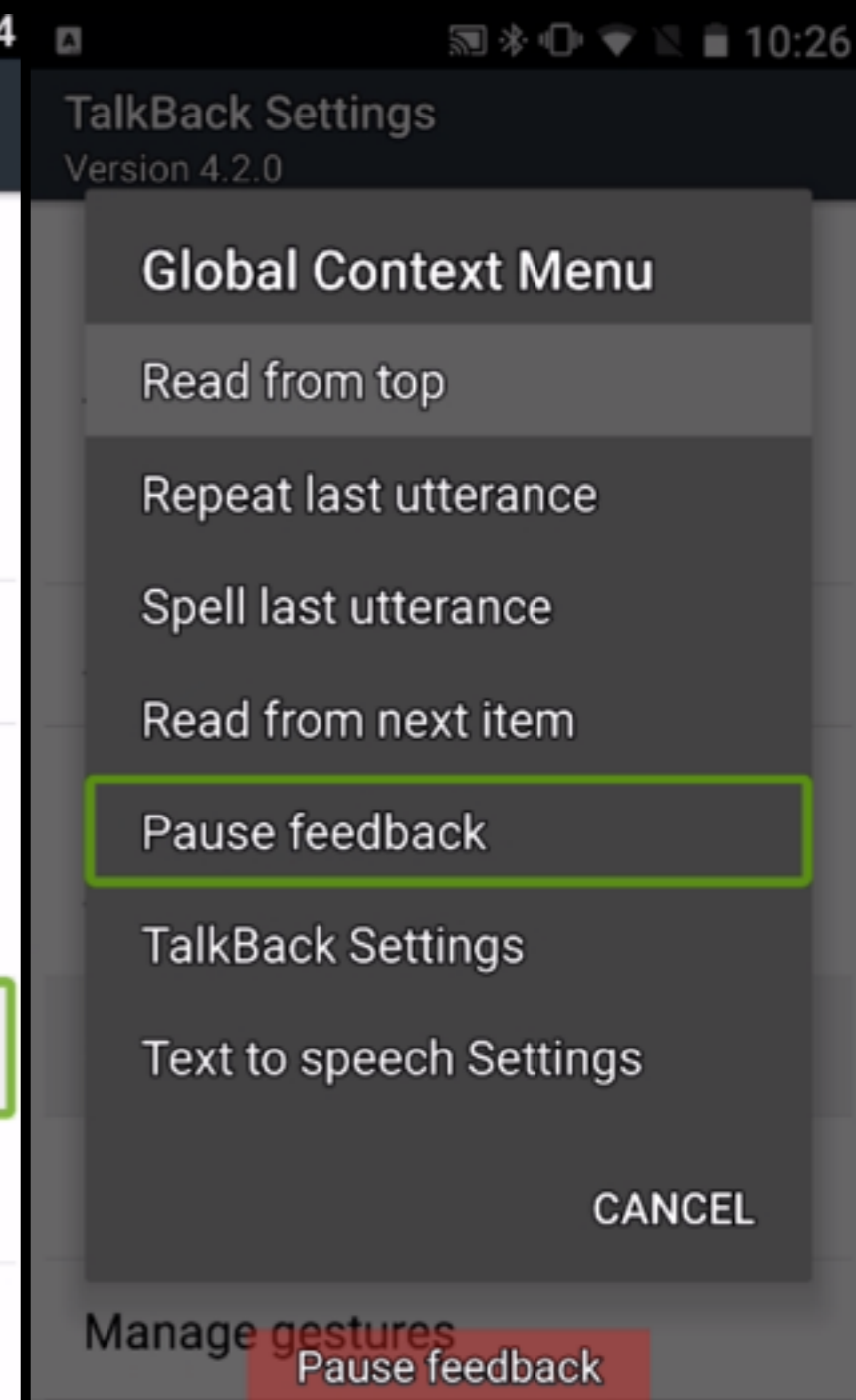
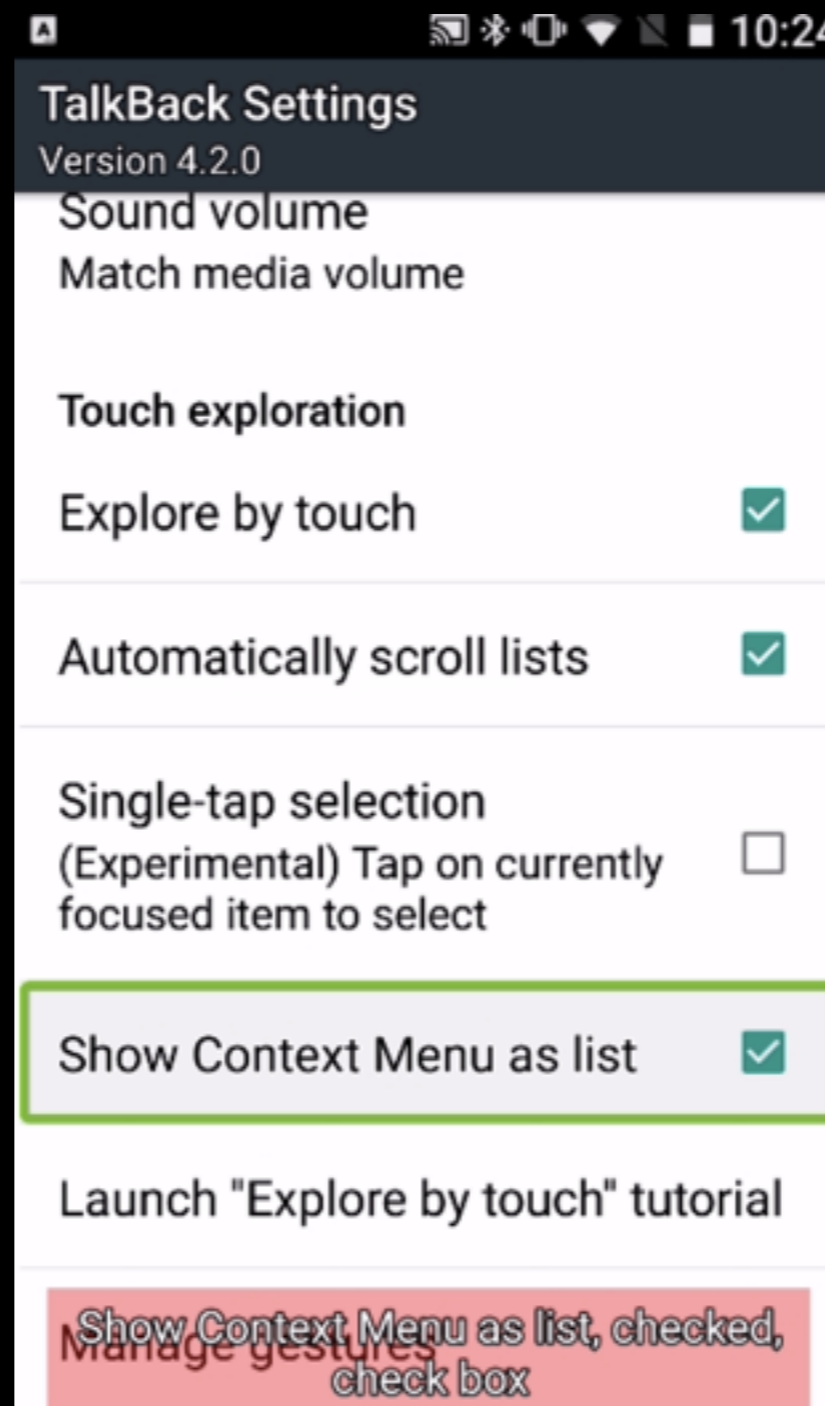


- **Explore by Touch**
- **Gesture Commands**



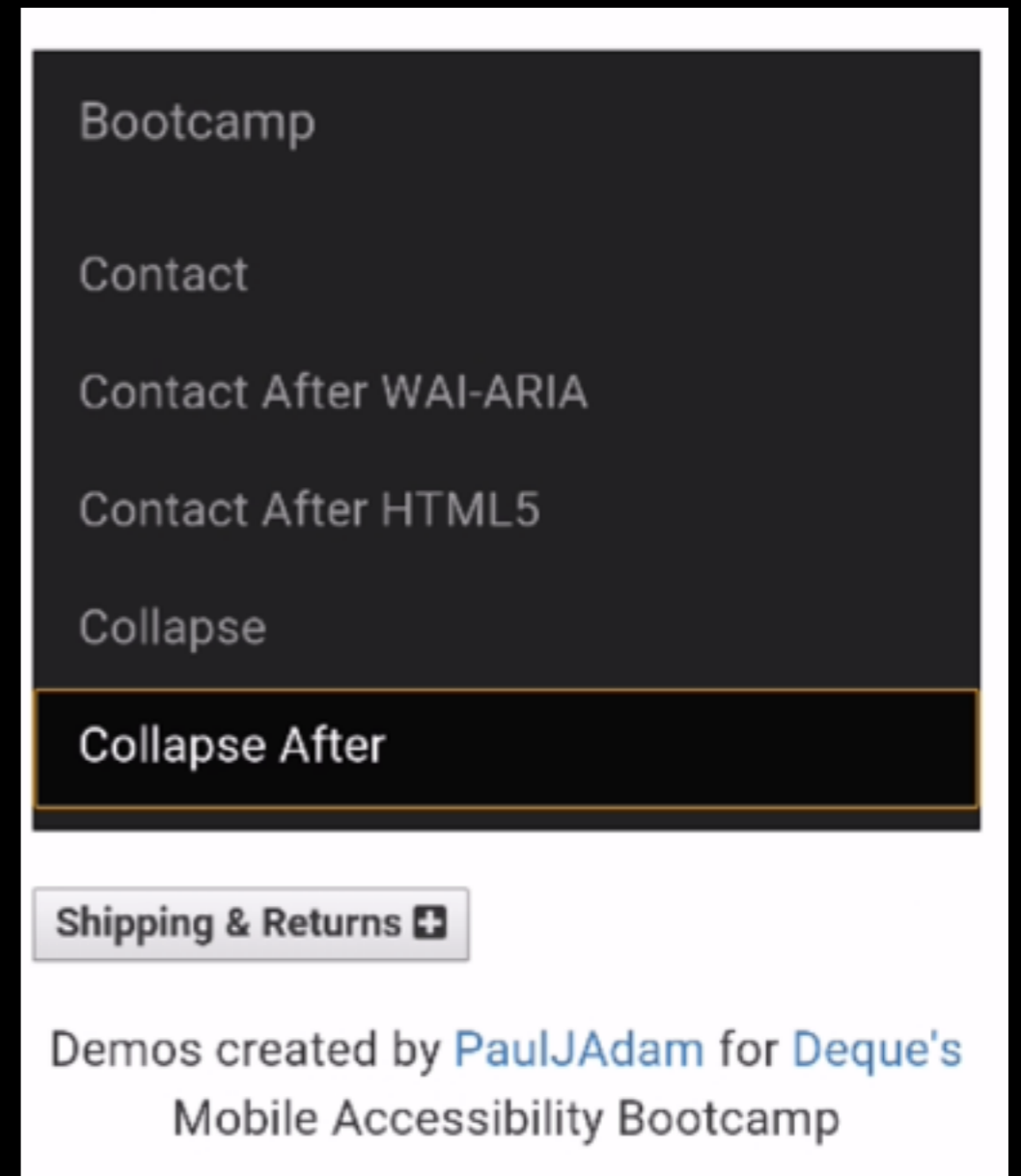
Global & Local Context Menus

- **Settings > Accessibility > TalkBack > Settings > Show Context Menu as list**



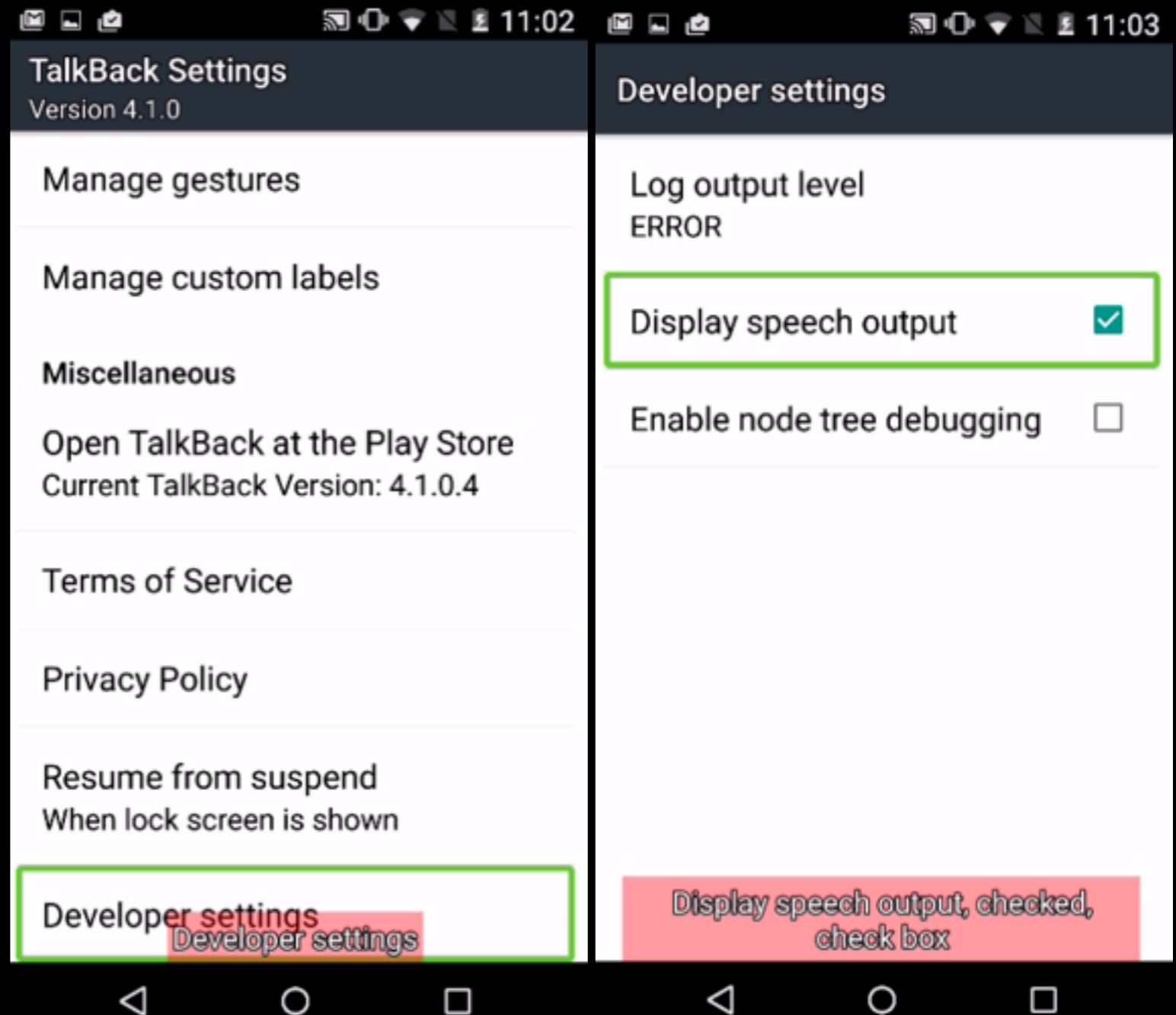
Android Keyboard Accessibility Testing

- **Differences from iOS**
- **Android Fragmentation, Nexus vs. Samsung (Android, TalkBack, & Browser Versions can be different)**



TalkBack Settings

- **Developer settings > Display speech output**
- **High Contrast Text**
- **Pause, Enable/Disable Talkback (Accessibility Shortcut)**



Firefox vs. Chrome

- Firefox Android (3-Finger Swipe Gestures, Keyboard & Quick Nav Keys)

10:32

pauljadam.com

Zip Code *

Website *

http://www.yourdomain.com

How Should We Get Back To You? *

Email

Phone

Mail

Message *

Question , Comment, etc.

Submit Contact Form

Demos created by PaulJAdam for Deque's Mobile Accessibility Rootcamp

not checked, Mail check box

10:31

pauljadam.com/bootcamp

Zip Code *

Website *

http://www.yourdomain.com

How Should We Get Back To You? *

Email

Phone

Mail

Message *

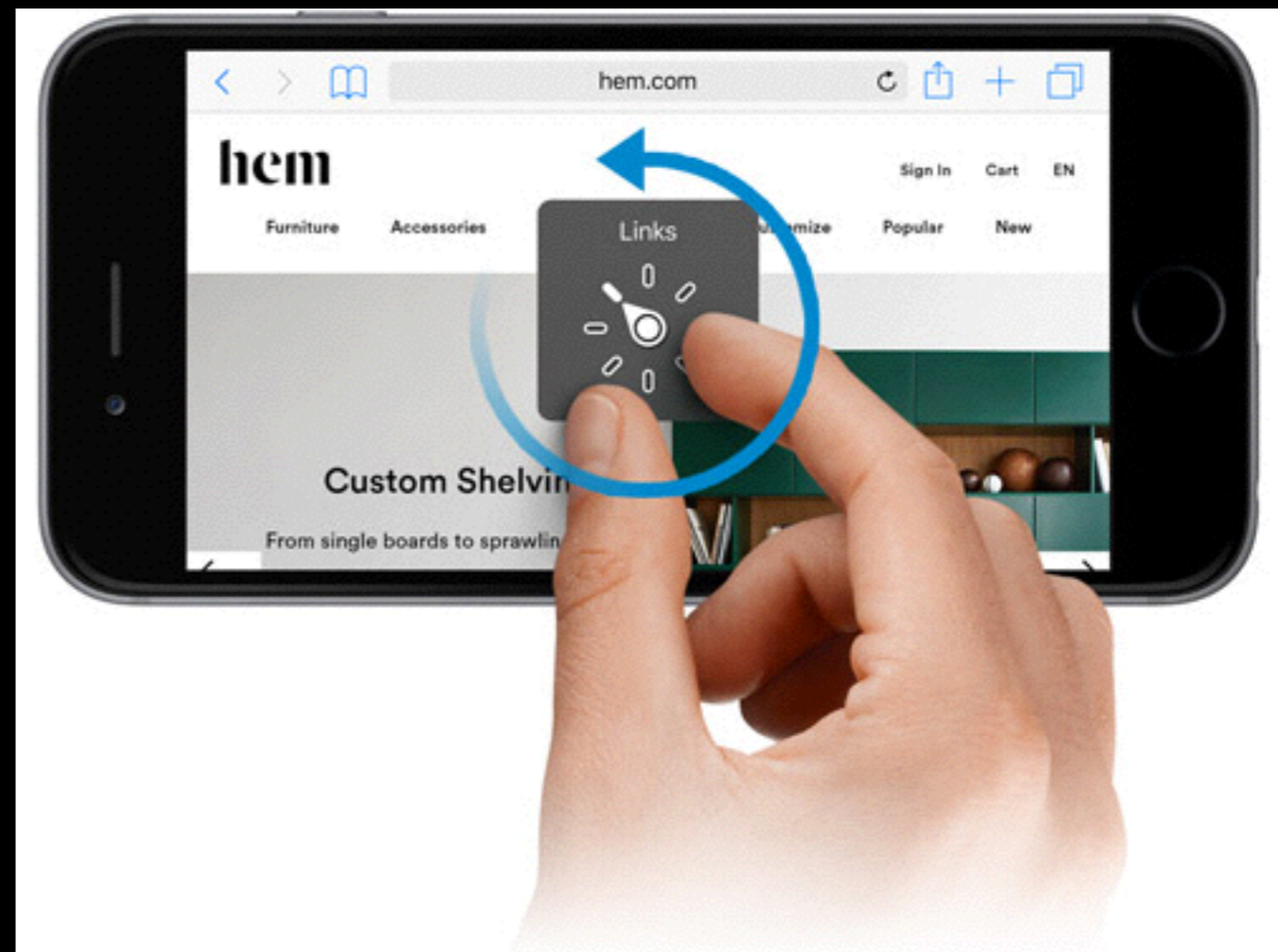
Question , Comment, etc.

Submit Contact Form

not checked, Mail check button How Should We Get Back To You? * required

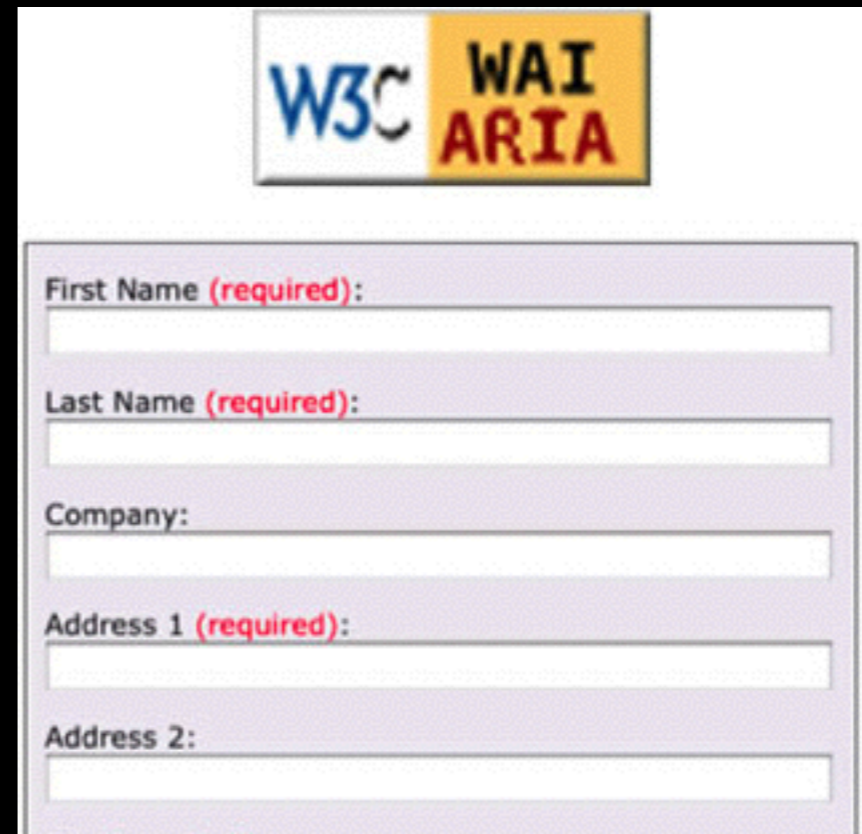
Hands-On Mobile Manual Screen Reader Testing Exercise

- **Mobile Accessibility QA Testing Checklist**
- **Test Before and After Demos with iOS VoiceOver or Android TalkBack or Both**
- **<http://www.washington.edu/accesscomputing/AU/>**
- **<http://www.w3.org/WAI/demos/bad/>**
- **<http://fyvr.net/acme>**



JavaScript Accessibility

- Use JavaScript to manage keyboard focus and update accessibility attributes of the user interface as it dynamically changes.



The image shows a screenshot of a web form with a light purple background. At the top right, there is a logo for W3C WAI ARIA. The form contains five input fields with the following labels: "First Name (required):", "Last Name (required):", "Company:", "Address 1 (required):", and "Address 2:". Each label is followed by a white input box. The "required" text is in red.

Focus Management

- **NEVER USE `tabindex="1+"`** (Always let the source code order dictate the focus & reading order)
- **`tabindex="0"` to make HTML elements keyboard focusable**
- **`tabindex="-1"` to make elements focusable with JS sending the focus to that element using `focus()`**



HTMLElement.focus()

IN THIS ARTICLE

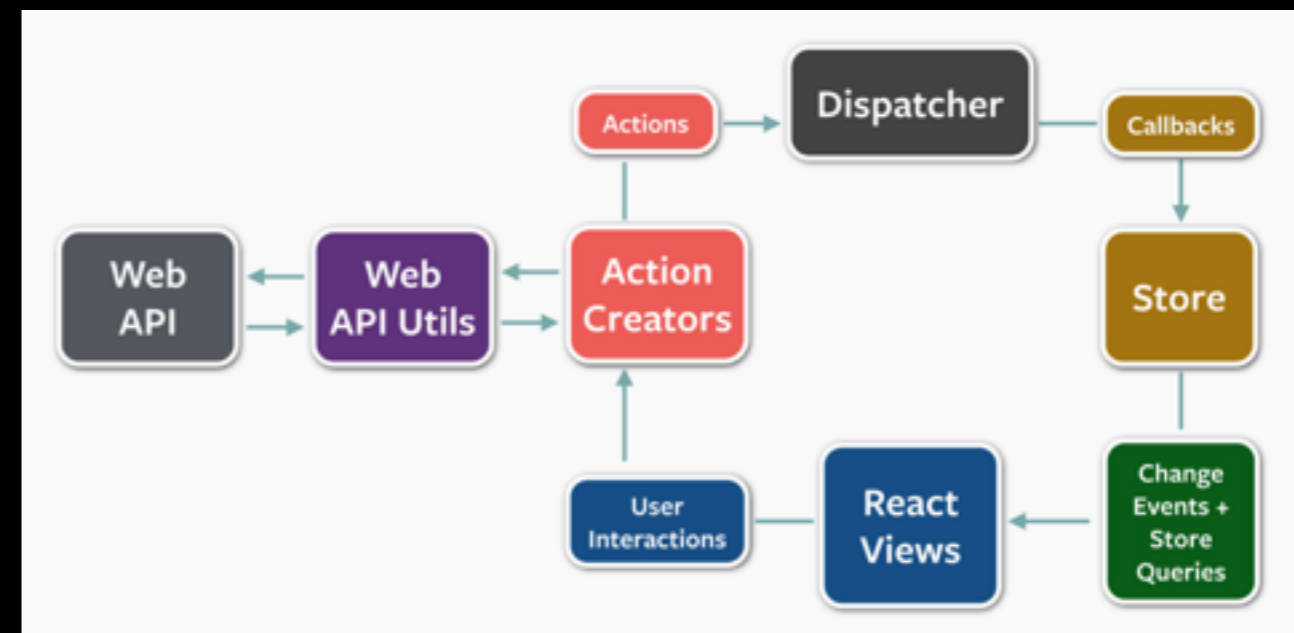
The `HTMLElement.focus()` method sets focus on the specified element, if it can be focused.

Syntax

```
element.focus()
```

JavaScript Frameworks & Single Page Applications

- jQuery & jQuery UI easy to make accessible
- AngularJS is improving and has some automatic framework fixes for custom controls e.g. ng-aria
- ReactJS, you must write semantic, standards-compliant HTML code e.g. `<label for>` `<input id>`
- Single Page Apps require `.focus()` Management & Unique Page `<title>`s



WAI-ARIA Accessibility

- **Web Accessibility Initiative – Accessible Rich Internet Applications**
- **<http://www.w3.org/TR/wai-aria/>**
- **Authoring Practices: <http://www.w3.org/TR/wai-aria-practices/>**



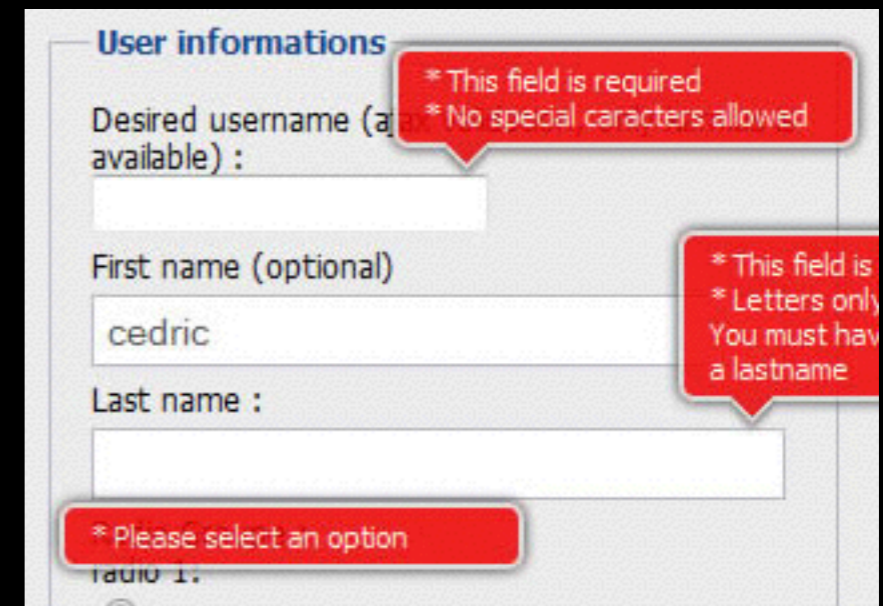
ARIA Navigation Landmarks

- pauljadam.com/demos/landmarks.html
- `role=banner`
- `role=search`
- `role=navigation`
- `role=main`
- `role=complementary`
- `role=contentinfo`
- `<nav role=navigation aria-label="Unique Container Name">`

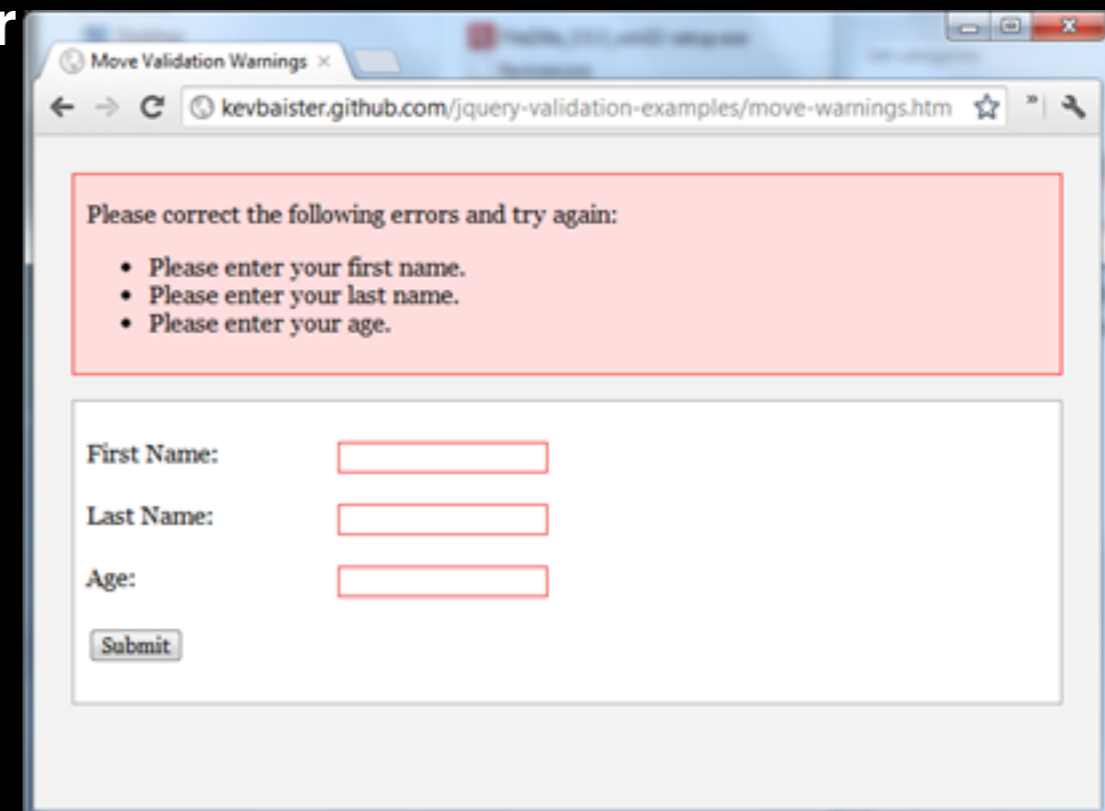
The screenshot displays a web browser interface for a demo site. The browser's address bar shows 'pauljadam.com'. The page title is 'WAI-ARIA Landmarks Site Navigation Structure Demo'. The page content is color-coded to show ARIA landmarks: a light green banner at the top, a light blue navigation bar, a light blue main content area, a light yellow complementary area, and a light orange footer. The main content area features a product card with a 'Landmarks' icon. The complementary area contains 'Related Products' with 'Add To Cart' buttons. The footer contains 'Company', 'Resources', and 'Customer Service' sections.

Enhancing Forms Accessibility with ARIA

- (aria-required, aria-invalid, aria-describedby, aria-labelledby)
- HTML5 Input Type Validation Works!
- <http://pauljadam.com/demos/mobileforma11y.html>
- JavaScript DOM Manipulation to Inject Error Messages & Send `.focus()`;
- Change WAI-ARIA Attributes
- Simple Form Validation Using WAI-ARIA `aria-describedby`, `aria-required`, `aria-invalid`, & jQuery `.focus()` <http://pauljadam.com/demos/aria-describedby-validation.html>



The screenshot shows a form titled "User informations" with three input fields. The first field, "Desired username (available)", has a red error message: "* This field is required" and "* No special characters allowed". The second field, "First name (optional)", contains the text "cedric" and has a red error message: "* This field is required" and "* Letters only. You must have a lastname". The third field, "Last name", is empty and has a red error message: "* Please select an option".



The screenshot shows a web browser window with the URL kevbaister.github.com/jquery-validation-examples/move-warnings.htm. The page displays a validation error summary in a red box: "Please correct the following errors and try again:" followed by a list of errors: "Please enter your first name.", "Please enter your last name.", and "Please enter your age.". Below the summary is a form with three input fields: "First Name:", "Last Name:", and "Age:". A "Submit" button is located at the bottom left of the form.

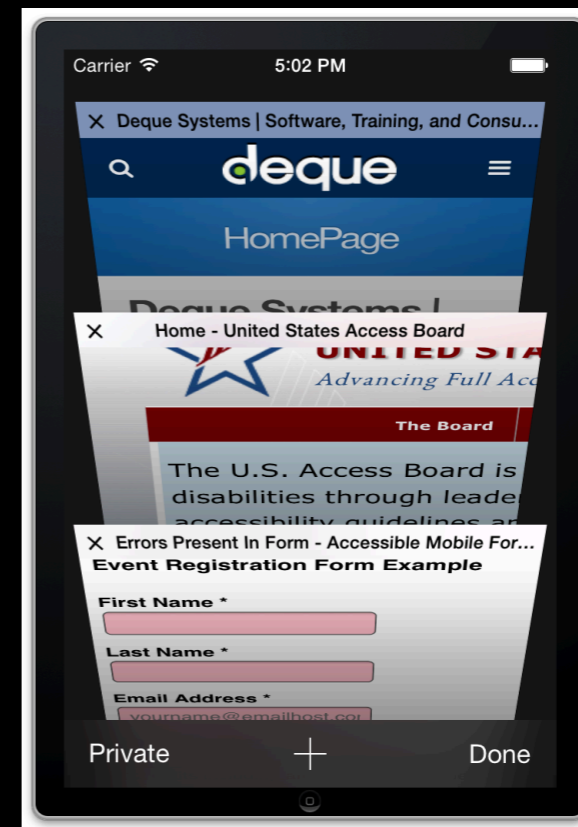
Speaking Messages to Screen Readers with WAI-ARIA Live Regions

- **role=alert**
- **aria-live="polite/assertive"**
- **Using ARIA Live Regions or role=alert to Identify Errors**
<http://pauljadam.com/demos/aria-alert-validation.html>
- **Live Error Messages**
- **Accessible Mobile Forms & Error Validation with HTML5, WAI-ARIA, & jQuery +(mobile)**
<http://pauljadam.com/demos/mobileforma11y.html>

Negative Words to Avoid in Your Form Error Messages	
Oops	Oops, something wasn't right
Error	✖ This Form Has Errors
Failed	Form submission failed!
Problem	There was a problem creating your account
Invalid	Invalid Fields
Wrong	Oops, something has gone wrong
Prohibited	3 errors prohibited this user from being saved

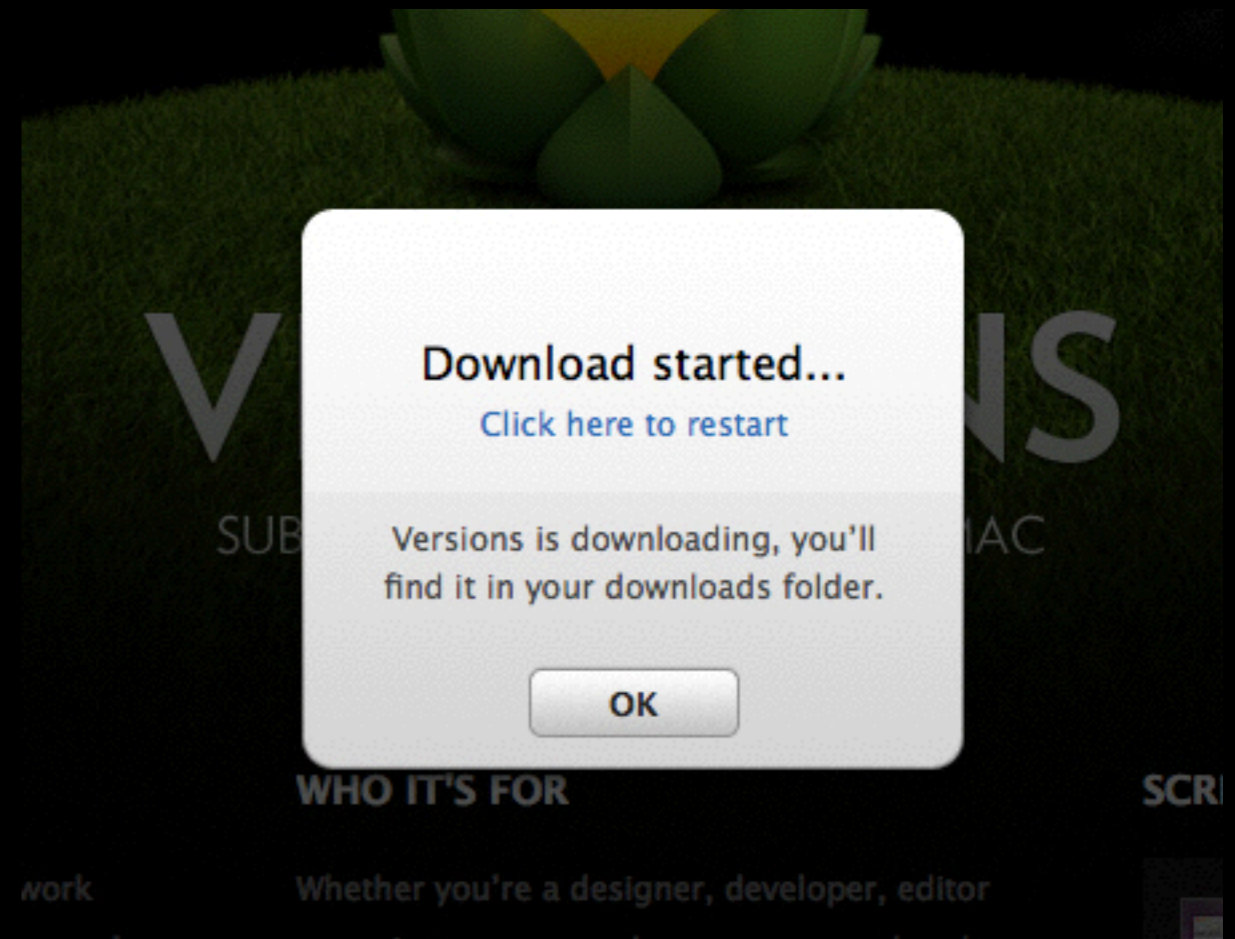
Highlighting Invalid Data & Error Messages in Page Title

- Styling Invalid & Valid Inputs with CSS
- `input[aria-invalid="true"]:required:invalid {background: pink;}`
- `input:required:valid {background: lightgreen;}`
- Error Message In Page Title
- `document.title = "Errors Present In Form - " + oldTitle;`



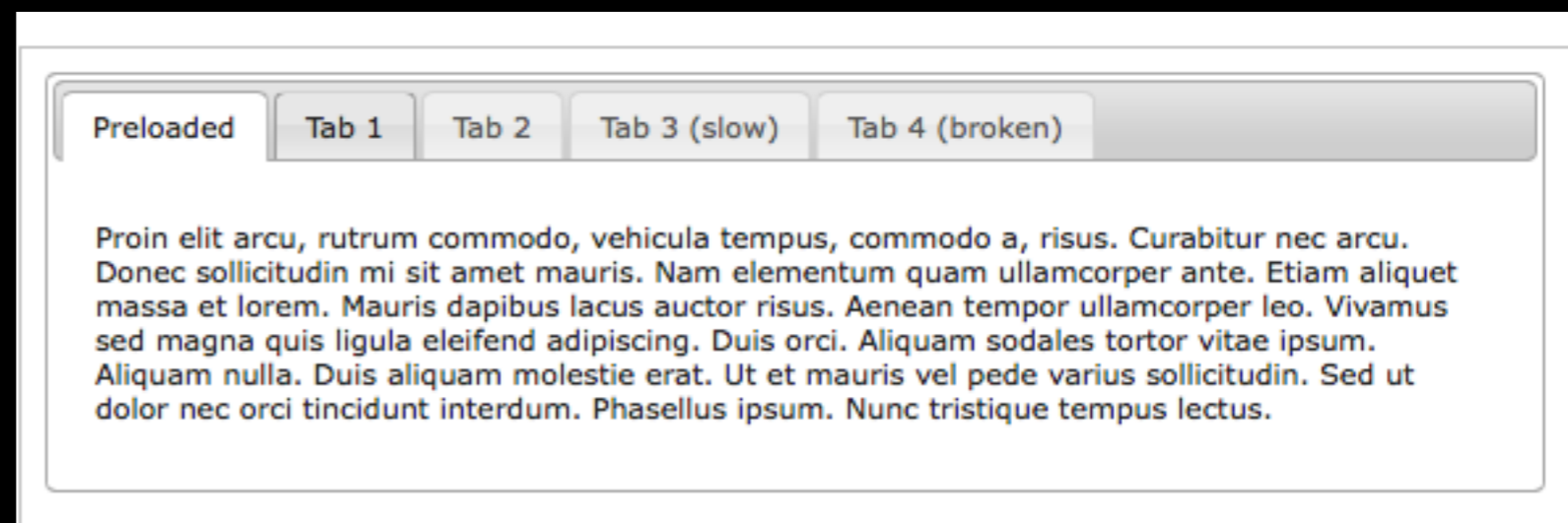
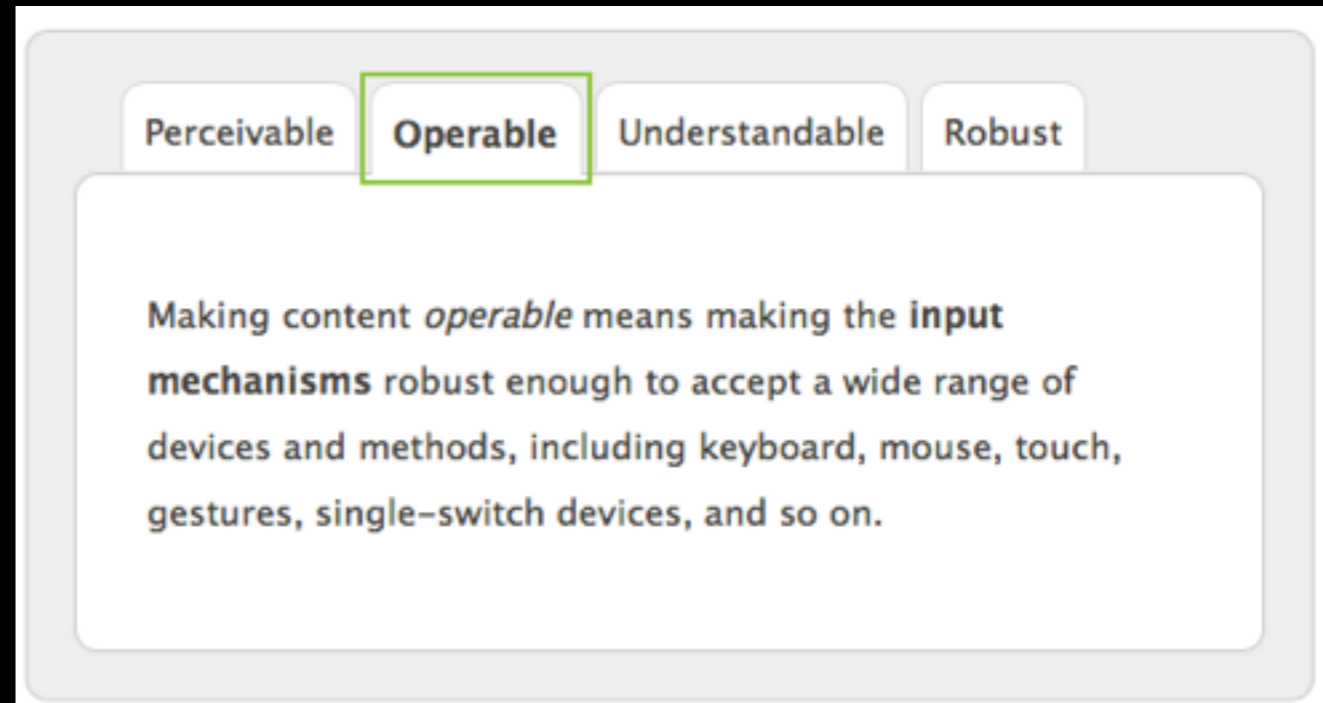
Accessible Alert & Modal Dialogs with ARIA

- **role=dialog, role=alertdialog, aria-labelledby, aria-describedby, aria-hidden**
- <http://pauljadam.com/demos/dialog.html>
- <http://pauljadam.com/demos/aria-role-alertdialog.html>
- <http://pauljadam.com/demos/dialog-DOM.html>

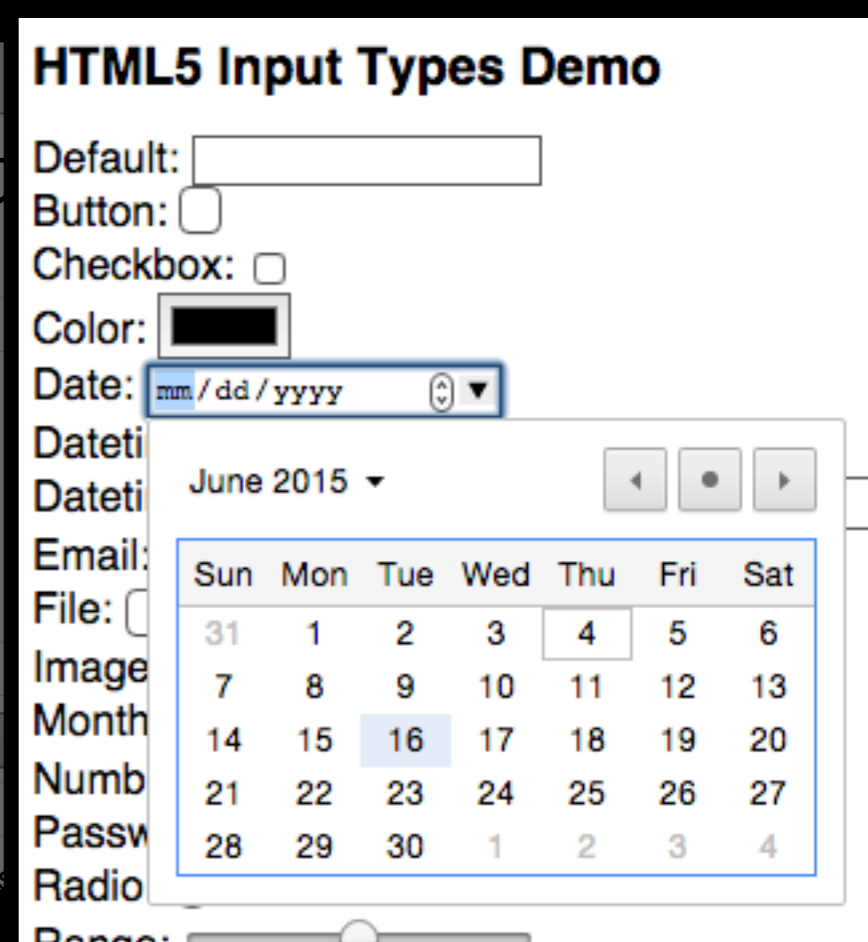
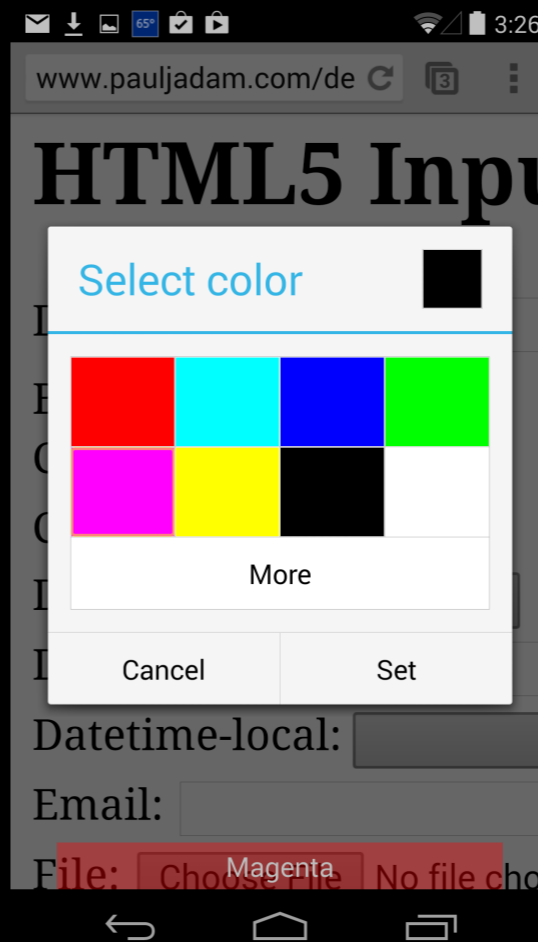
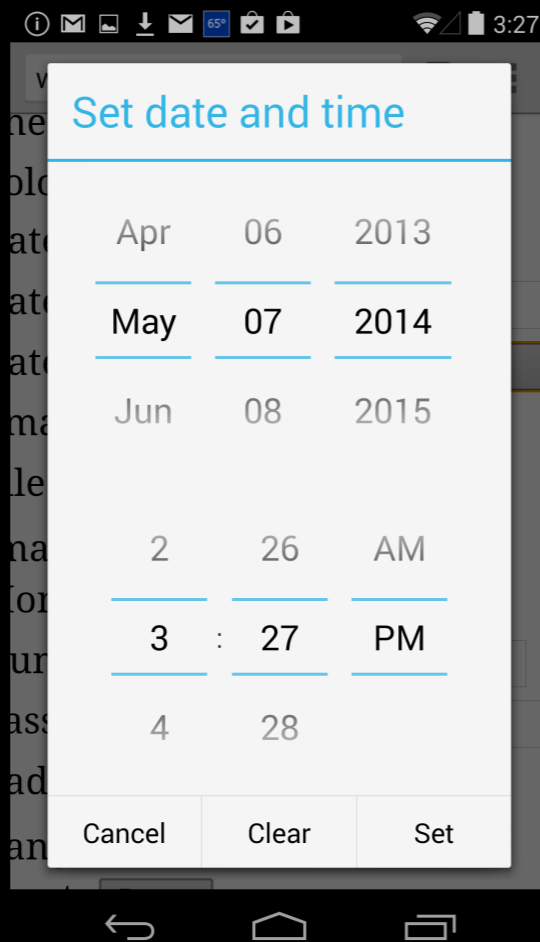
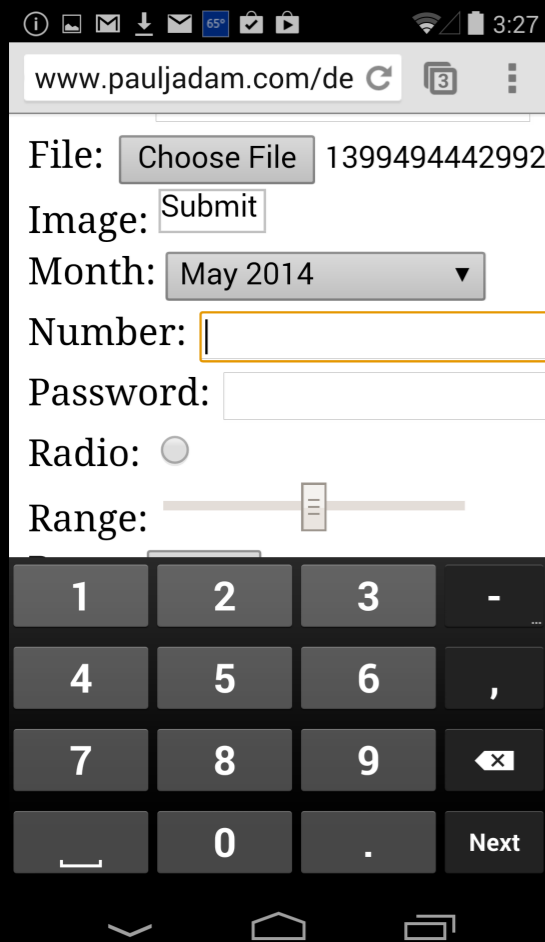


Tabs and Expandable/ Collapsible Widgets

- (aria-expanded, role=tab, aria-selected)
- <http://jqueryui.com/tabs>
- <http://pauljadam.com/demos/aria-expanded.html>
- <https://dequeuniversity.com/library/aria/tabpanels-accordions/tabpanel>



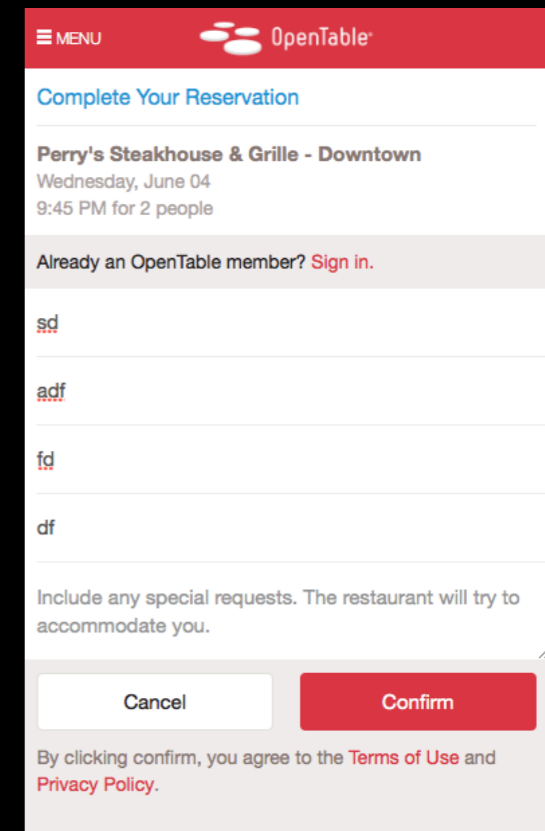
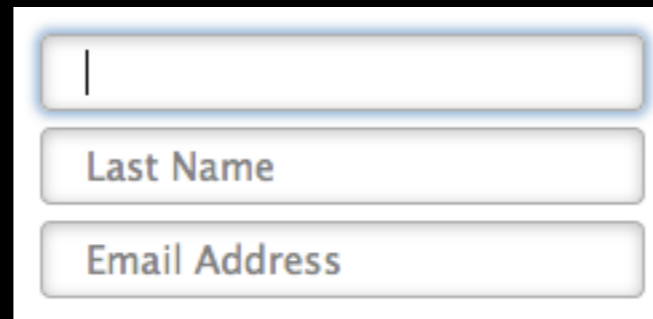
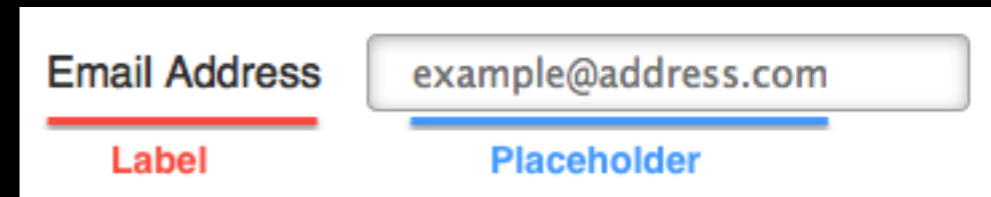
HTML5 Input Types



<http://pauljadam.com/demos/html5-input-types.html>

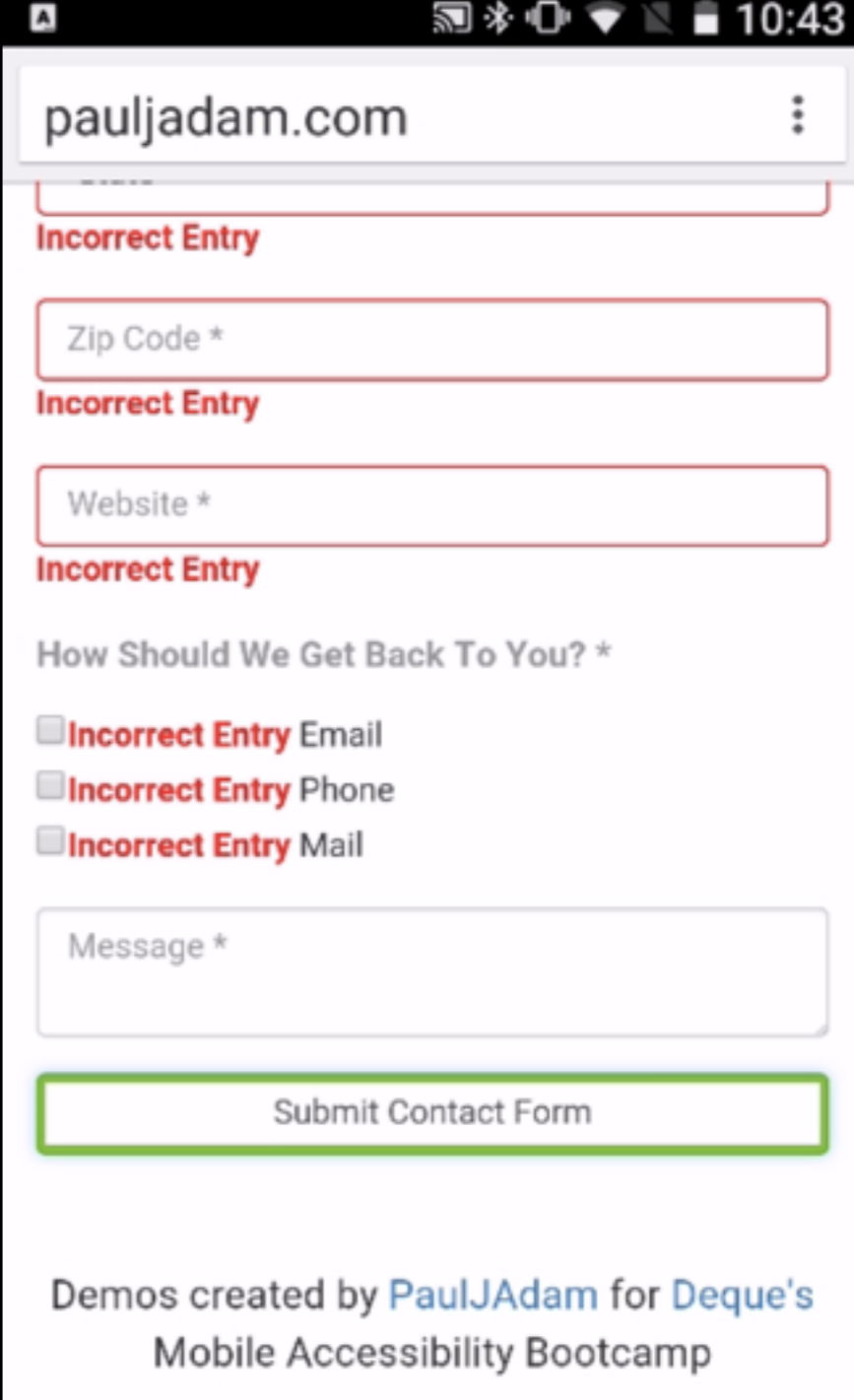
Labels vs. Placeholders

- **Placeholders Disappear On Focus & After Typing**
- **Light Gray Text Fails Contrast**
- **Problem for cognitive accessibility and general usability.**
- **Top Align <Label>s for Mobile**
- **Placeholders are for Examples Of Input Values**



Hands-On ARIA Landmarks & Forms Exercise

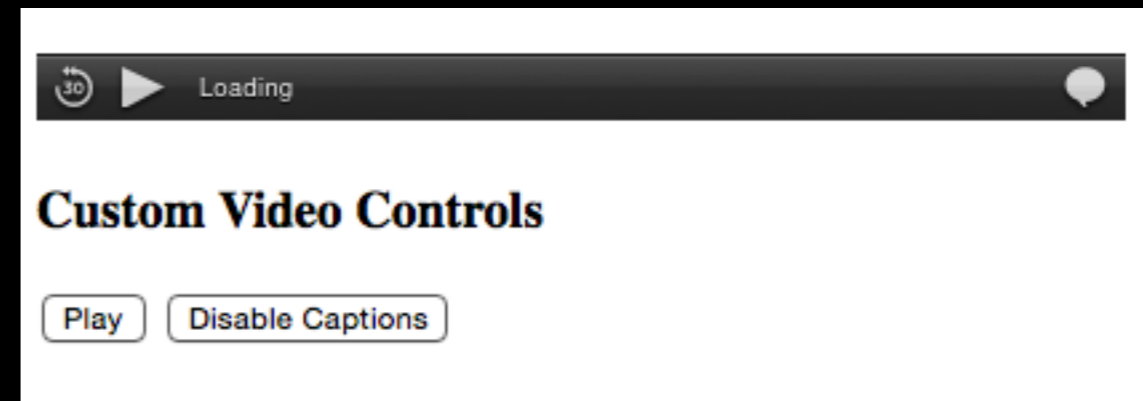
- Add the navigation, main, form, and contentinfo landmarks.
- Add `<labels>`
- Add `aria-required=true`
- Fix heading, fix contrast
- Re-enable Pinch-to-zoom
- Add `<fieldset>` `<legend>`



The screenshot shows a mobile contact form on a browser. The address bar displays 'pauljadam.com'. The form contains several input fields, each with a red border and the text 'Incorrect Entry' below it. The fields are: a name field (empty), a 'Zip Code *' field, a 'Website *' field, and a 'Message *' field. Below the message field is a 'Submit Contact Form' button with a green border. At the bottom, there is a section titled 'How Should We Get Back To You? *' with three radio button options: 'Incorrect Entry Email', 'Incorrect Entry Phone', and 'Incorrect Entry Mail'. The footer text reads 'Demos created by PaulJAdam for Deque's Mobile Accessibility Bootcamp'. The top status bar shows the time as 10:43 and various system icons.

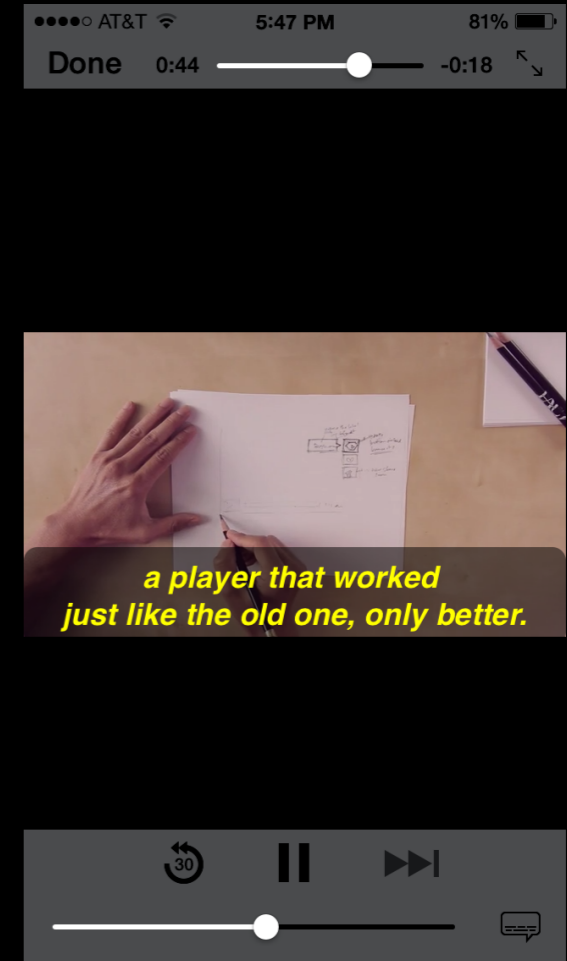
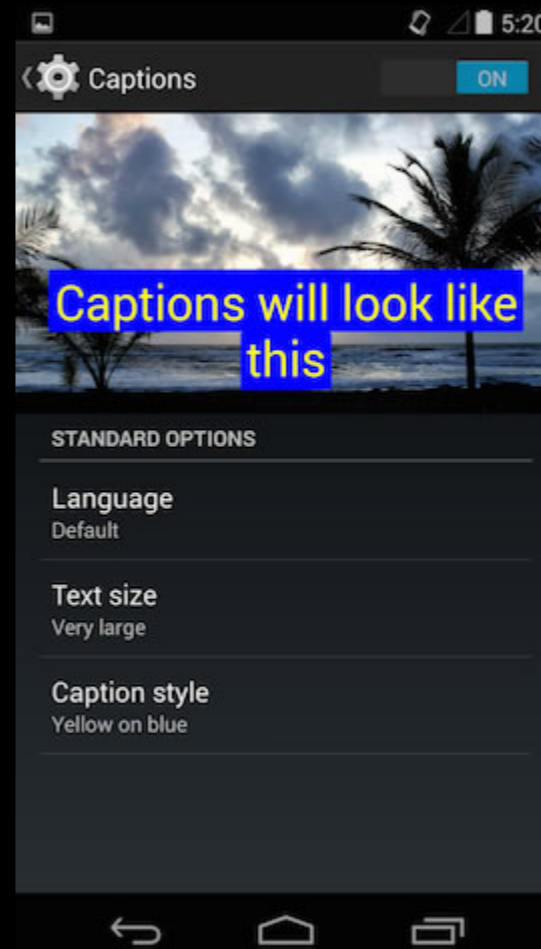
HTML5 Video Keyboard & Mobile Accessibility

- Native `<video>` elements are not accessible, the Play button, etc. is not keyboard or screen reader operable.
- Create custom HTML `<button>` based media player controls that play the `<video>` through JavaScript and is keyboard accessible.
- <http://pauljadam.com/demos/html5-video-a11y.html>



Accessible Video Captions (WebVTT) & Audio Descriptions

- **HTML5, Web Video Text Tracks Format, ARIA Live Regions**
- **User Customizable Closed Captions**
- **Android 4.4+**
- **iOS 7.0+**



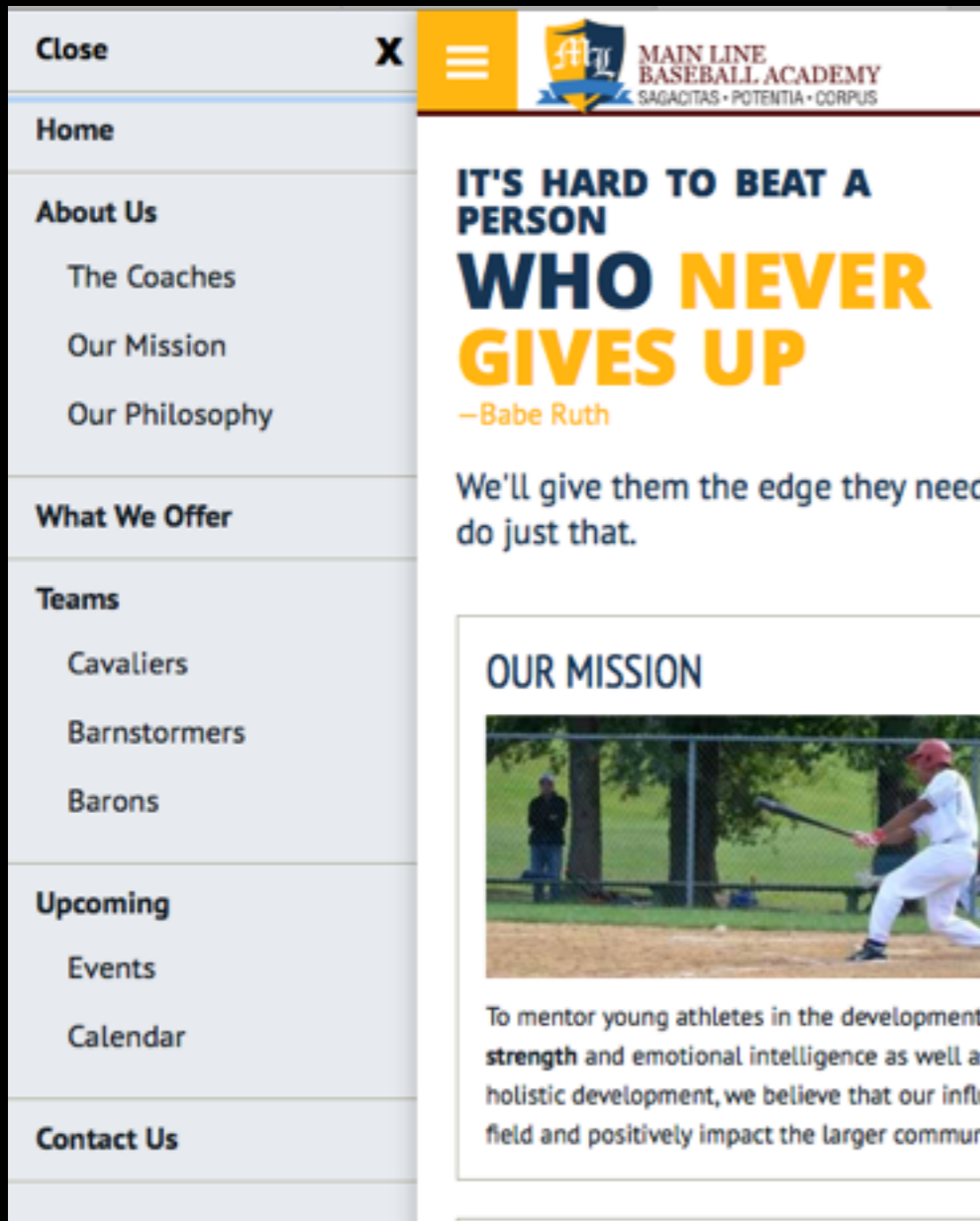
Hands-On HTML5 Forms Exercise



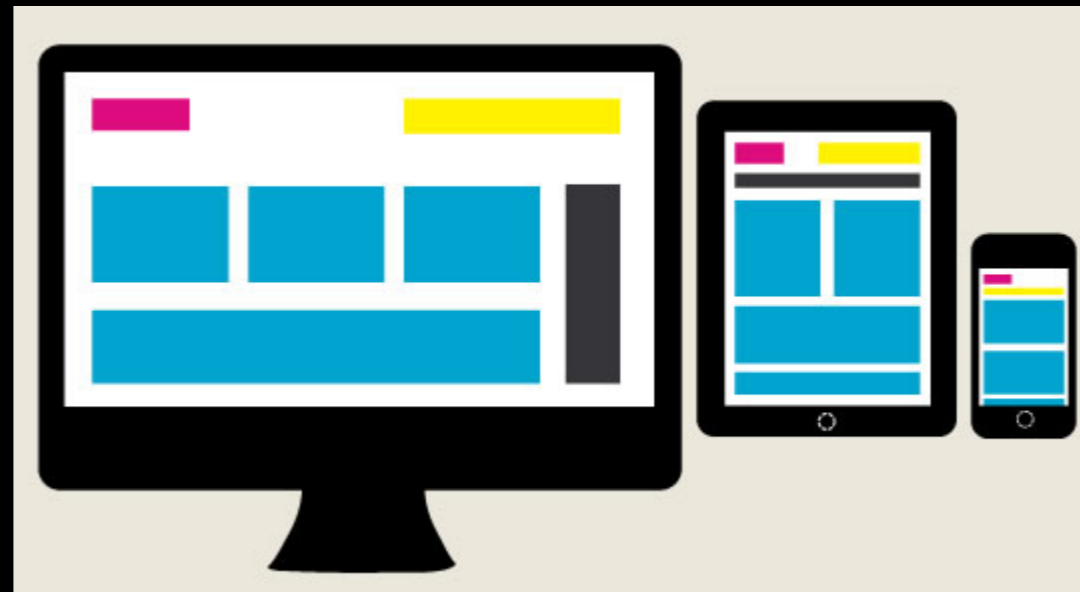
- **Add input types, type=tel, type=email, etc.**
- **Replace aria-required with HTML5 required**
- **Add autofocus to first input**

A screenshot of a mobile browser displaying a contact form on the website pauljadam.com. The form includes fields for Phone #, Street Address, Apt, Suite or Floor (Optional), and City. A numeric keypad is visible at the bottom of the screen, indicating that the Phone # field is currently active. The browser's address bar shows the URL pauljadam.com and the time is 11:18. The status bar at the top shows various system icons and the time 11:18.

Responsive Web Design



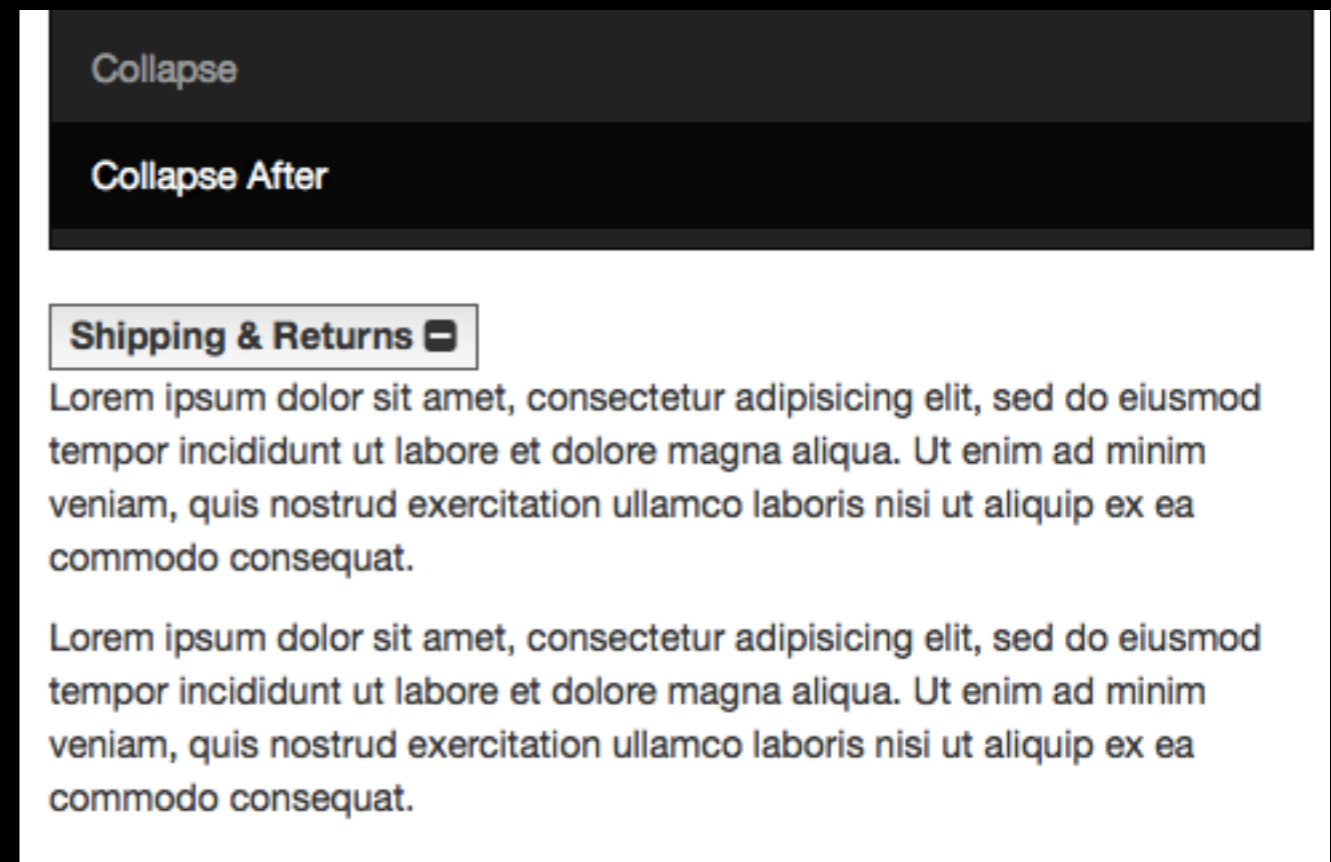
<http://www.mainlinebaseball.com>



- **Challenge! Find the class a website with accessible RWD. Test with VoiceOver or TalkBack.**

Hands-on activity on Responsive design

- **Hands-On Expandable RWD Section Exercise**



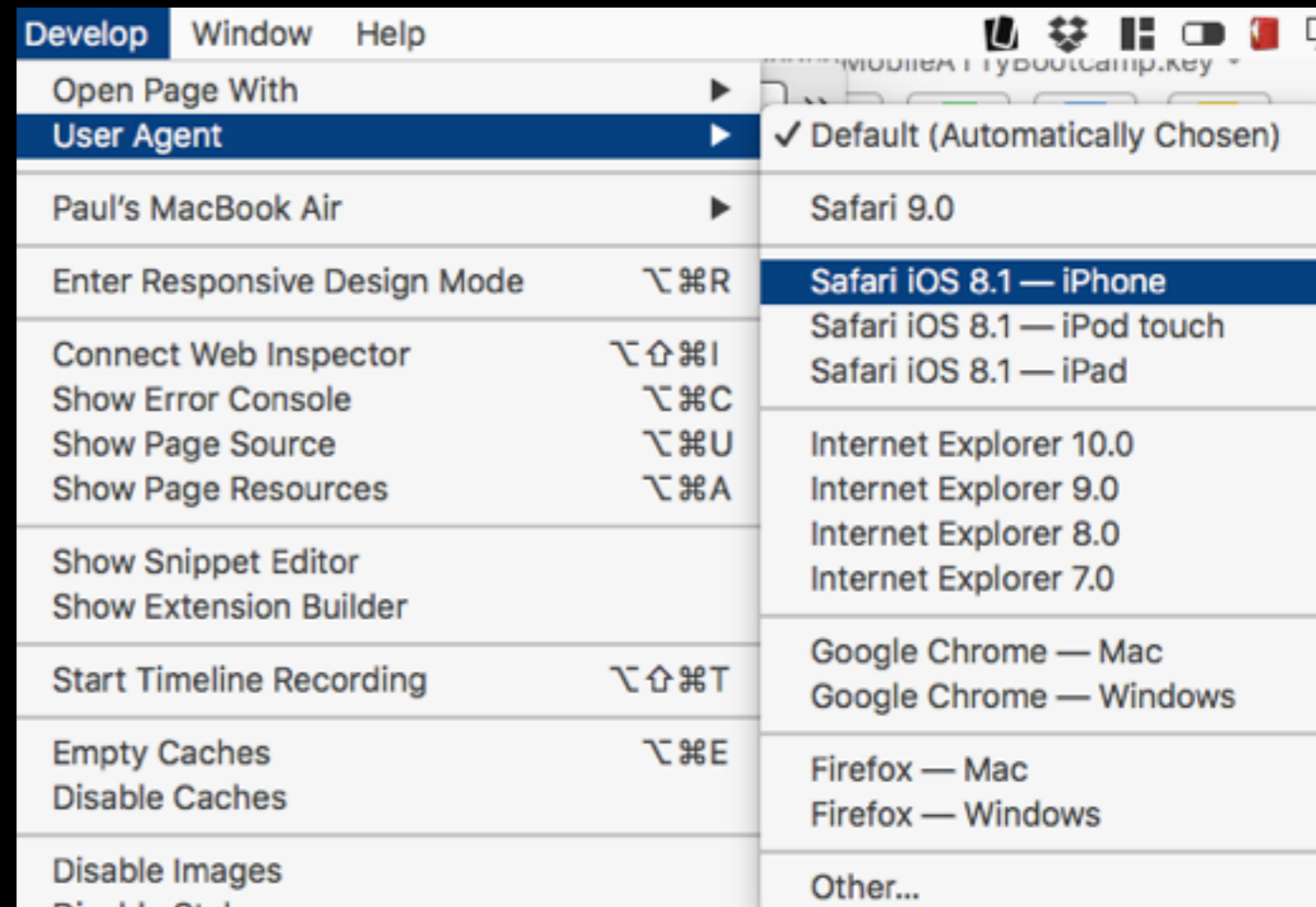
Mobile Accessibility Testing Tools

- **Browser Extensions**
 - **aXe (Firefox, Chrome)**
 - **FireEyes Standalone (User Agent Switching)**
 - **Chrome Accessibility Developer Tools**
 - **WAVE Evaluation Tool for Chrome**



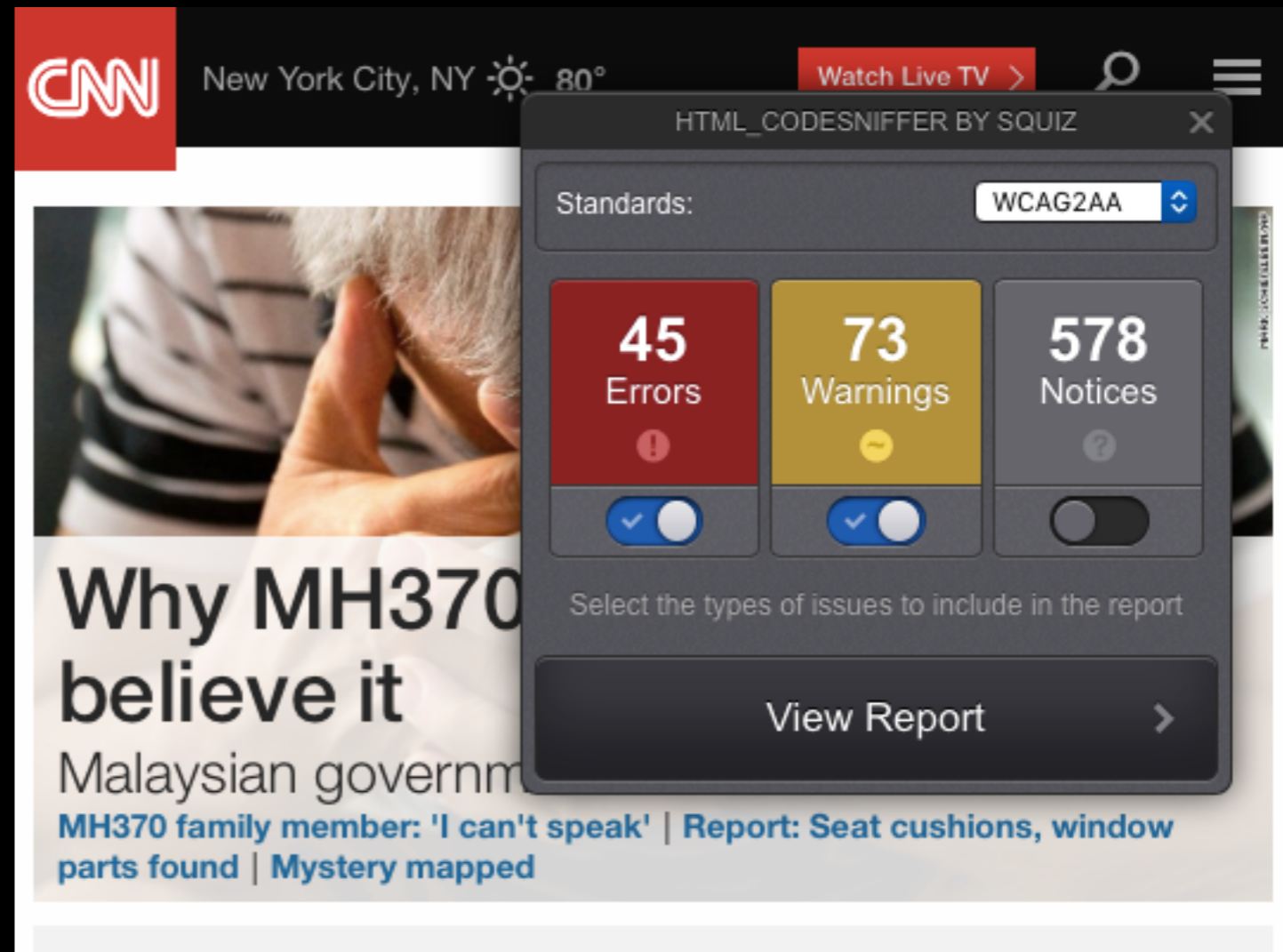
Mobile User Agents & Web Inspecting

- **Safari OS X User Agent Switching**
- **Mobile Web Inspecting to Debug Accessibility Issues**



JavaScript Bookmarklets

- **Safari + ALL Browsers**
- **HTML_CodeSniffer**
- **Jim Thatcher's Favelets**
- **PaulJAdam's Bookmarklets for Accessibility Testing**
- **tota11y**
- **NC State Web Evaluation Tools Bookmarklet**
- **accessibility-bookmarklets.org**

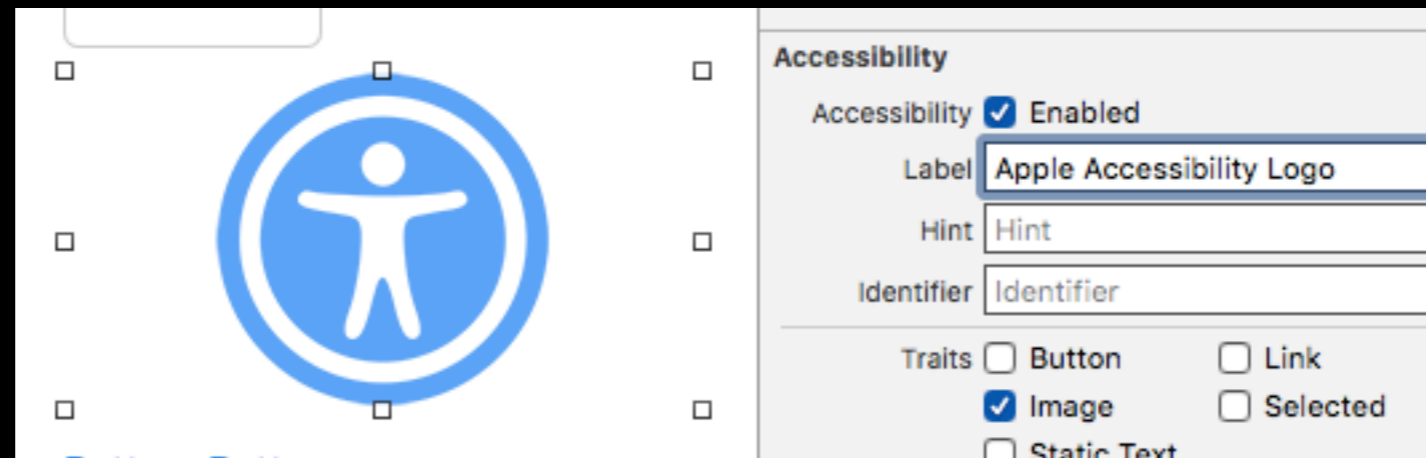


Hands-On Activity Test Before & After Demos with Testing Tools

- <http://www.washington.edu/accesscomputing/AU/>
- <http://www.w3.org/WAI/demos/bad/>
- <http://fyvr.net/acme>

The screenshot shows the 'Accessible University' website with several accessibility errors highlighted. At the top, the title 'Accessible University' is enclosed in a red box. Below it, a navigation menu contains buttons for 'About AU', 'Academics', 'Admissions', and 'Visitors', each with an 'IMG NO ALT Error' label. A 'Welcome!' section features a photo of a person sitting on the grass, with a 'NO ALT' error label. Below the photo is a paragraph of text describing the fictional university. A horizontal line graphic below the text has an 'alt="horizontal line graphic"' label. A 'Bienvenido!' section features a photo of two people walking, with a 'NO ALT' error label. On the right side, there is a 'Apply Now!' form with fields for Name, Email, City, State/Province, Zip/Postal Code, and Country, each with an '<input Error>' label.

Native iOS App Accessibility



- developer.apple.com
- <https://developer.apple.com/accessibility/ios/>

UIAccessibility API

- (Label, Hint, Trait, etc.)
- <http://pauljadam.com/iosvsandroida11y/#native>

```
tableviewEditingModeButtonTest > Frameworks > UIKit.framework > Headers
When setting accessibility traits, combine custom traits with
[super accessibilityTraits]. An incorrect combination of custo
will cause accessibility clients to incorrectly interpret the
Use common sense when combining traits.
*/
typedef uint64_t UIAccessibilityTraits;

// Used when the element has no traits.
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitNone;

// Used when the element should be treated as a button.
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitButton;

// Used when the element should be treated as a link.
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitLink;

// Used when an element acts as a header for a content section
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitHeader N

// Used when the text field element should also be treated as a
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitSearchFi

// Used when the element should be treated as an image. Can be
example.
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitImage;

/*
Used when the element is selected.
For example, a selected row in a table or a selected button wi
*/
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitSelected

// Used when the element plays its own sound when activated.
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitPlaysSou

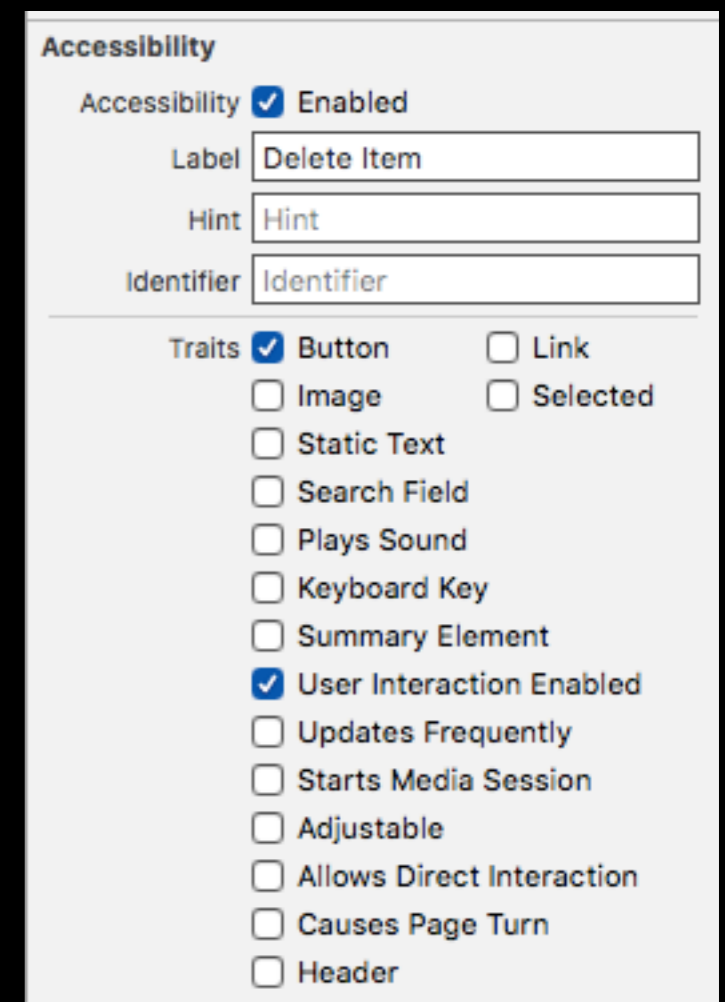
// Used when the element acts as a keyboard key.
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitKeyboard

// Used when the element should be treated as static text that
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitStaticTe

NSTextStorage.h
UIAccelerometer.h
UIAccessibility.h
UIAccessibilityAdditions.h
UIAccessibilityConstants.h
UIAccessibilityElement.h
UIAccessibilityIdentification.h
UIAccessibilityZoom.h
UIActionSheet.h
UIActivity.h
UIActivityIndicatorView.h
UIActivityItemProvider.h
UIActivityViewController.h
UIAlert.h
UIAlertView.h
UIAppearance.h
UIApplication.h
UIAttachmentBehavior.h
UIBarButtonItem.h
UIBarCommon.h
UIBarItem.h
UIBezierPath.h
UIButton.h
UICollectionView.h
UICollectionViewCell.h
UICollectionViewController.h
UICollectionViewFlowLayout.h
UICollectionViewLayout.h
UICollectionViewTransitionLayout.h
UICollisionBehavior.h
UIColor.h
UIControl.h
UIDataDetectors.h
UIDatePicker.h
```

Xcode Overview

- **Fixing Accessibility Issues in Interface Builder or Pure Code**
- **`lastNameTextField.accessibilityLabel = "Last Name"`**
- **`firstNameTextField.accessibilityLabel = firstNameLabel.text`**



The screenshot shows the Accessibility inspector in Xcode. The title is "Accessibility". Under "Accessibility", there is a checked checkbox for "Enabled". Below this are three text input fields: "Label" with the text "Delete Item", "Hint" with the text "Hint", and "Identifier" with the text "Identifier". Under the "Traits" section, there are several checkboxes: "Button" (checked), "Image", "Link", "Selected", "Static Text", "Search Field", "Plays Sound", "Keyboard Key", "Summary Element", "User Interaction Enabled" (checked), "Updates Frequently", "Starts Media Session", "Adjustable", "Allows Direct Interaction", "Causes Page Turn", and "Header".

Accessibility Inspector



Accessibility Inspector

▼ Hierarchy

- ▼ AXApplication
 - ▼ AXWindow:AXStandardWindow
 - ▼ AXScrollArea
 - AXCheckBox

▼ Attributes

isAccessibilityEnabled	YES
accessibilityParent	<AXScrollArea>
isAccessibilityFocused	NO
accessibilityRole	AXCheckBox
accessibilityTopLevelUIElement	<AXWindow:AXStandardWindow>
accessibilityTitle	<nil>
accessibilityHelp	<nil>
accessibilityValue	1
accessibilityRoleDescription	checkbox
accessibilityWindow	<AXWindow:AXStandardWindow>
accessibilityFrame	x=1059.00 y=1648.00 w=93.00 h=153.00

▼ Actions

- accessibilityPerformPress

▼ Warnings And Errors

- ⚠ Missing value for accessibilityTitle, accessibilityLabel, or accessibilityTitleUIElement

Common Native iOS Accessibility Problems & Solutions

Speak Notifications & Send VoiceOver Focus

```
/*  
UIAccessibilityPostNotification  
  
This function posts a notification to assistive applications.  
Some notifications specify a required or optional argument.  
Pass nil for the argument if the notification does not specify otherwise.  
See UIAccessibilityConstants.h for a list of notifications.  
*/  
UIKIT_EXTERN void UIAccessibilityPostNotification(UIAccessibilityNotifications notification, id argument);
```

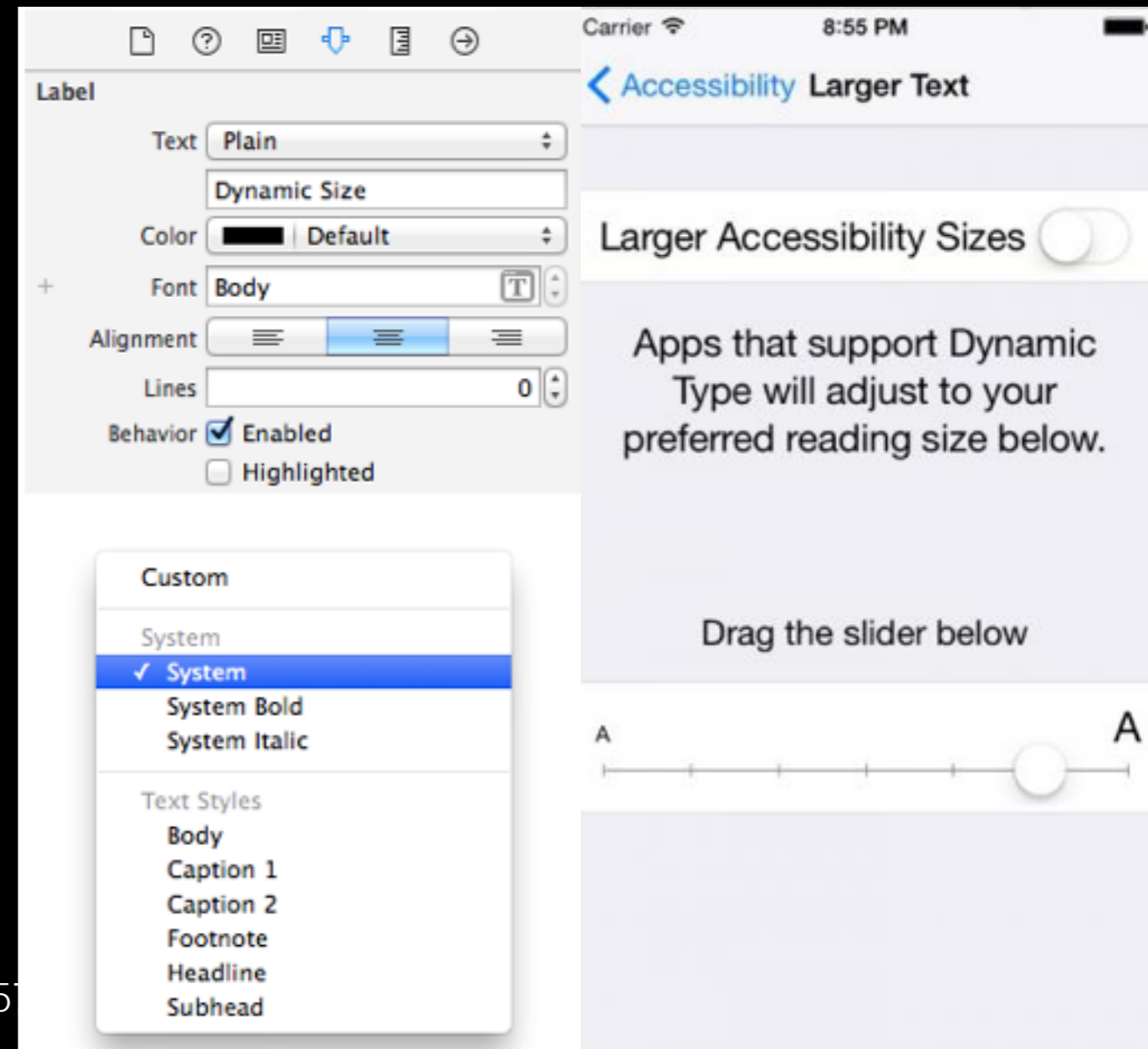
Easy Modal Dialog!

```
/*  
Informs whether the receiving view should be considered modal by accessibility. If YES, then  
elements outside this view will be ignored. Only elements inside this view will be exposed.  
default == NO  
*/  
@property(nonatomic) BOOL accessibilityViewsModal NS_AVAILABLE_IOS(5_0);
```

<http://pauljadam.com/iosvsandroida11y/#native>

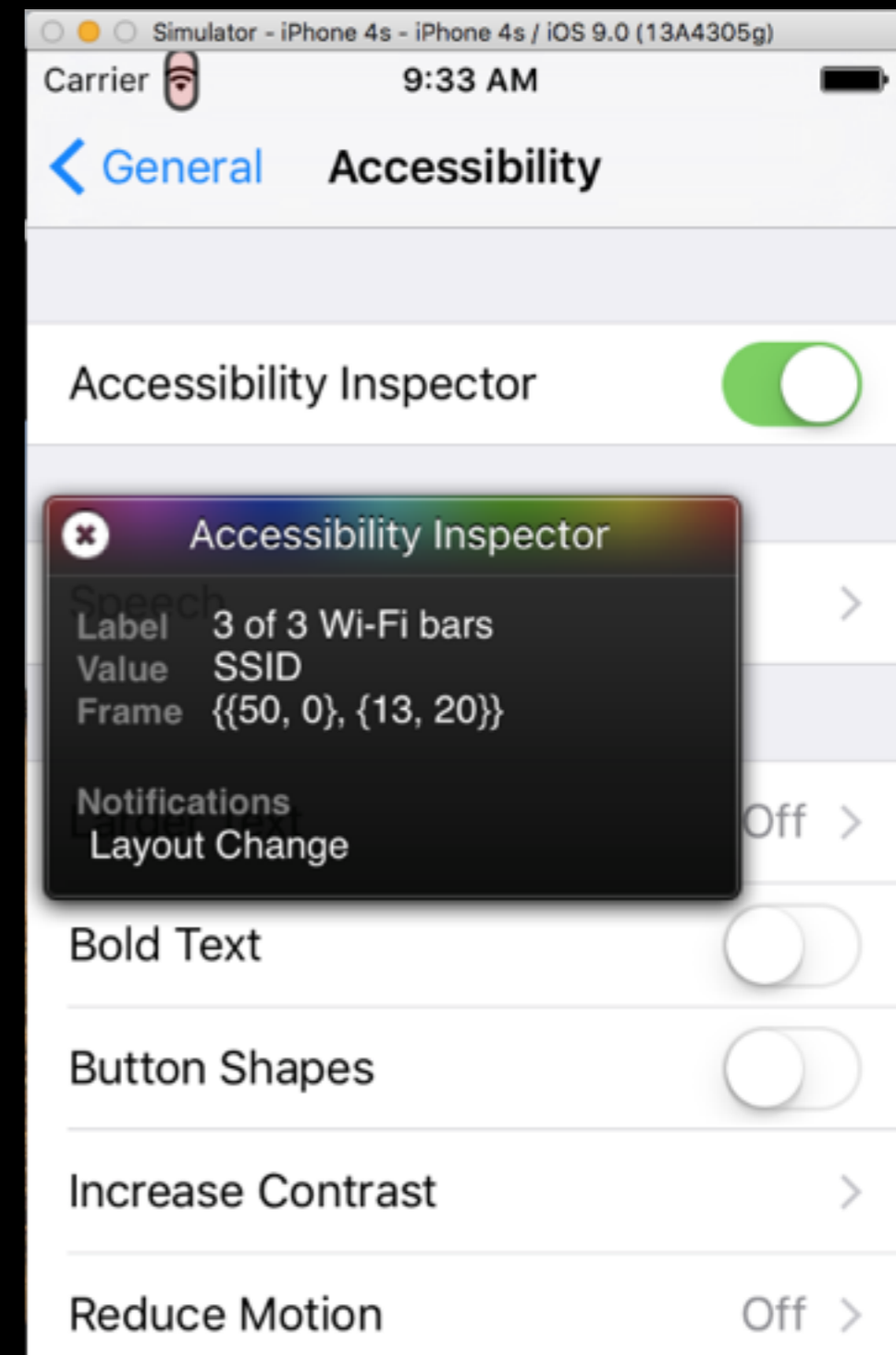
Dynamic Type

- <http://www.electricpeelsoftware.com/2014/09/04/dynamic-type.html>
- <http://www.iphonelife.com/blog/31369/swift-programming-101-mastering-dynamic-type-ios>



Simulator & Device Testing

- iOS Simulator Testing
- VoiceOver On-Device Testing



Apple Watch Accessibility

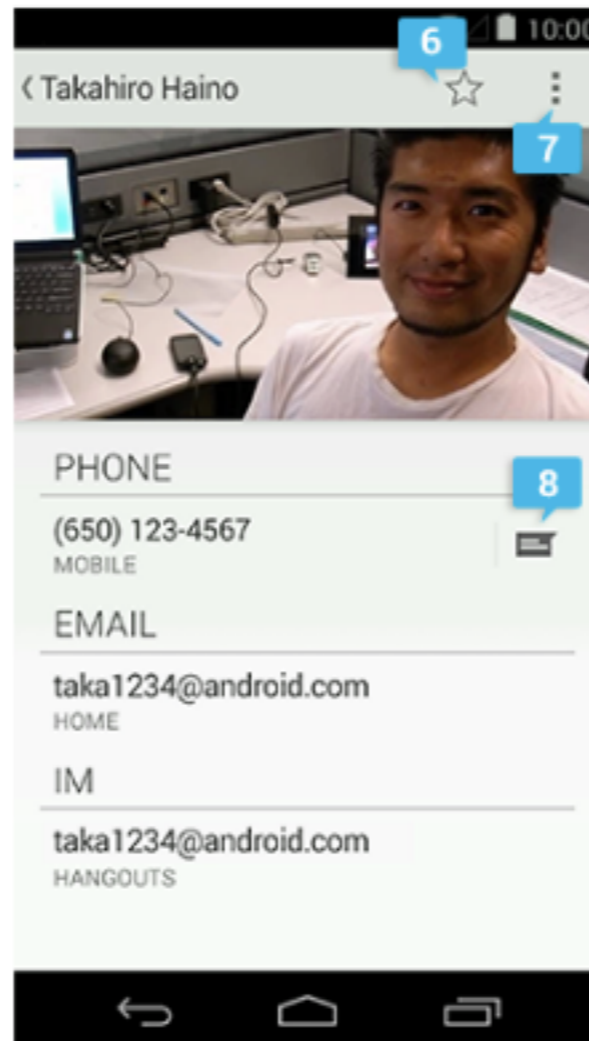
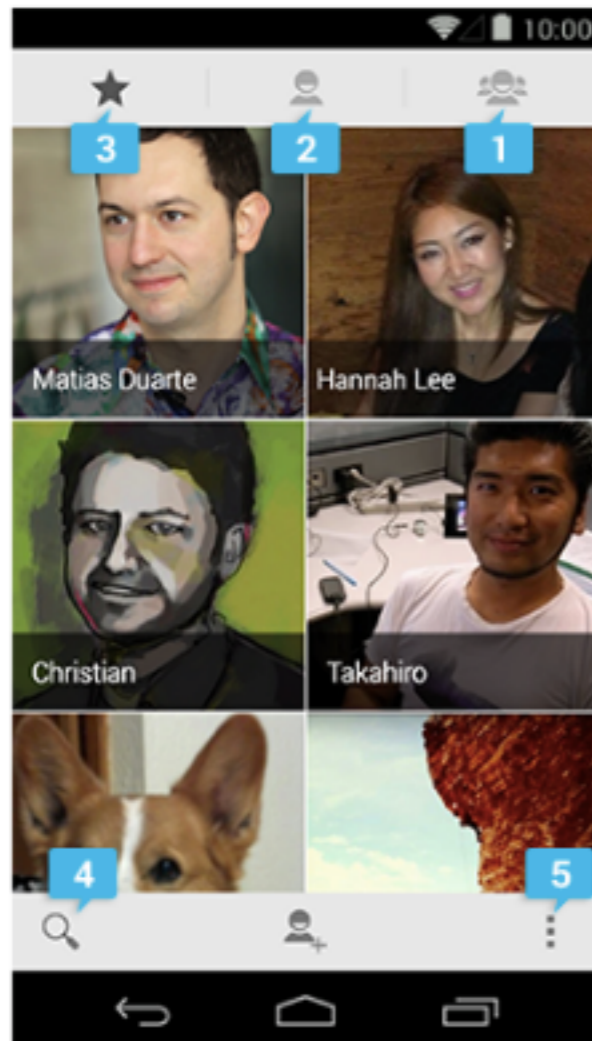
- <https://www.apple.com/accessibility/watch/>



Native Android App Accessibility

Label visual UI elements meaningfully

In your wireframes, **label functional UI components** that have no visible text. Those components might be buttons, icons, tabs with icons, and icons with state (like stars). Developers can use the `contentDescription` attribute to set the label.



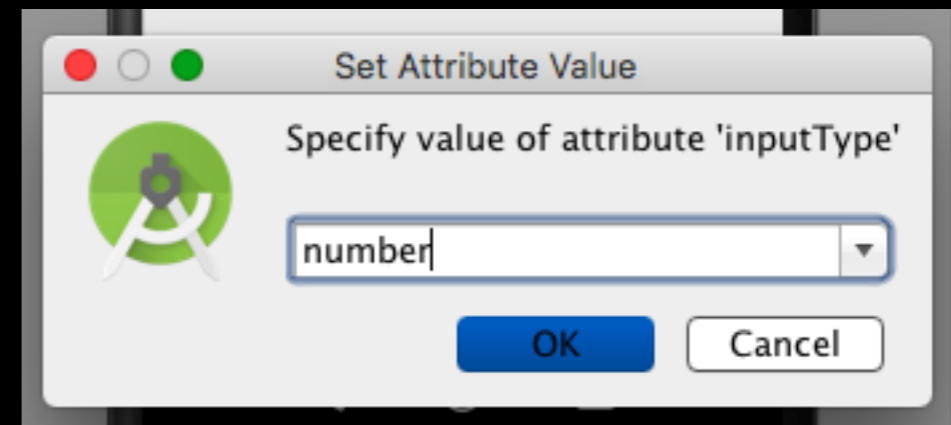
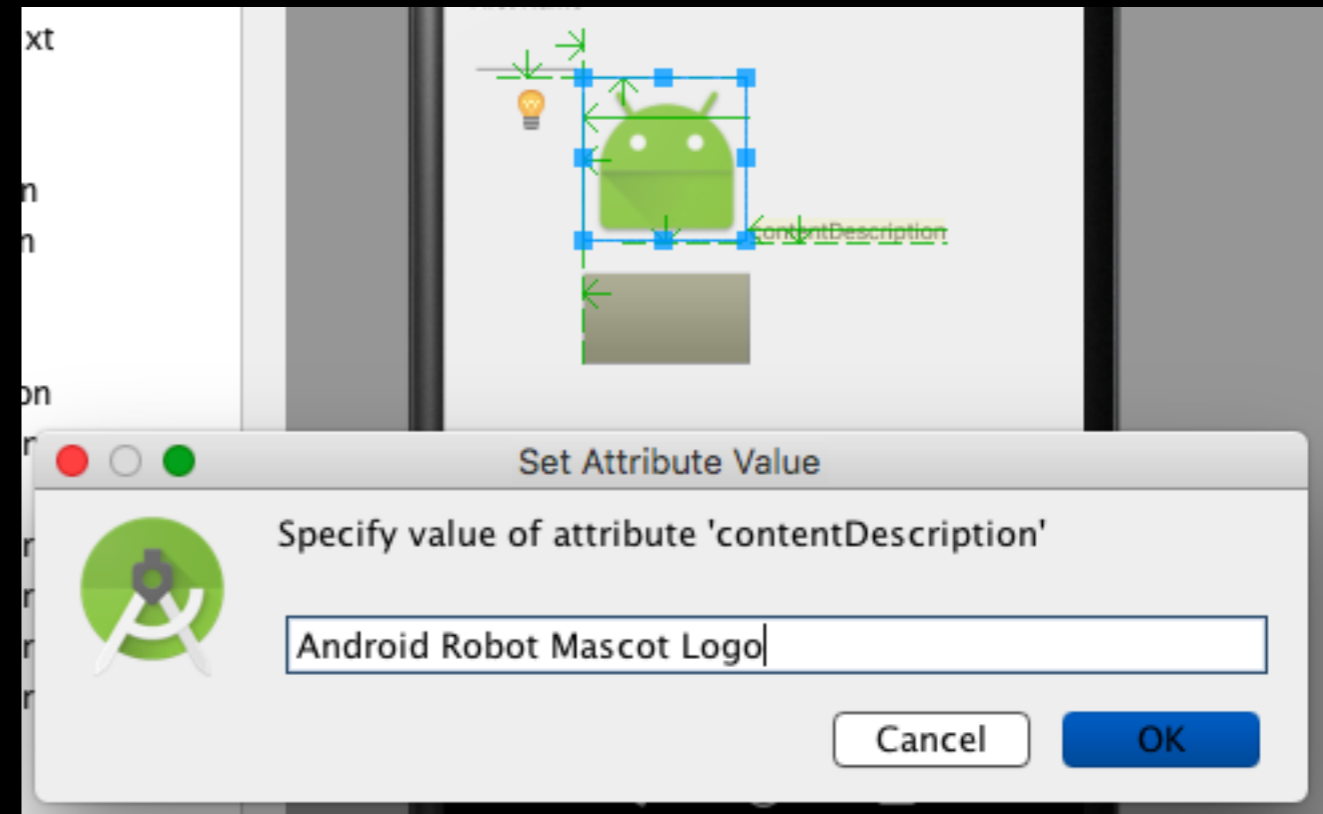
1. group
2. all contacts
3. favorites
4. search
5. action overflow button
6. *when starred: remove from favorites*
when not starred: add to favorites
7. action overflow button
8. text message

Android Accessibility API

- <http://developer.android.com/guide/topics/ui/accessibility/index.html>
- **Common Android Accessibility Problems & Solutions**
- **String contentDescription = "Select " + strValues[position];
label.setContentDescription(contentDescription);**
- **Android Accessibility API Capabilities/Limitations/Version Numbers**

Android Studio

- **Fixing Accessibility Issues in Design View, XML View, Java Code View**
- **Android Lint Accessibility Tooltip Errors**
- **Add missing accessibility properties via error dialogs.**



Android Lint Accessibility Errors & Solutions

- `<TextView>` `android:labelFor` Property

- `<TextView`

- `android:id="@+id/tv_caption"`

- `android:labelFor="@+id/edit_rate"`

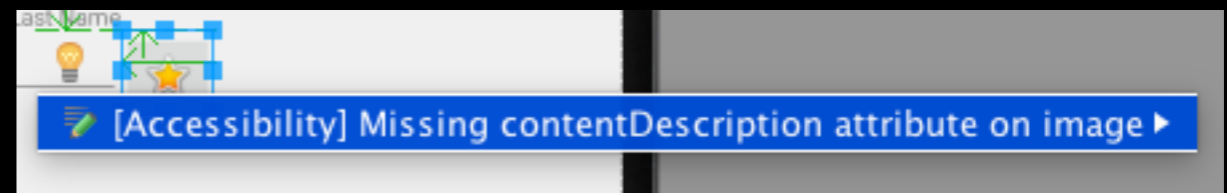
- `android:text="@string/txt_rate">`

- `</TextView >`

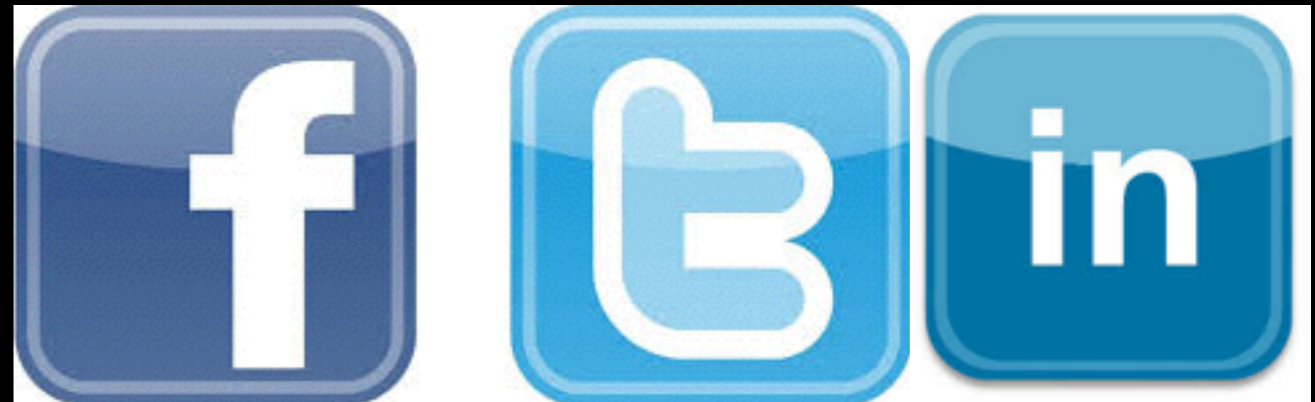
- `<EditText`

- `android:id="@+id/edit_rate">`

- `</EditText>`



Hands-on Native iOS and Android Accessibility Testing



- **Hands-On Native iOS or Android Accessibility Testing of Popular Apps Activity**



Mobile Frameworks



- **Open Source / Progressive Enhancement**

- **Proprietary / No Semantics**

Hands-on Live testing of Participant-suggested Mobile Websites and Apps

- **Presenter will test mobile sites & apps in class as suggested by attendees.**
- **If no suggestions test popular mobile websites for a11y issues.**

Questions?

- **Connecting with Deque**

- **Twitter**

- **@deque**[systems](#)



- **LinkedIn**

- **Deque** [Systems](#)



- **Web**

- **www.deque.com**



- **Email**

- **info@deque.com**

