

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

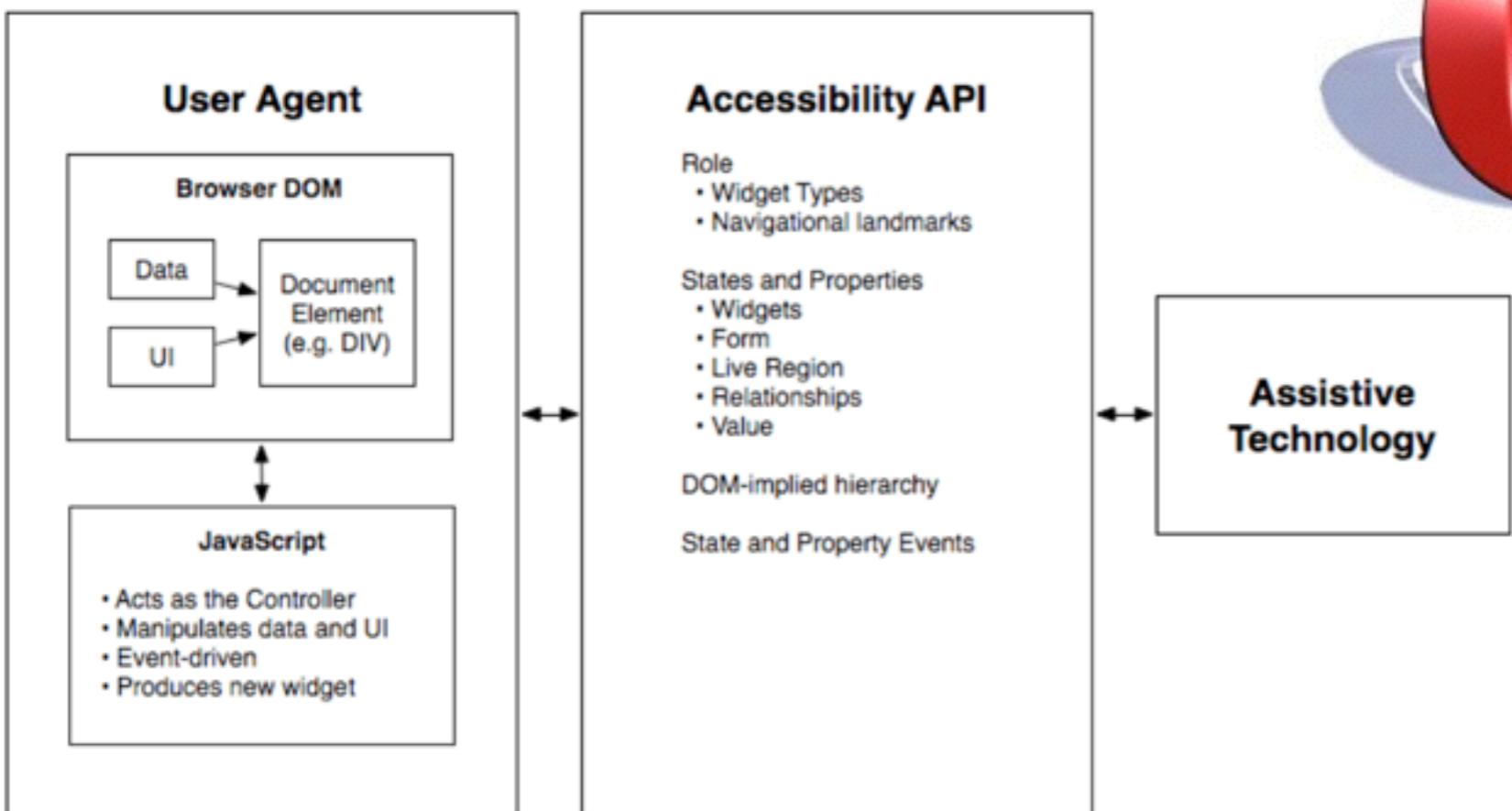
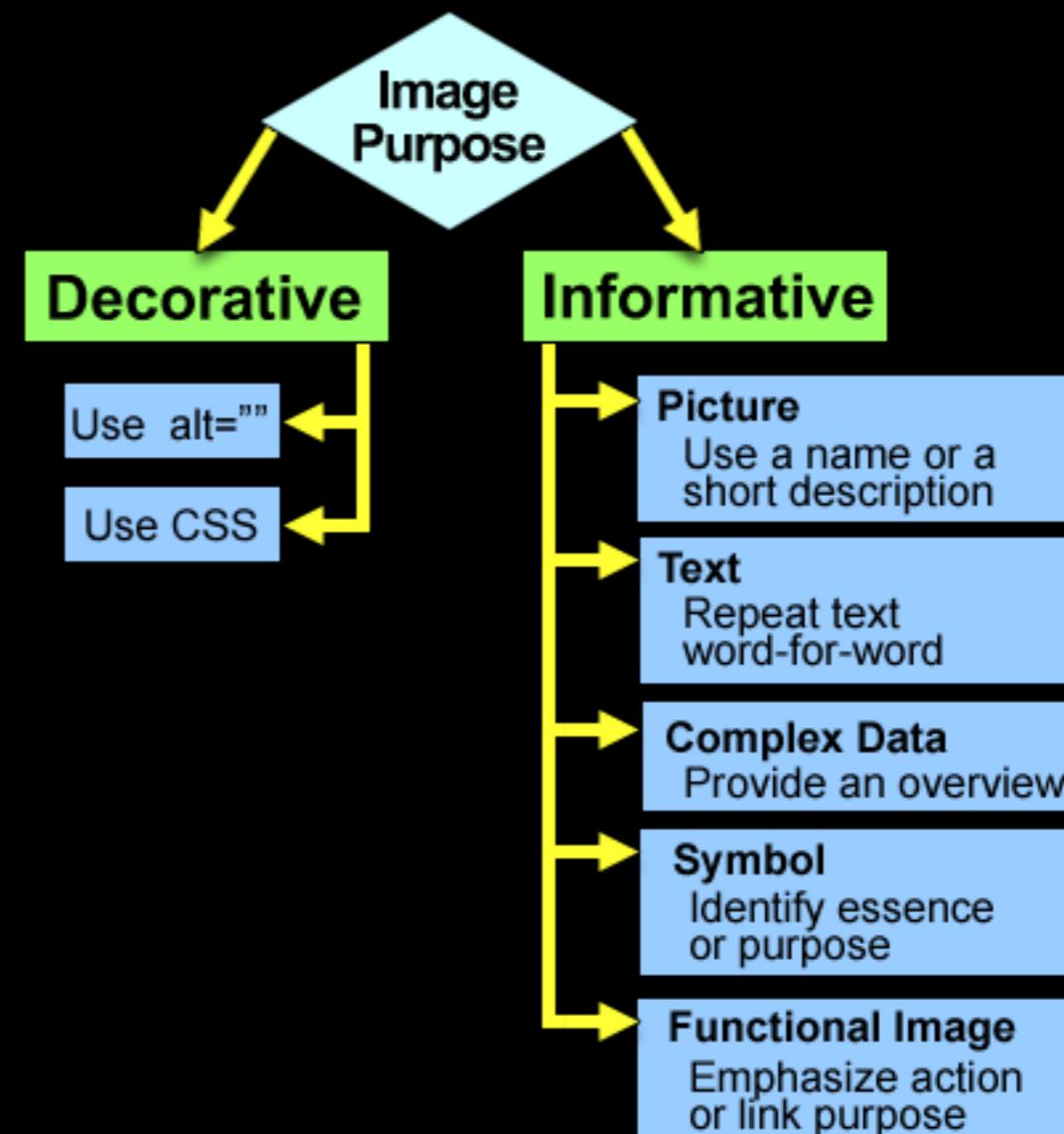


Figure 1: The contract model with accessibility APIs

Accessible Name & ALT Text

- Active (links/buttons), Decorative, Informational, CSS Icons, Background Images, etc
- <http://pauljadam.com/demos/accessiblelenameimg.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/ecllasses/104_images/1.3_lecture_short_how.html



Headings

Deque Blog



EST PRACTICE

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 a property that defines what an element does. It is important to set up traits correctly so users are not confused when clicking on [...]

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



EST PRACTICE

Accessible Text Input in Android

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

[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, Train...ibility and Section 508 Compliance
- 2: "It's Easy to be Smart After the Fact." Deque University
- 3: That defense doesn't work here....nowledge 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 F...Web Accessibility Testing Product
 - 2: Accessibility Stories
 - 2: Deque Highlighted Partnership
 - 3: We are the official jQuery acces...and jquery partners in accessibility
 - ↳ 2: Essential Guide to Digital Accessibility Infographic

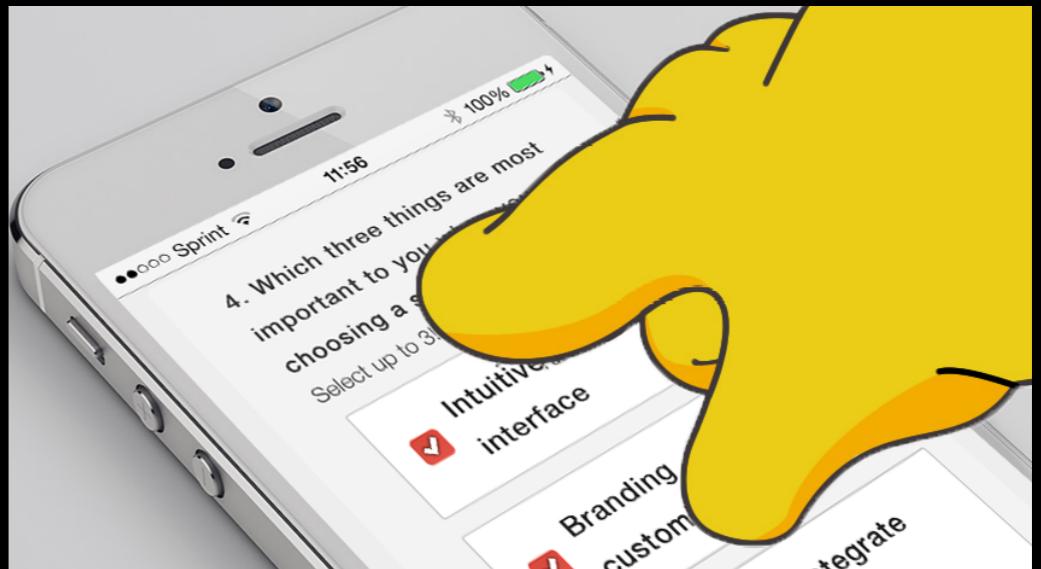
Data Tables

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

2015 Sales Report

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**

Build your trip

Flight Hotel Car Cruise Activities

Book together & save!

Flight + Hotel
 Flight + Car
 Flight + Hotel + Car
 Hotel + Car

Trip information:

Roundtrip One way Multiple destinations

My dates are flexible (popular US routes only)

Leaving from: San Jose, CA (SJC-San Jo) Going to: _____

Departing: _____ Time: _____ Returning: _____ Time: _____

Any Any

Adults (19-64): Seniors (65+): Children (0-18):

Additional options:
[Airline, first or business class, nonstop only](#)

Search for flights

Build your trip

Flight Hotel Car Cruise Activities

Book together & save!

Flight + Hotel
 Flight + Car
 Flight + Hotel + Car
 Hotel + Car

Trip information:

Roundtrip One way Multiple destinations

My dates are flexible (popular US routes only)

Leaving from: San Jose, CA (SJC-San Jo) Going to: _____

Departing: _____ Time: _____ Returning: _____ Time: _____

Any Any

Adults (19-64): Seniors (65+): Children (0-18):

Additional options:
[Airline, first or business class, nonstop only](#)

Search for flights

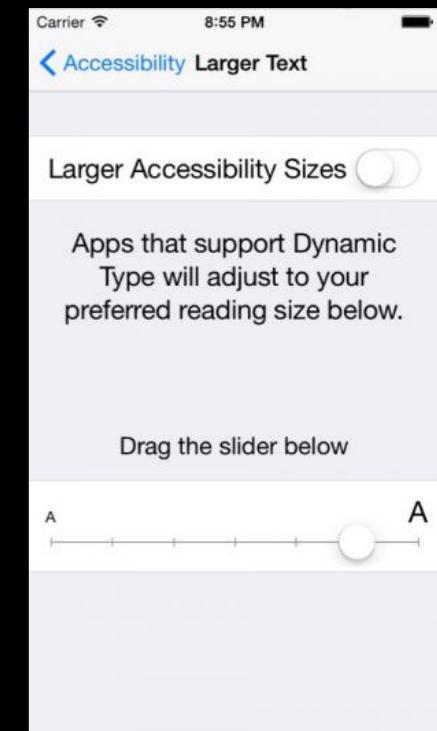
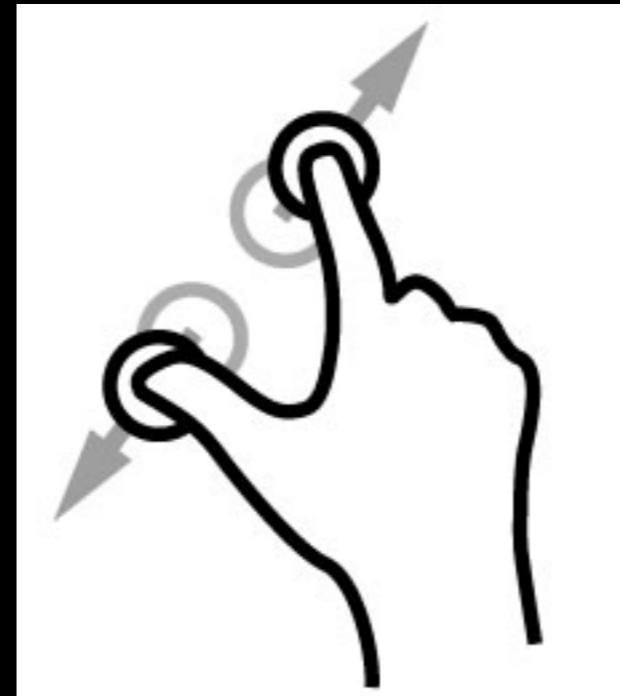
Keyboard Accessibility

- **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>**

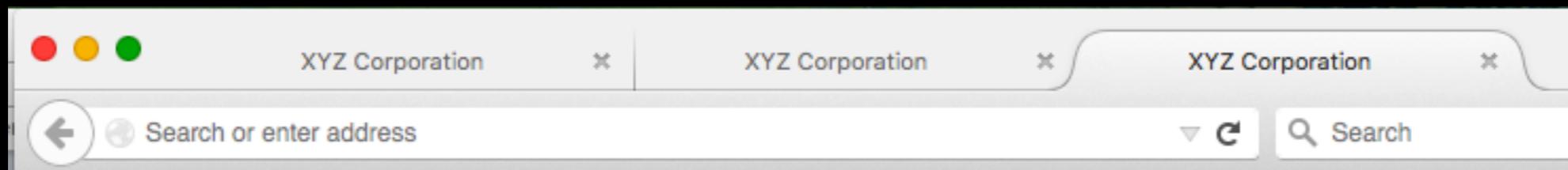


Text Size & Readability

- **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>



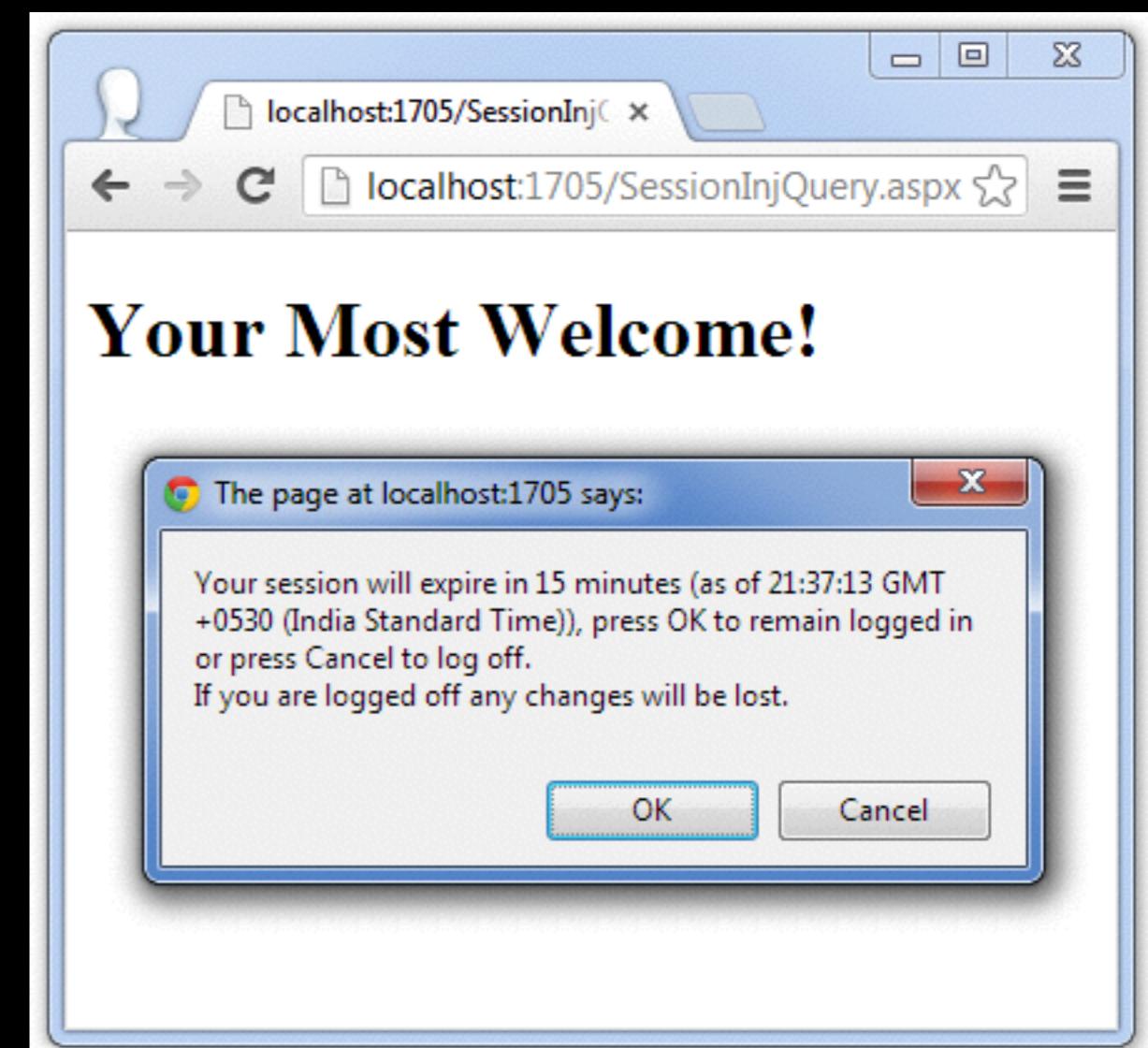
Page Titles



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

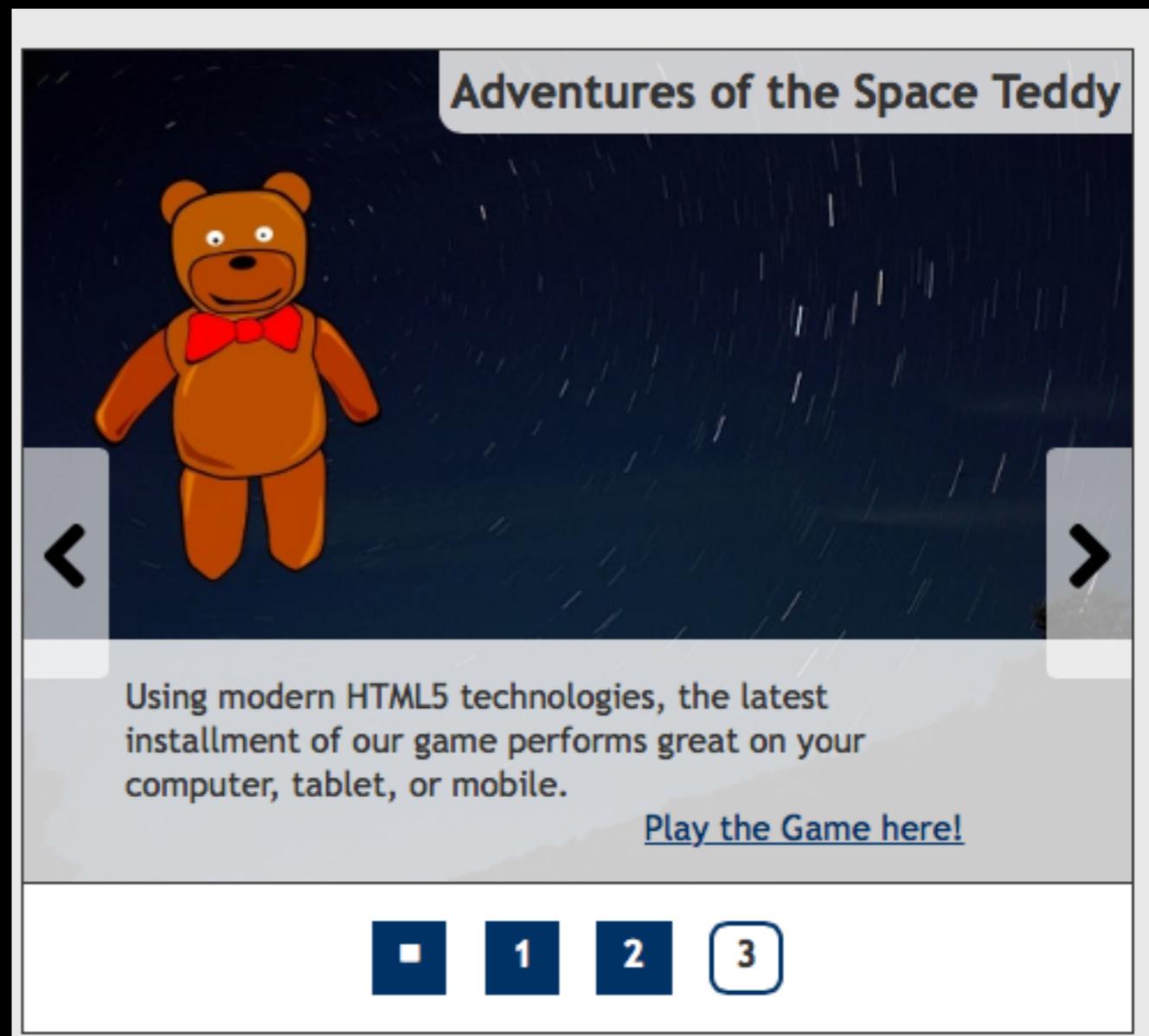
Session Timeouts

- **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>



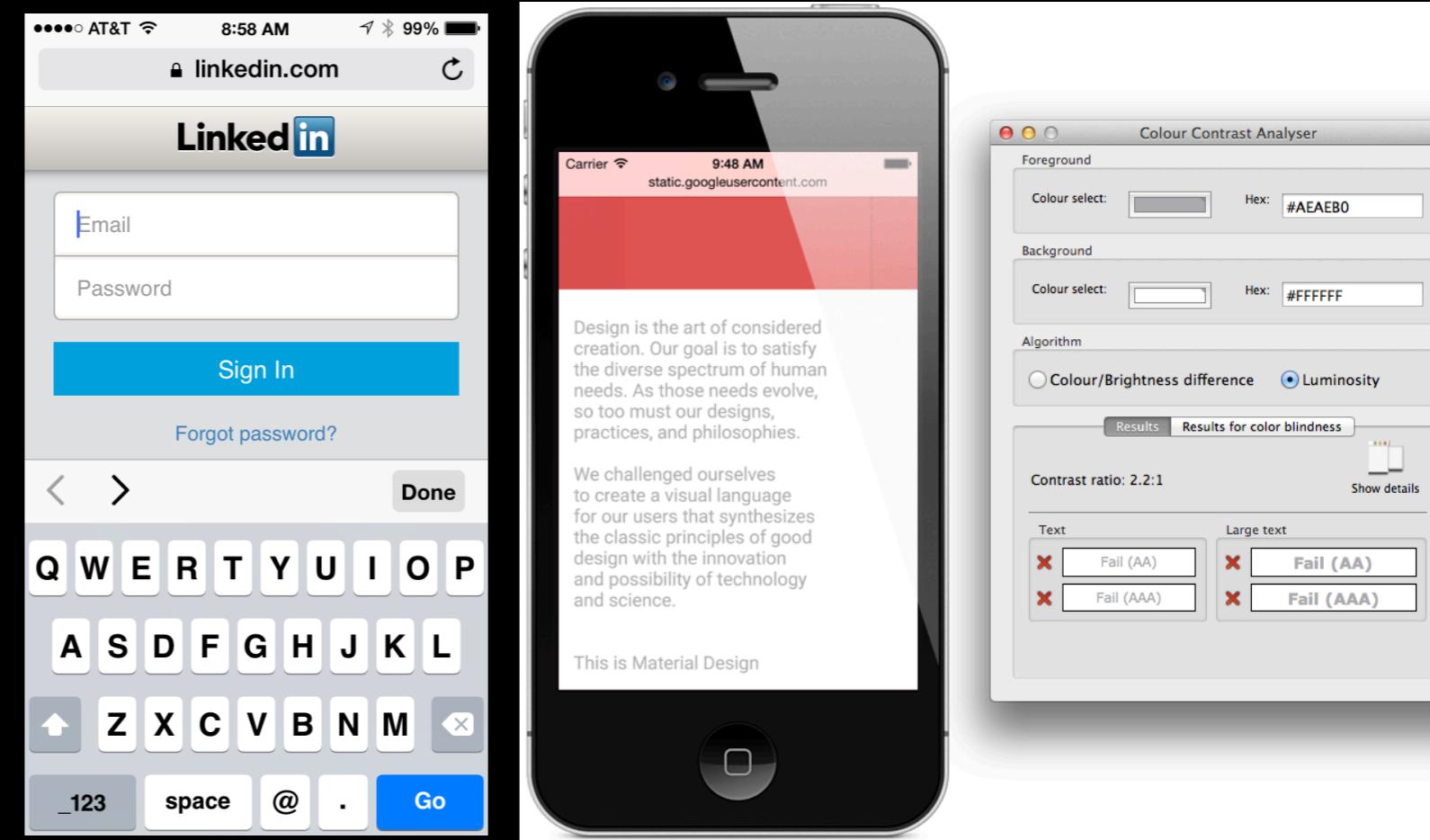
Focus Outline & Visibility

- 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>



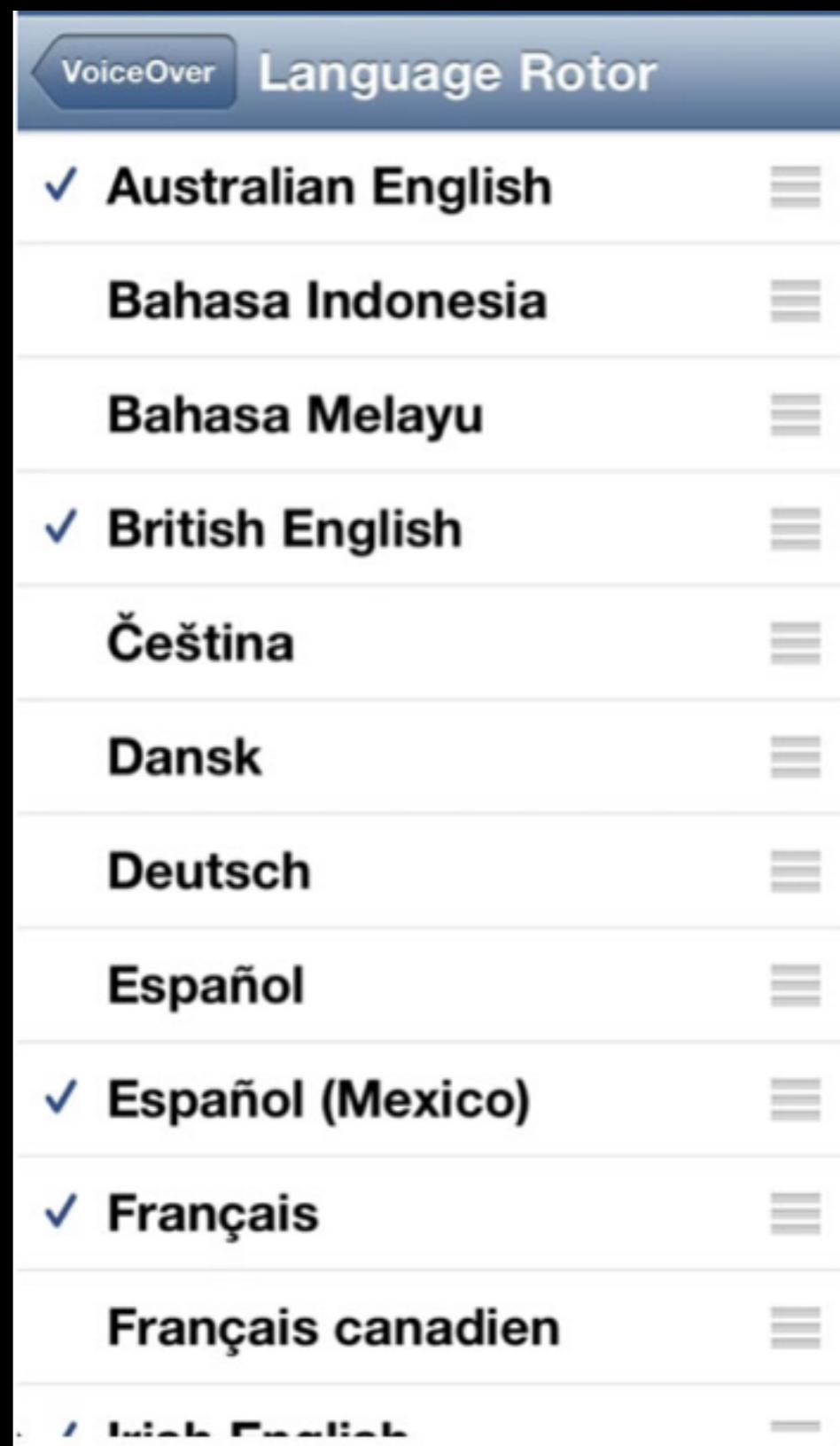
Use of Color & Contrast

- 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!



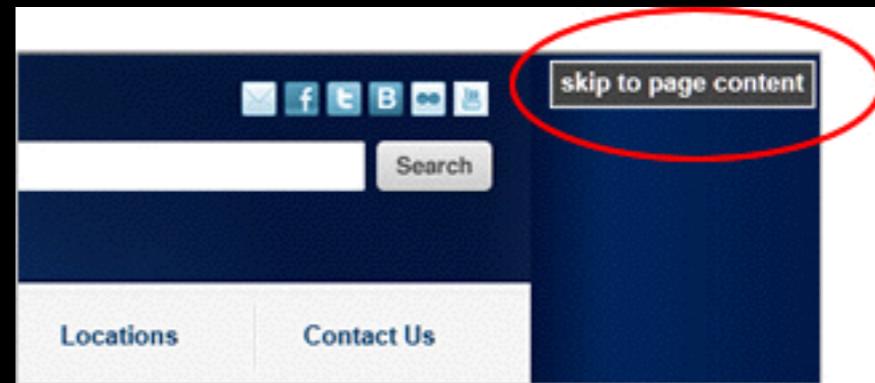
Language & i18n

- 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

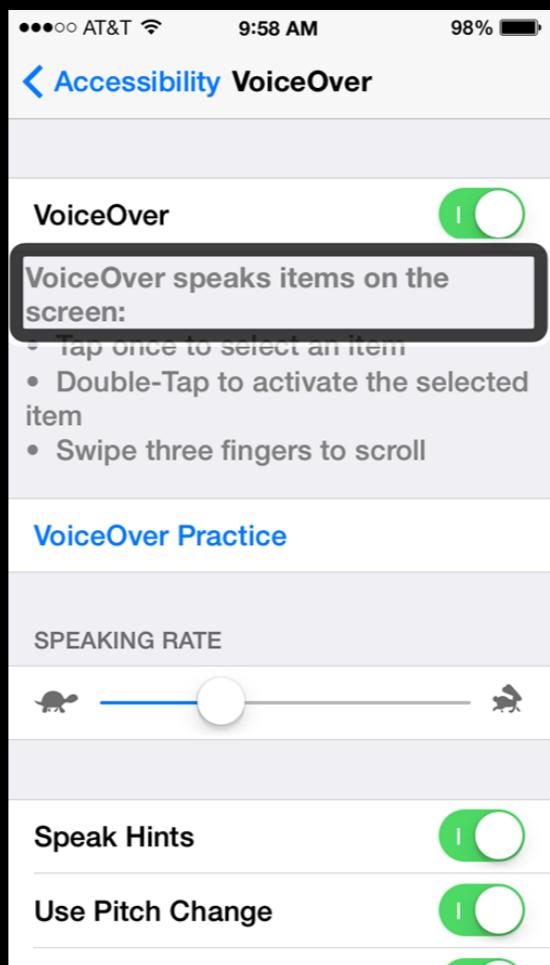


- **VoiceOver iOS Cheatsheet**
- **TalkBack Android Cheatsheet**
- **Mobile Accessibility QA Testing Checklist**



VoiceOver for iOS

- Gestures
- Rotor



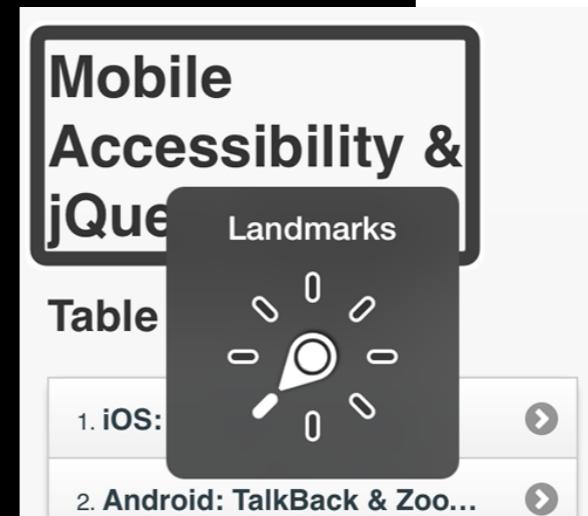
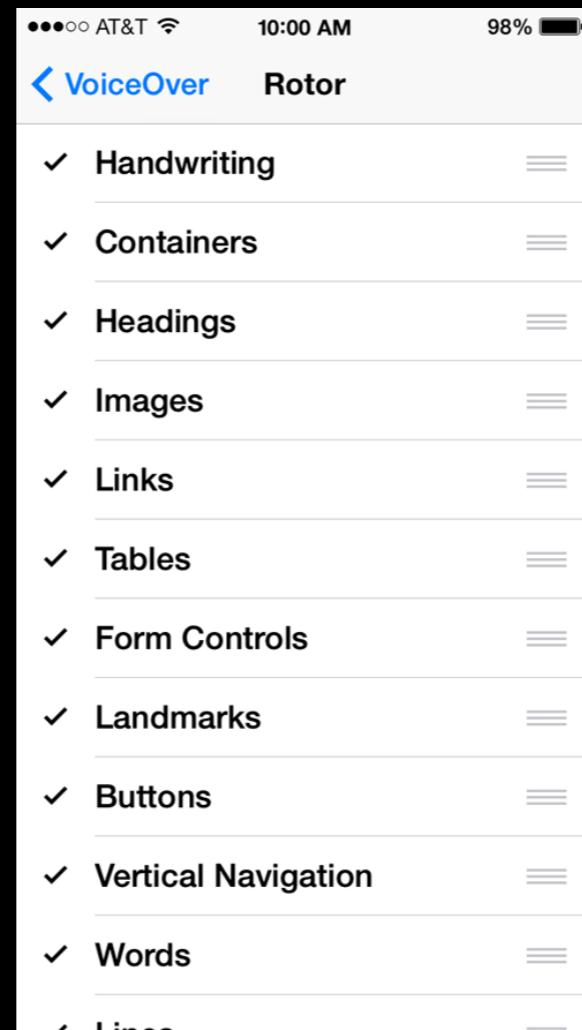
VoiceOver Keyboard Testing

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

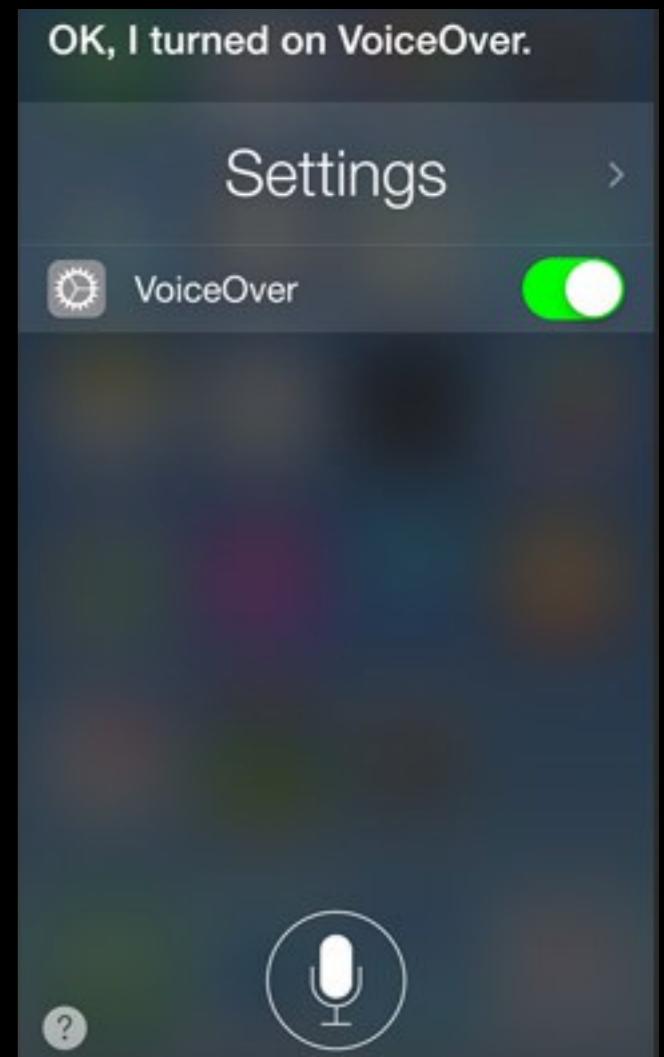
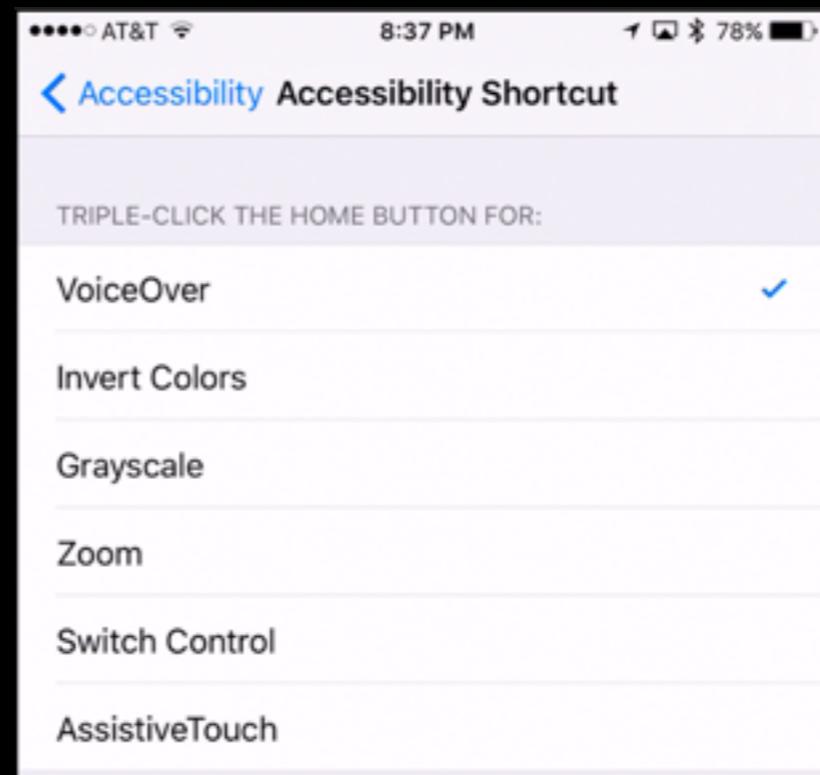


Data Table Testing on iOS

- VoiceOver Rotor
- Rows
- Vertical Navigation



Accessibility Shortcut (Triple-Click Home Button for VoiceOver)



- Siri voice commands can also enable/disable VoiceOver.

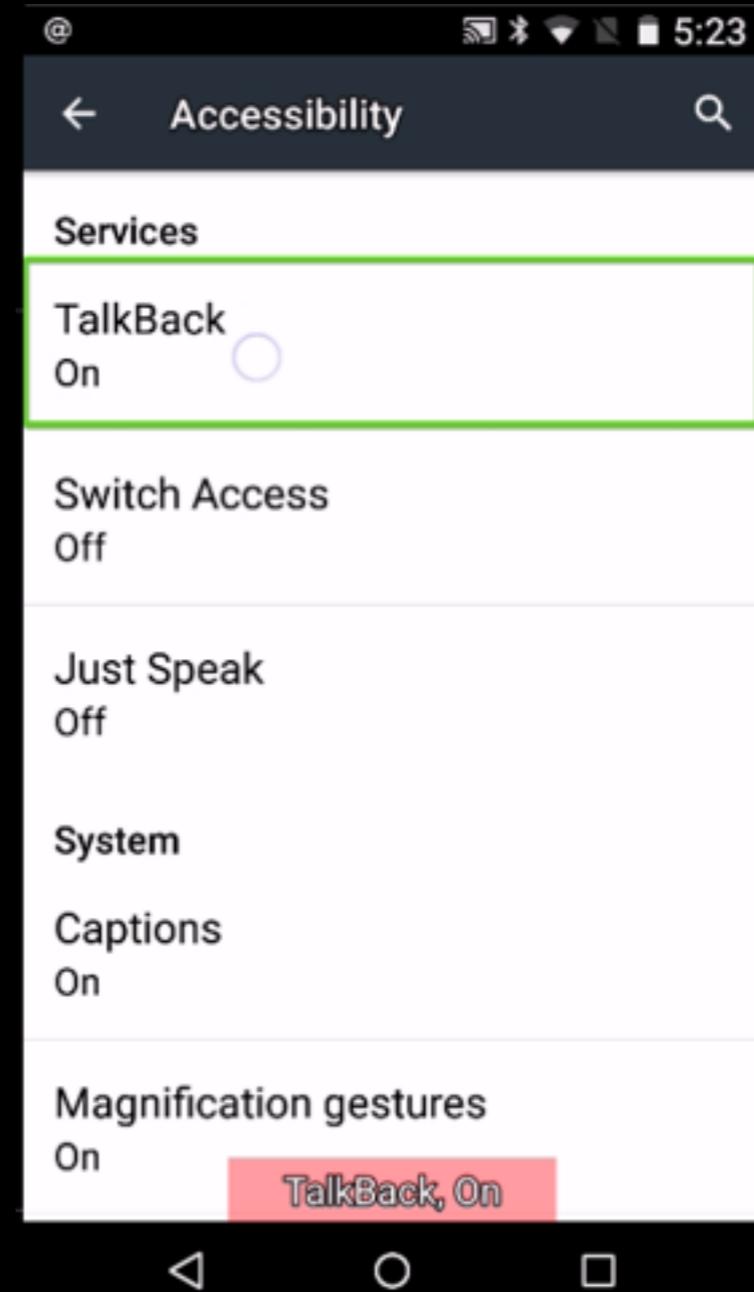
Virtual Keyboard Settings

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

TalkBack for Android

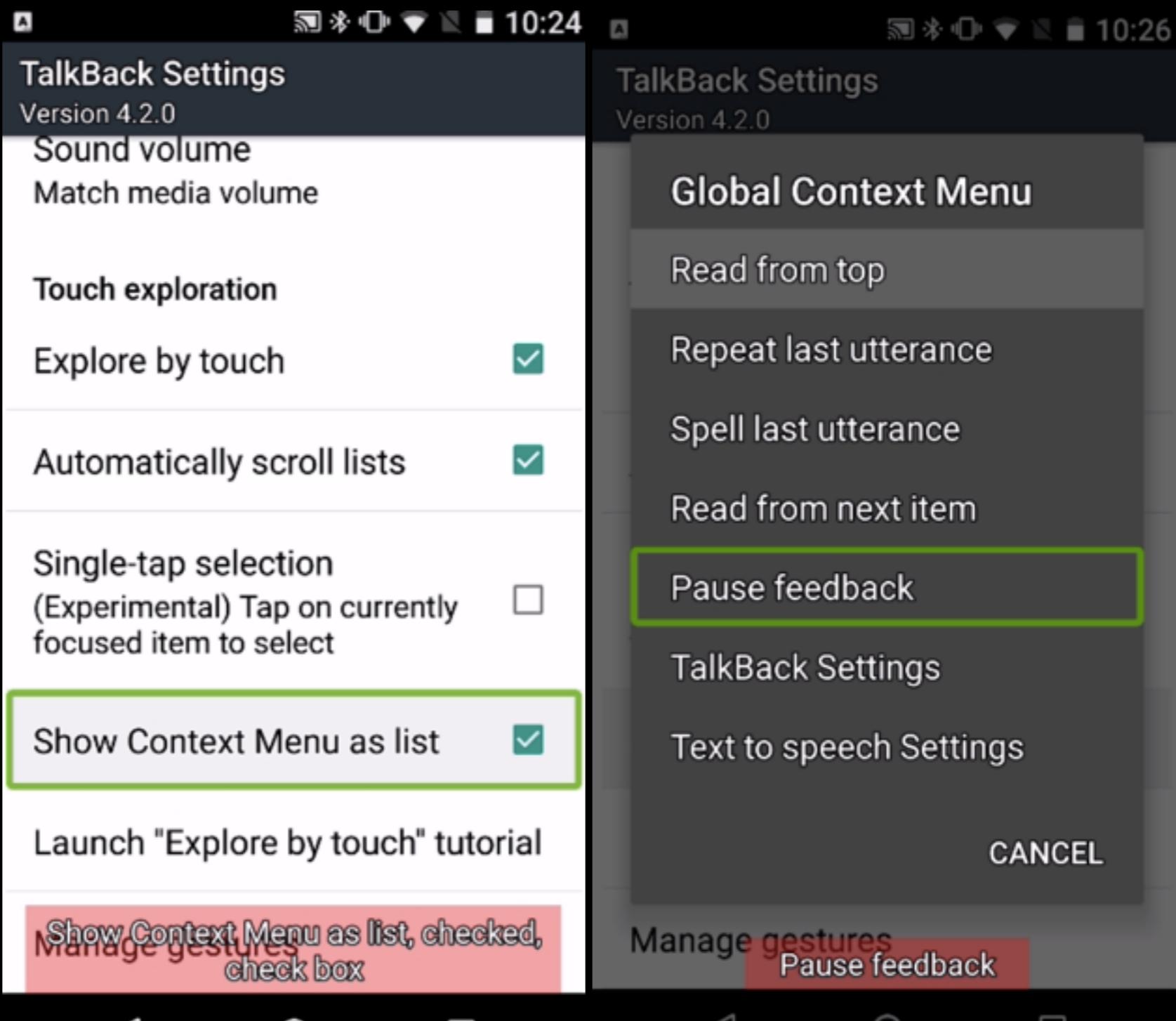


- **Explore by Touch**
- **Gesture Commands**



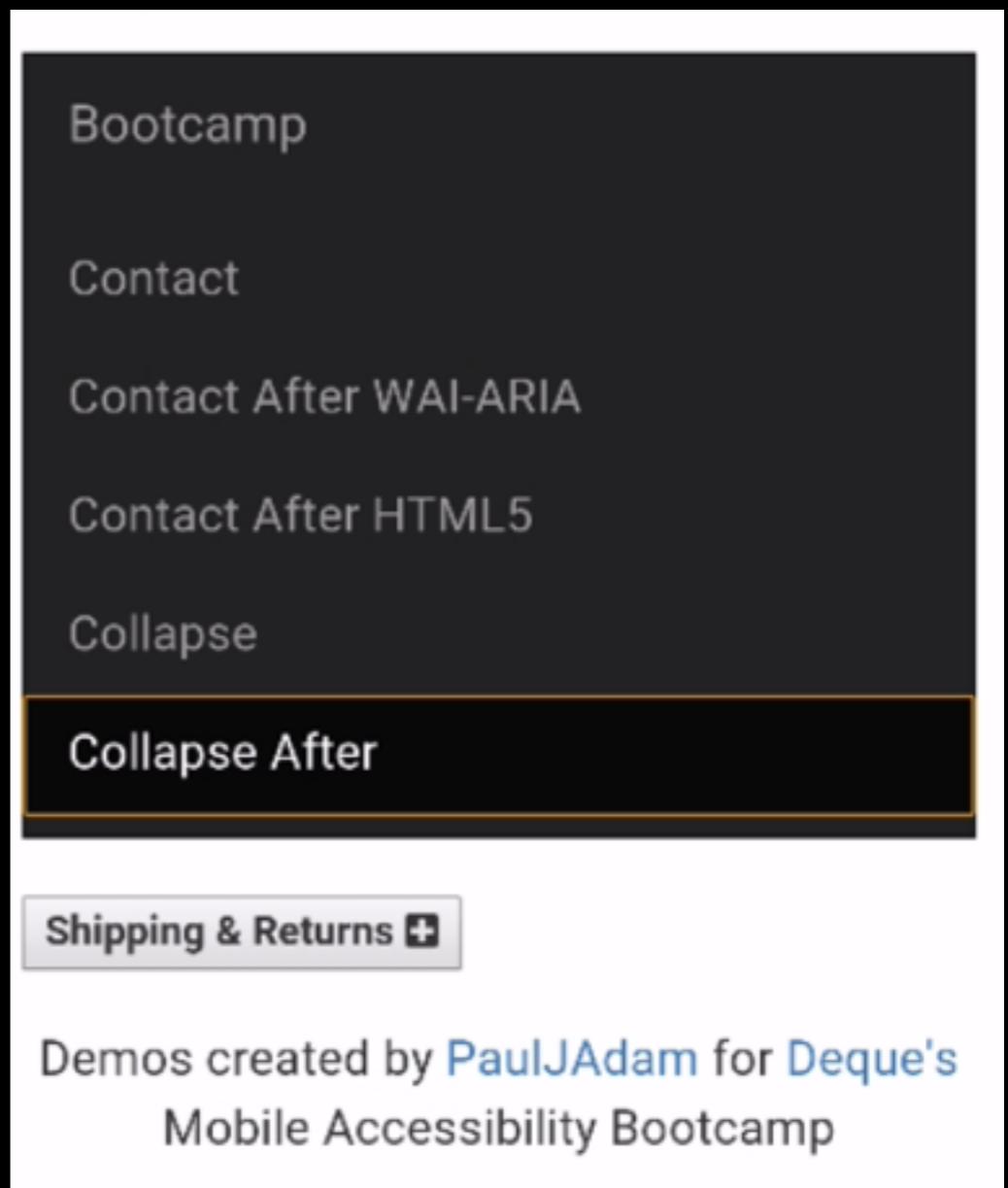
Global & Local Context Menus

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



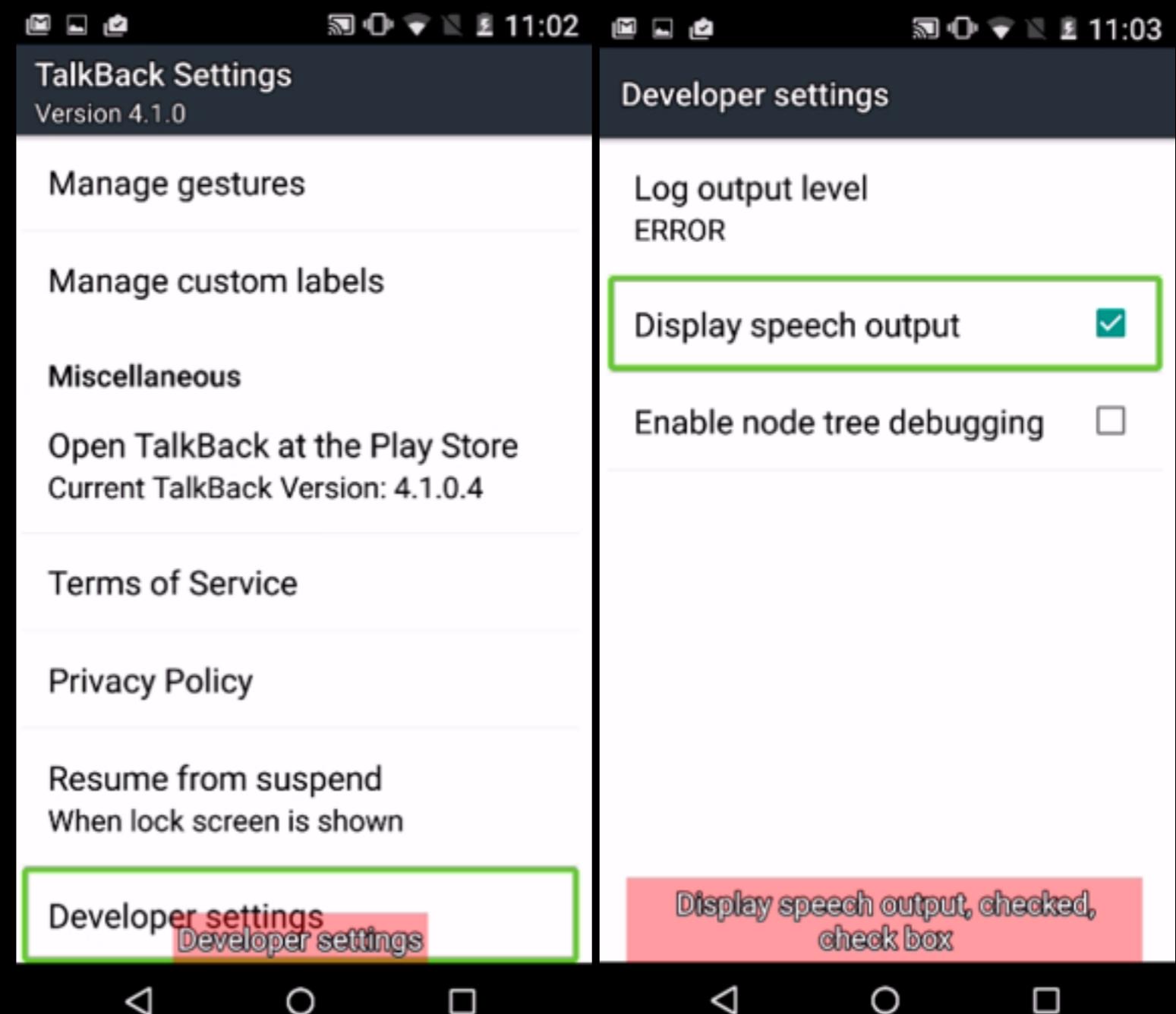
Android Keyboard Accessibility Testing

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



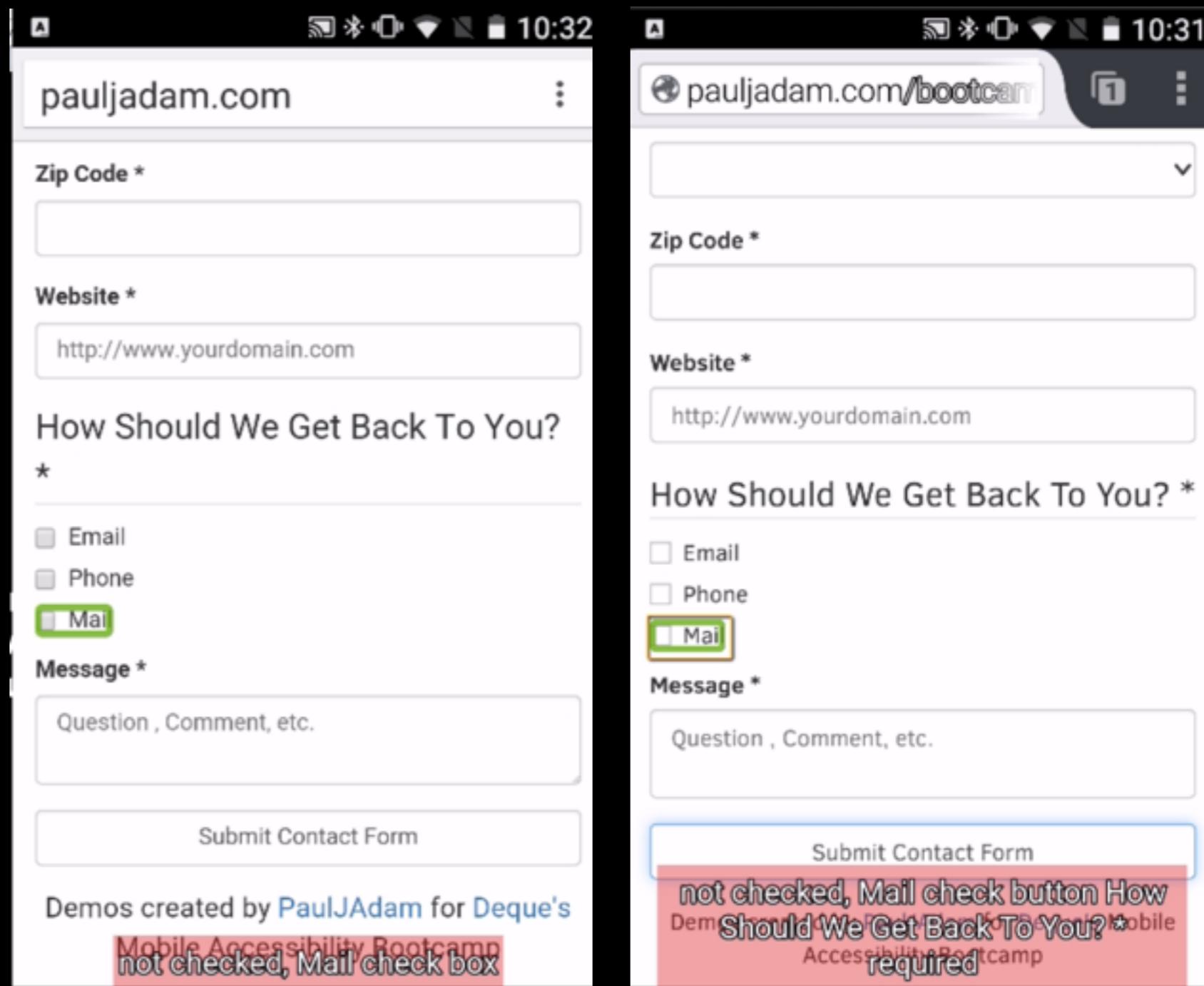
TalkBack Settings

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



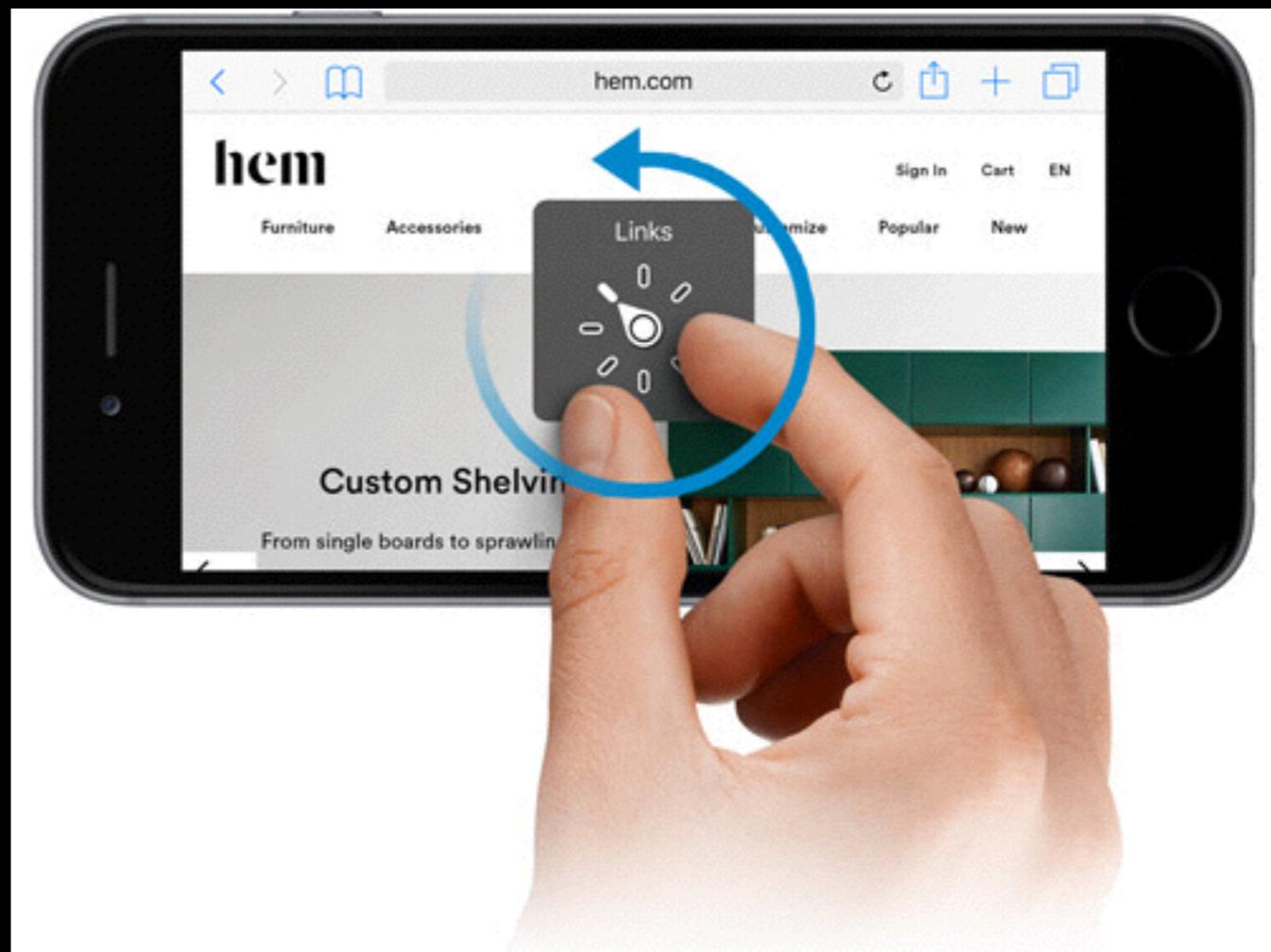
Firefox vs. Chrome

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



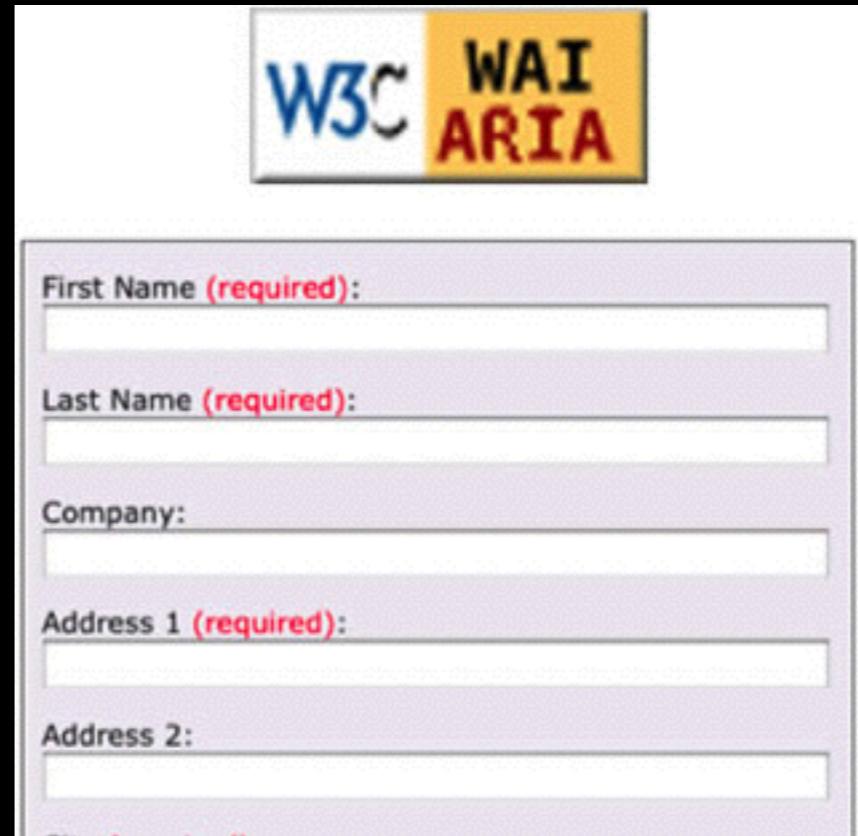
Hands-On Mobile Manual Screen Reader Testing Exercise

- **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.washington.edu/accesscomputing/AU/)
- [http://www.w3.org/WAI/
demos/bad/](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 screenshot shows a web form with a light purple background. At the top, there is a logo consisting of the letters 'W3C' in blue on a white background next to a yellow square containing the letters 'WAI' in black and 'ARIA' in red. Below the logo, the form has five input fields:

- First Name (required):
- Last Name (required):
- Company:
- Address 1 (required):
- Address 2:

Each input field is accompanied by a red label indicating it is required. The entire form is enclosed in a light purple border.

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

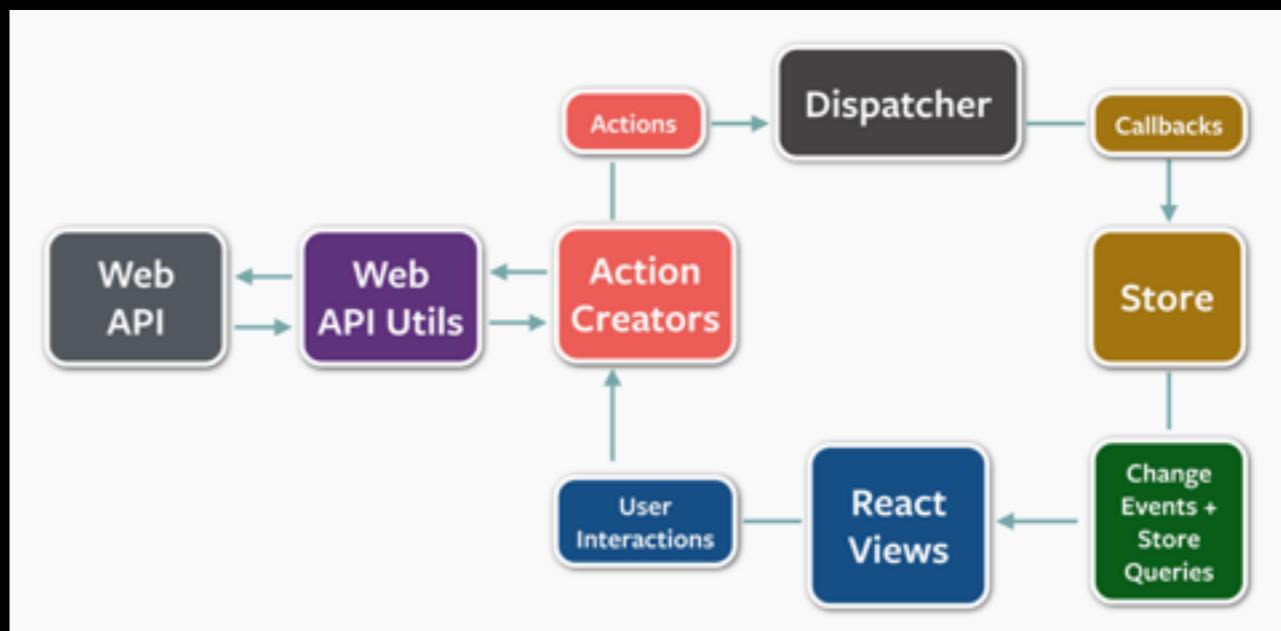
The `HTMLElement.focus()` method sets focus on the specified element, if it can be focused.

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



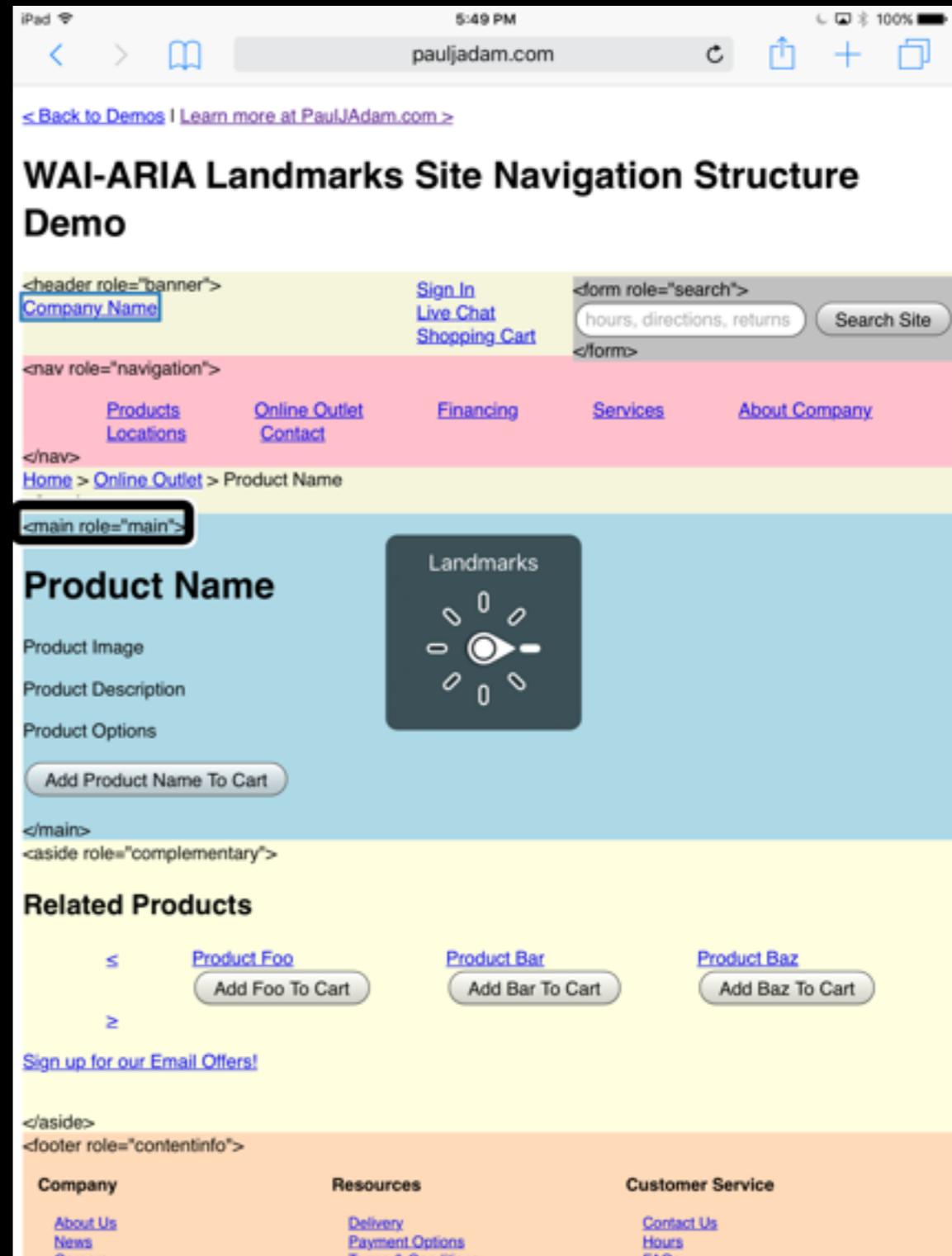
WAI-ARIA Accessibility

- **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">**



Enhancing Forms Accessibility with ARIA

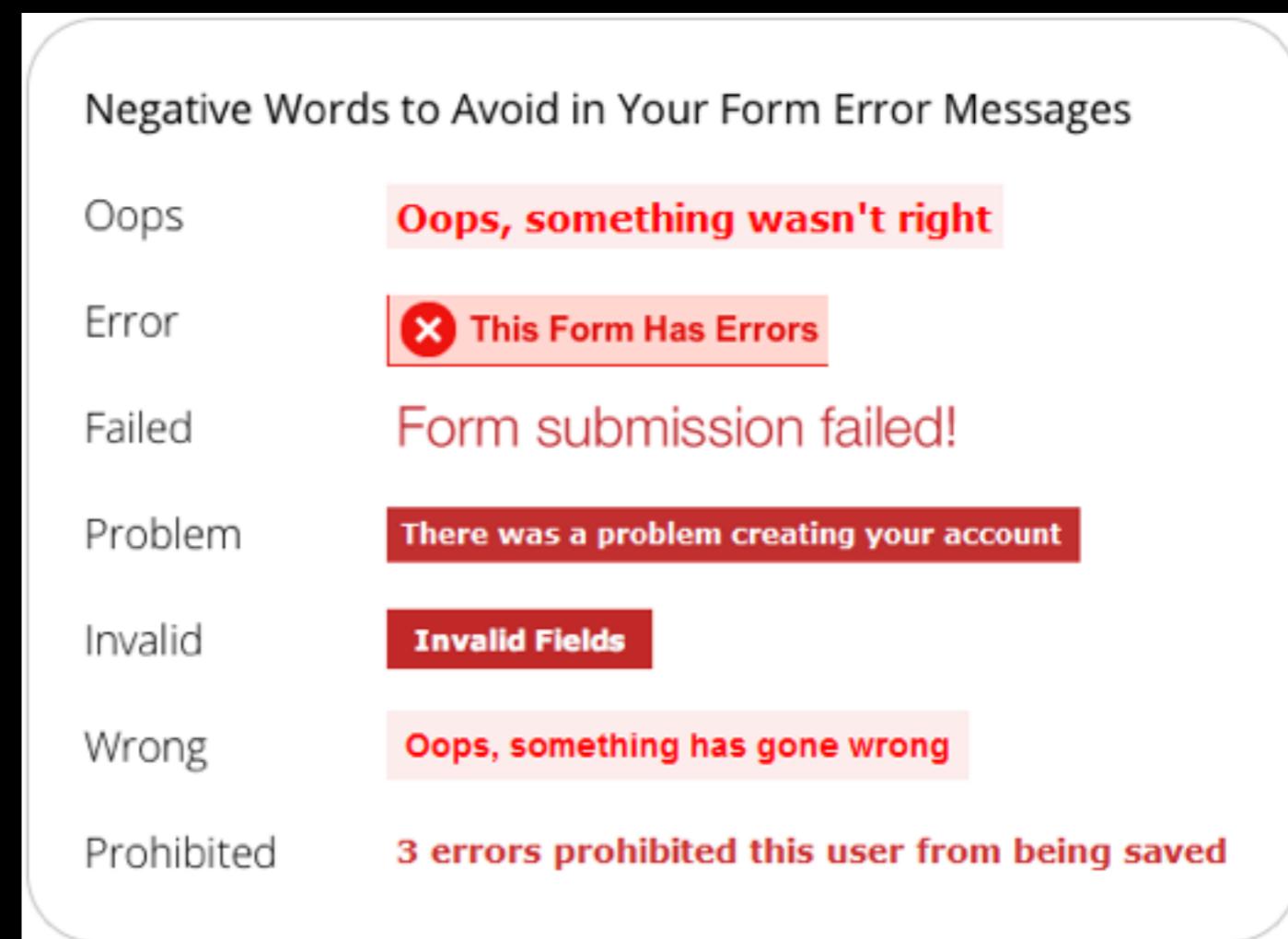
- (**aria-required**, **aria-invalid**, **aria-describedby**, **aria-labelledby**)
- **HTML5 Input Type Validation Works!**
- <http://pauljadam.com/demos/mobileforma11y.html>
- **JavaScript DOM Manipulation to Inject Error Messages & Send .focus();**
- **Change WAI-ARIA Attributes**
- **Simple Form Validation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

A screenshot of a web form titled "User informations". It contains three input fields: "Desired username (already available)", "First name (optional)", and "Last name". The "First name" field has the value "cedric". Red speech bubbles provide validation feedback: one bubble above the first field says "* This field is required" and "* No special characters allowed"; another bubble to the right of the second field says "* This field is required" and "* Letters only"; a third bubble below the third field says "* Please select an option".

A screenshot of a browser window titled "Move Validation Warnings". The address bar shows the URL "kevbaister.github.com/jquery-validation-examples/move-warnings.htm". A red message box at the top of the page contains the text: "Please correct the following errors and try again:" followed by a bulleted list: "• Please enter your first name.", "• Please enter your last name.", and "• Please enter your age.". Below the message box is a form with three input fields labeled "First Name", "Last Name", and "Age", each with a red border indicating they are invalid. A "Submit" button is located below the input fields.

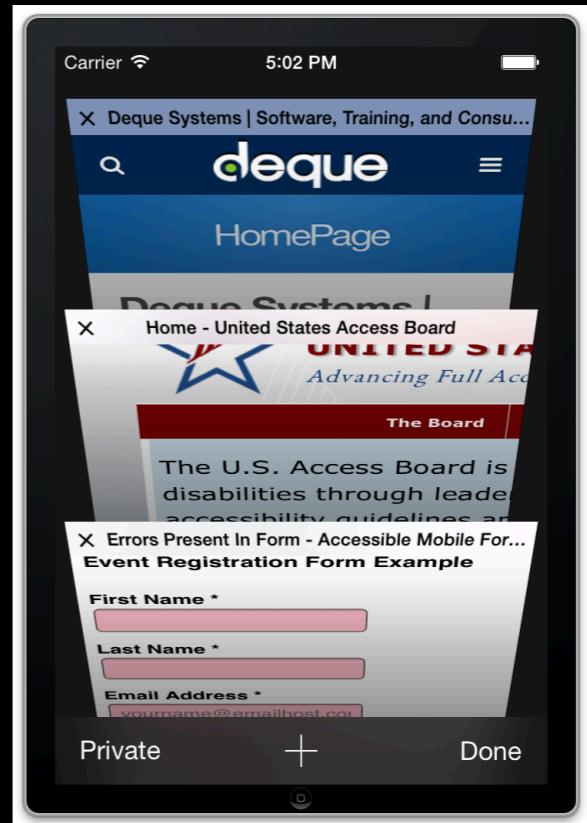
Speaking Messages to Screen Readers with WAI-ARIA Live Regions

- **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>



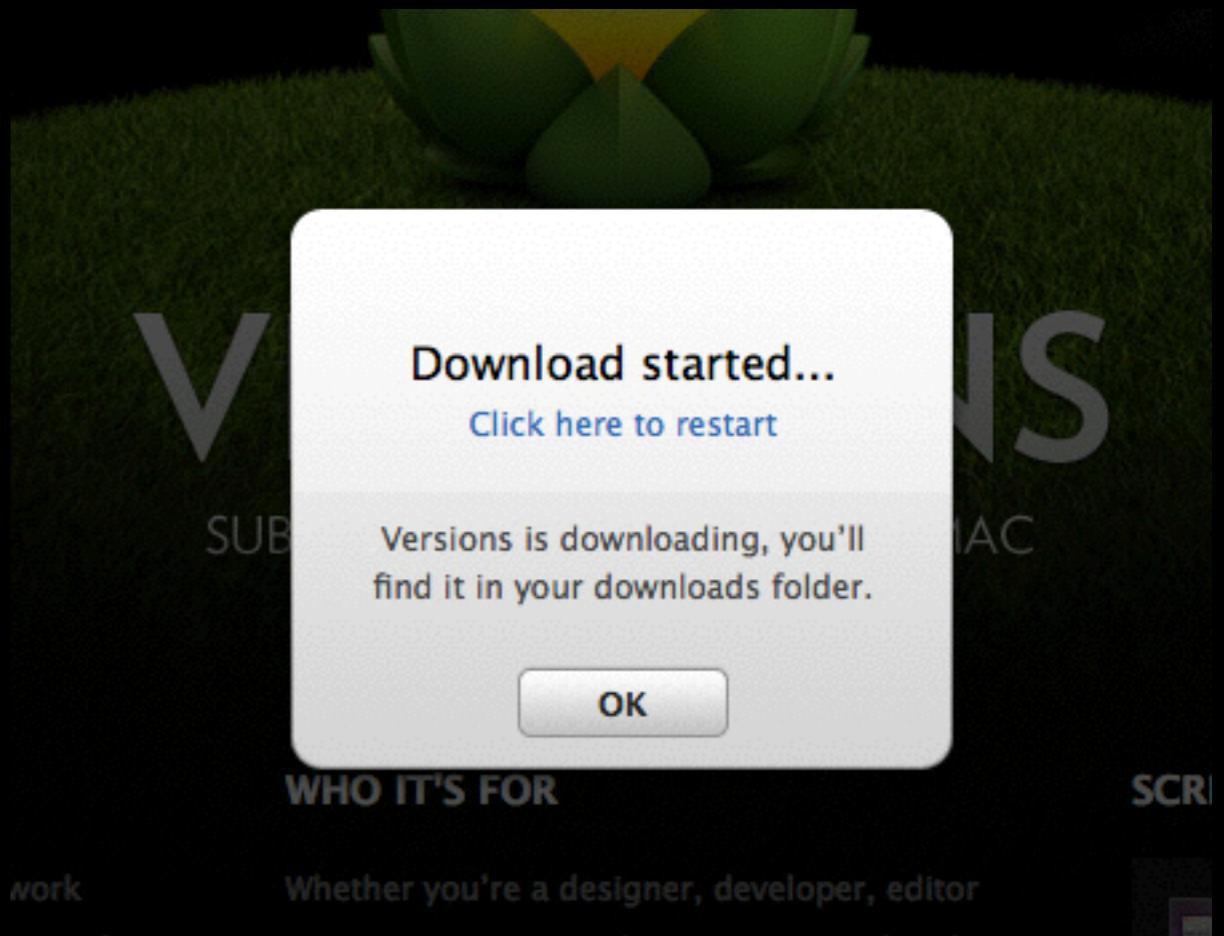
Highlighting Invalid Data & Error Messages in Page Title

- **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;`**



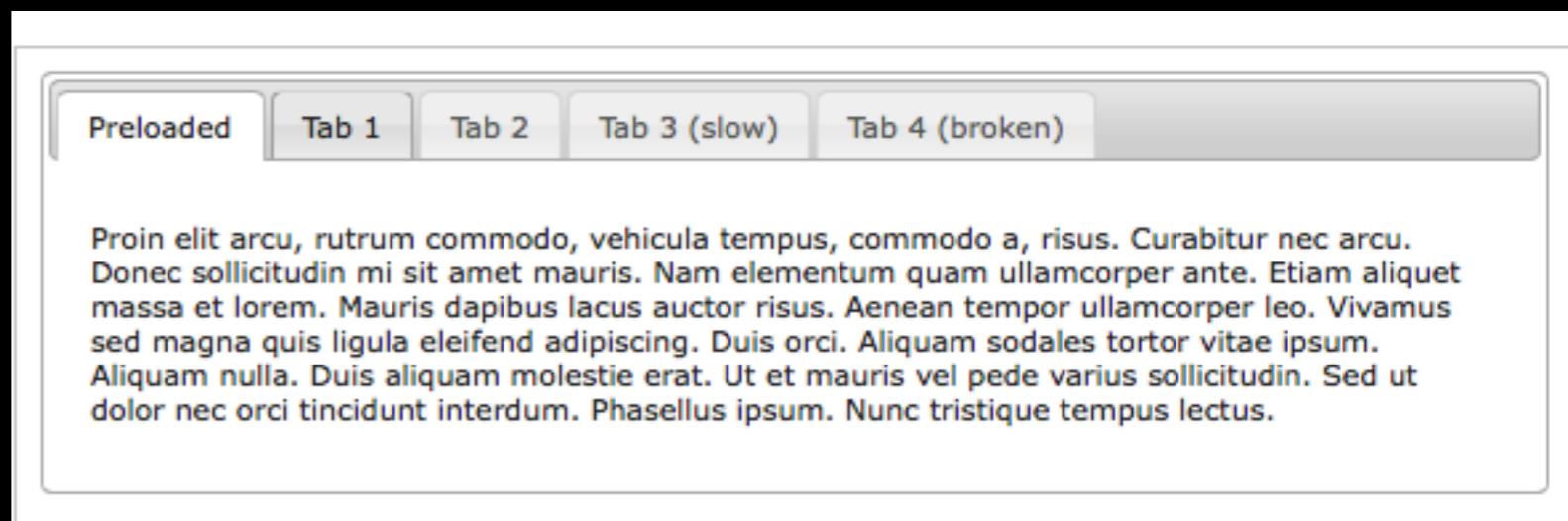
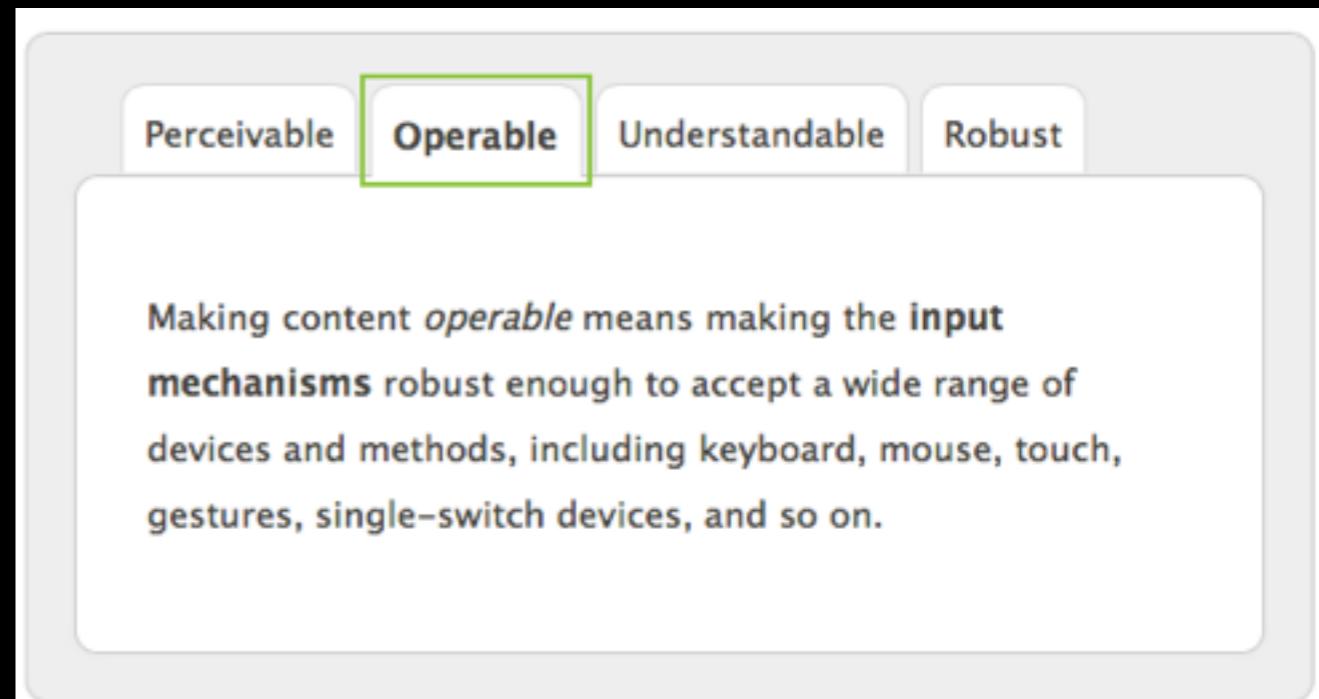
Accessible Alert & Modal Dialogs with ARIA

- **role=dialog, role=alertdialog,
aria-labelledby, aria-
describedby, aria-hidden**
- [http://pauljadam.com/
demos/dialog.html](http://pauljadam.com/demos/dialog.html)
- [http://pauljadam.com/
demos/aria-role-
alertdialog.html](http://pauljadam.com/demos/aria-role-alertdialog.html)
- [http://pauljadam.com/
demos/dialog-DOM.html](http://pauljadam.com/demos/dialog-DOM.html)

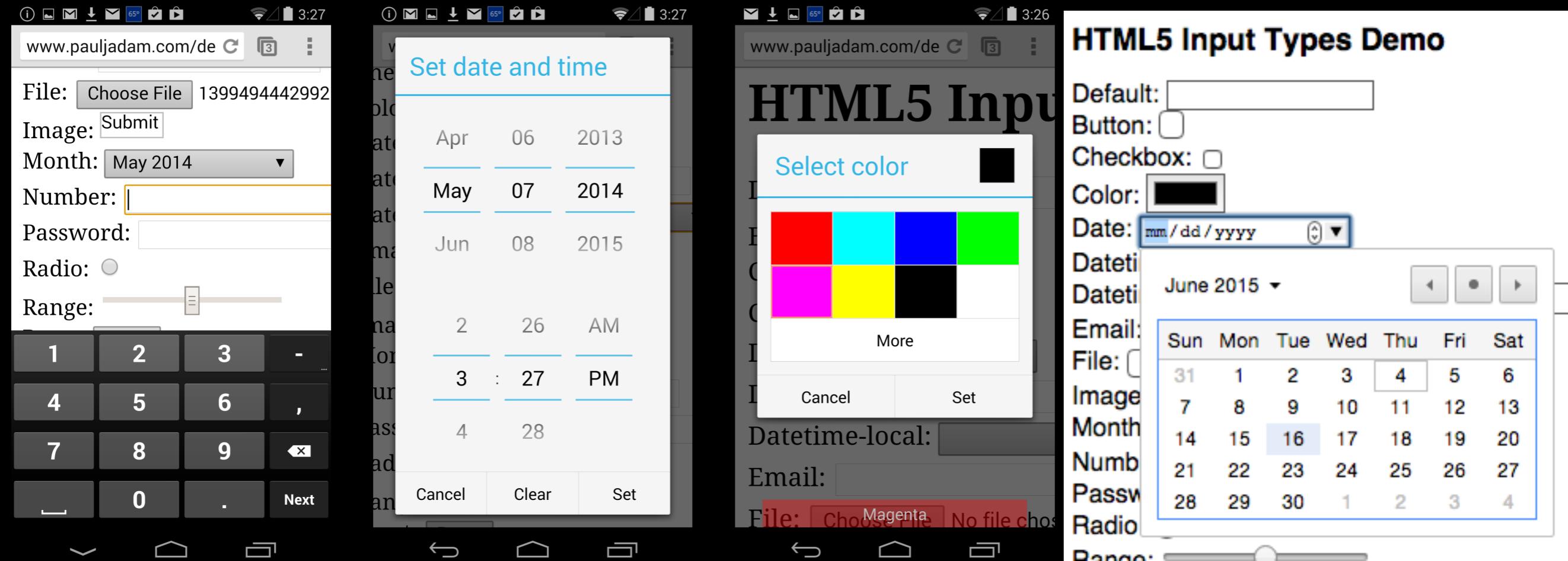


Tabs and Expandable/ Collapsible Widgets

- (**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>



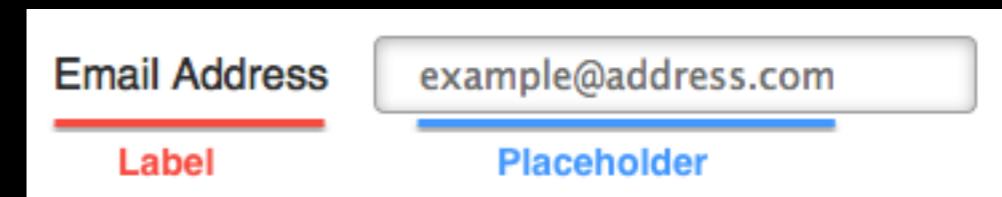
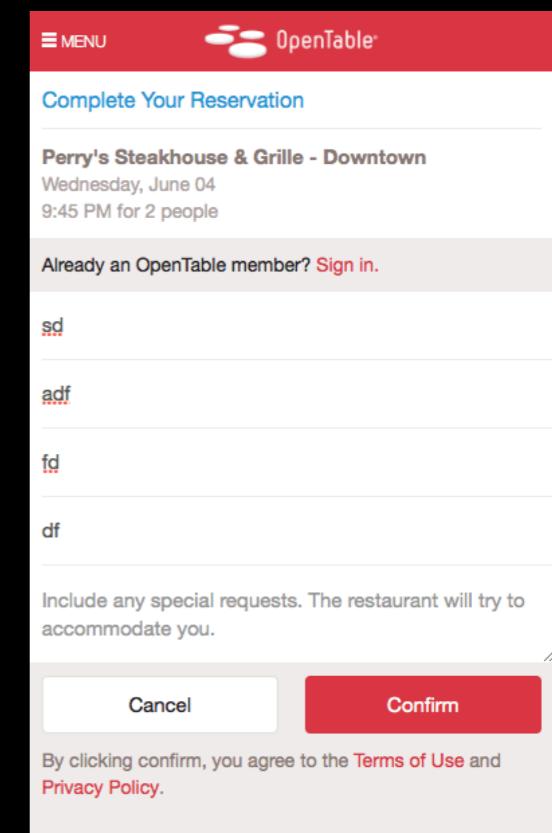
HTML5 Input Types



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

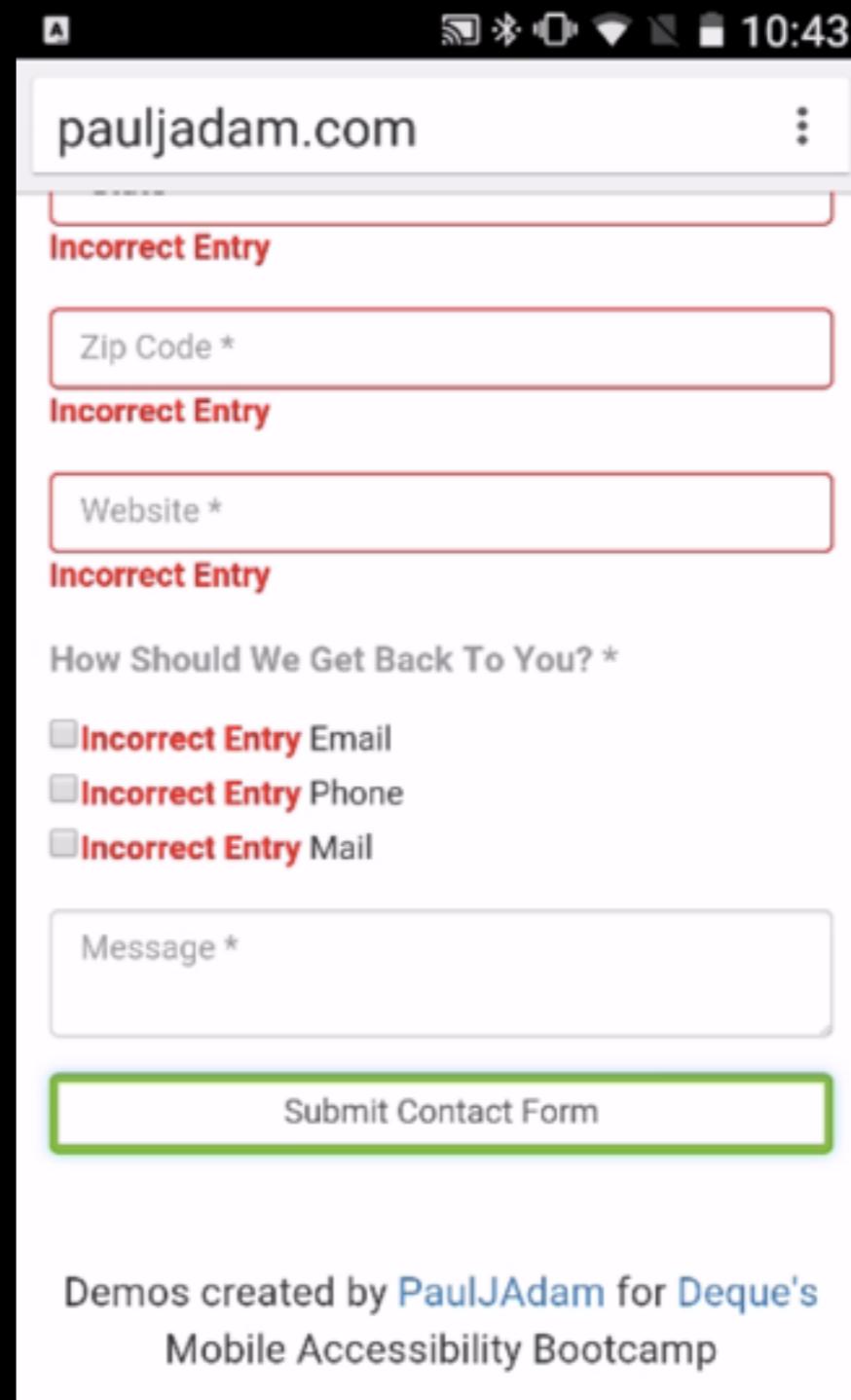
Labels vs. Placeholders

- **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**

A screenshot of a mobile application interface. It features three input fields stacked vertically. The top field has a single character "I" in its input area. The middle field is labeled "Last Name" and the bottom field is labeled "Email Address", both in a light gray placeholder font.

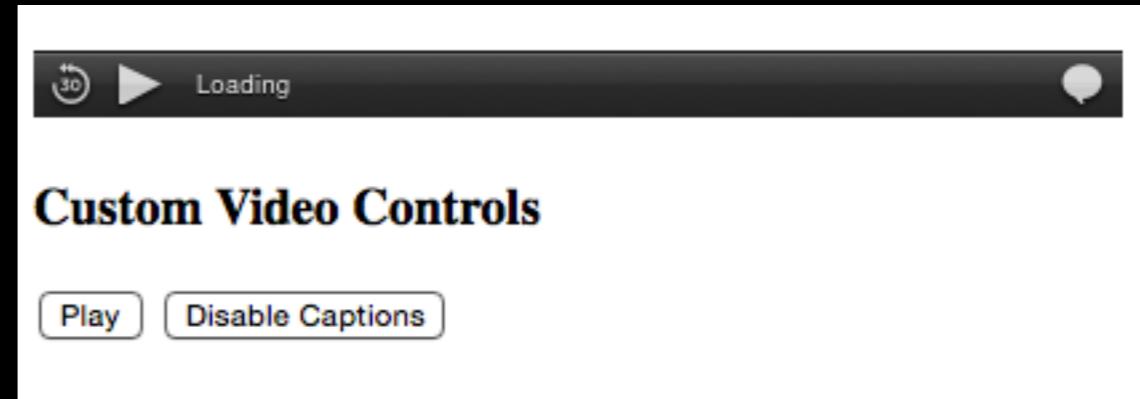
Hands-On ARIA Landmarks & Forms Exercise

- 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>**



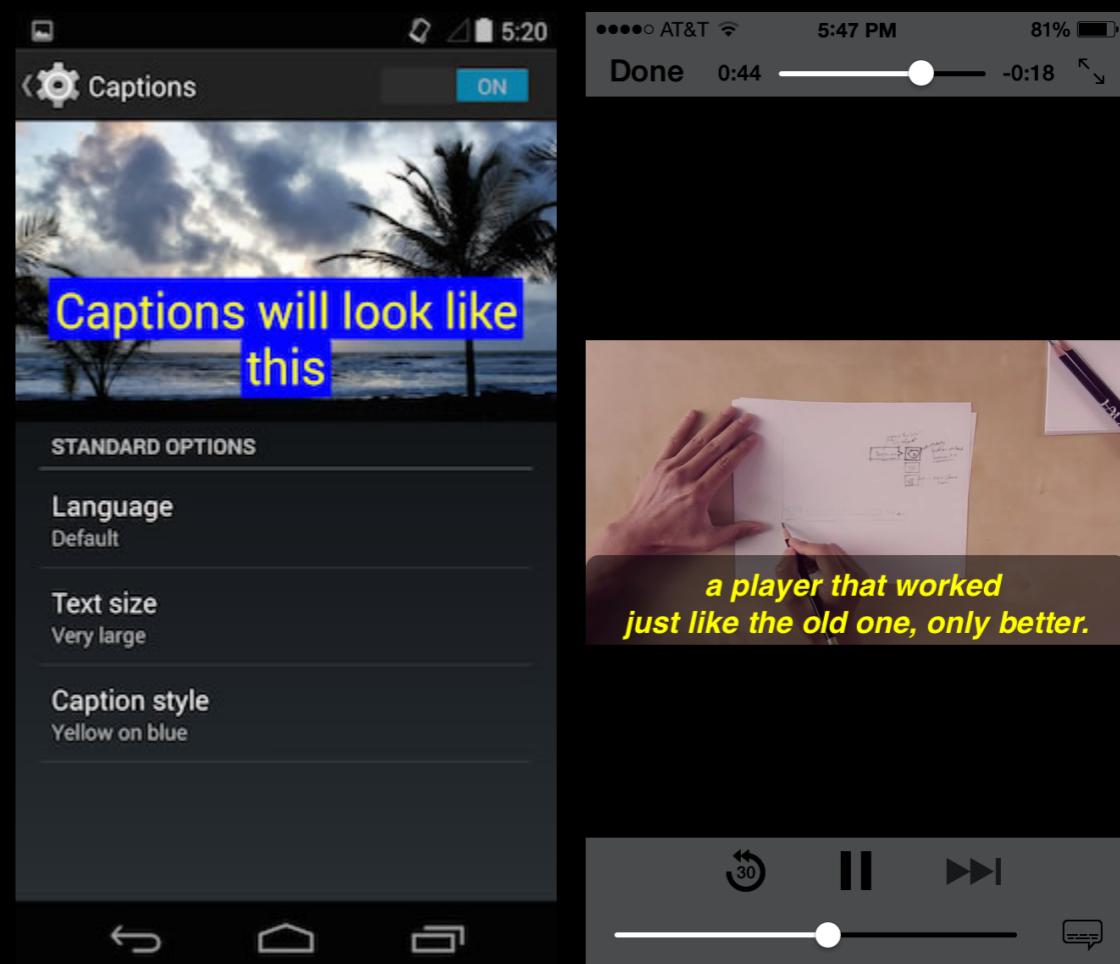
HTML5 Video Keyboard & Mobile Accessibility

- 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

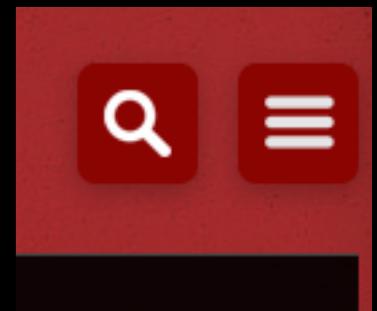
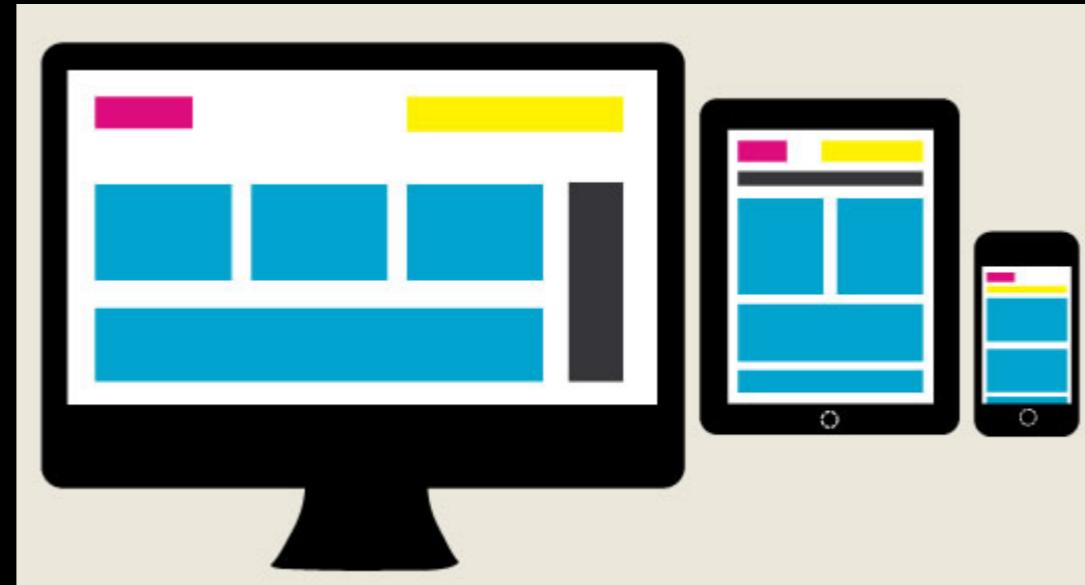
- Add **input types**, `type=tel`, `type=email`, etc.
- Replace `aria-required` with **HTML5 required**
- Add **autofocus** to first input

A screenshot of a mobile device displaying a web page from pauljadam.com. The page contains a form with fields for "Phone # *", "Street Address *", "Apt, Suite or Floor (Optional)", and "City *". An input field for "Phone # *" has a red asterisk indicating it is required. A numeric keypad is overlaid on the bottom right of the screen, with standard digits 1-9, a decimal point, and special characters like *, #, 0, +, and -.

Responsive Web Design

The screenshot shows the Main Line Baseball Academy website. On the left is a vertical navigation menu with links: Close, Home, About Us, The Coaches, Our Mission, Our Philosophy, What We Offer, Teams, Cavaliers, Barnstormers, Barons, Upcoming Events, Calendar, and Contact Us. At the top right is a yellow header bar with the Main Line Baseball Academy logo and the text "MAIN LINE BASEBALL ACADEMY SAGACITAS • POTENTIA • CORPUS". The main content area features a quote by Babe Ruth: "IT'S HARD TO BEAT A PERSON WHO NEVER GIVES UP" with a attribution "-Babe Ruth". Below the quote is a subtext: "We'll give them the edge they need to do just that." There is also a section titled "OUR MISSION" with a photo of a baseball player and a brief description.

<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**

The screenshot shows a dark-themed web page with several sections. At the top, there are two buttons: "Collapse" and "Collapse After". Below these are two large, dark rectangular blocks. The first block contains the text "Shipping & Returns" followed by a small icon. The second block contains a paragraph of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat." Below this, another identical paragraph is shown.

Collapse

Collapse After

Shipping & Returns

Lore ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lore ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

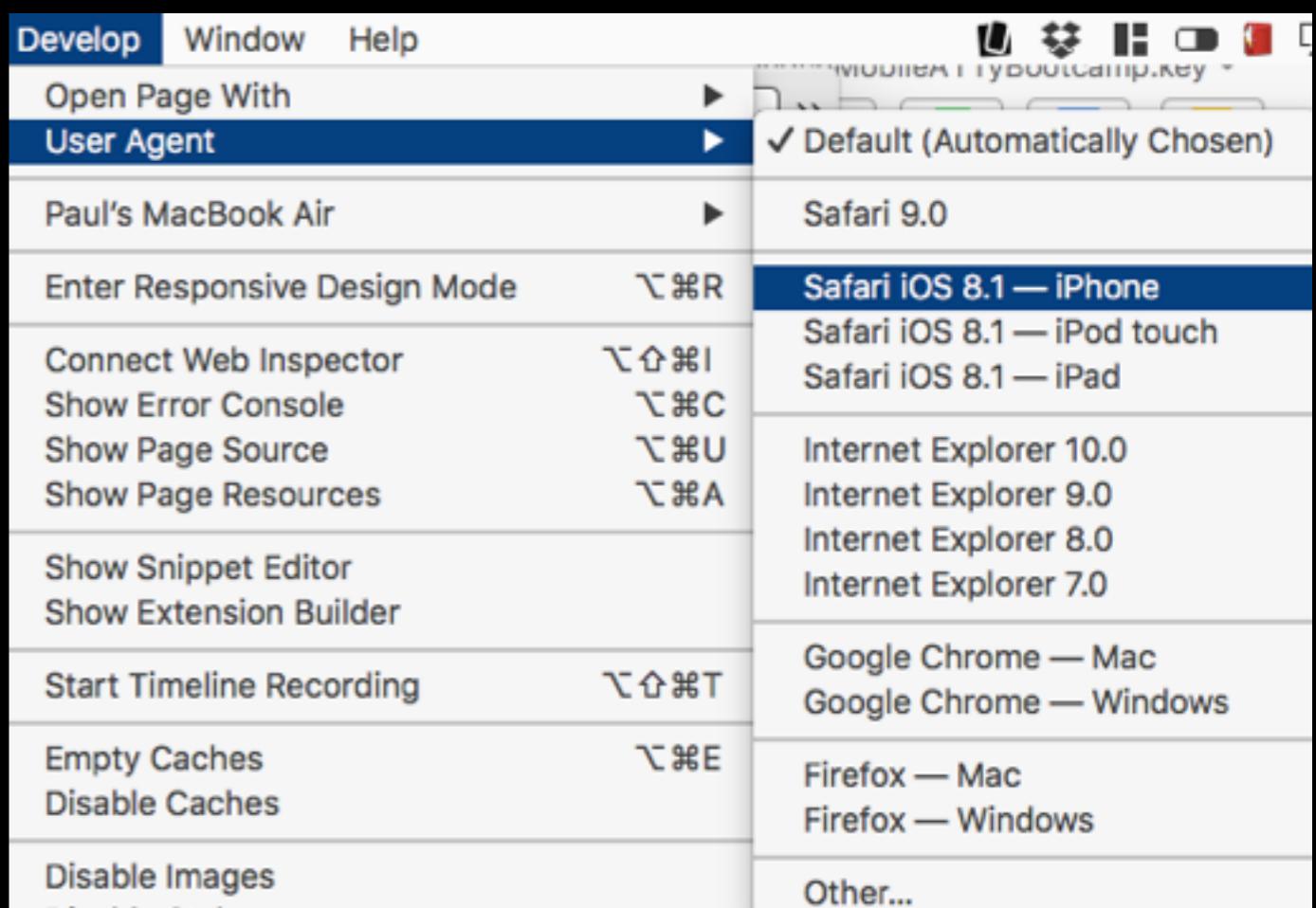
Mobile Accessibility Testing Tools

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



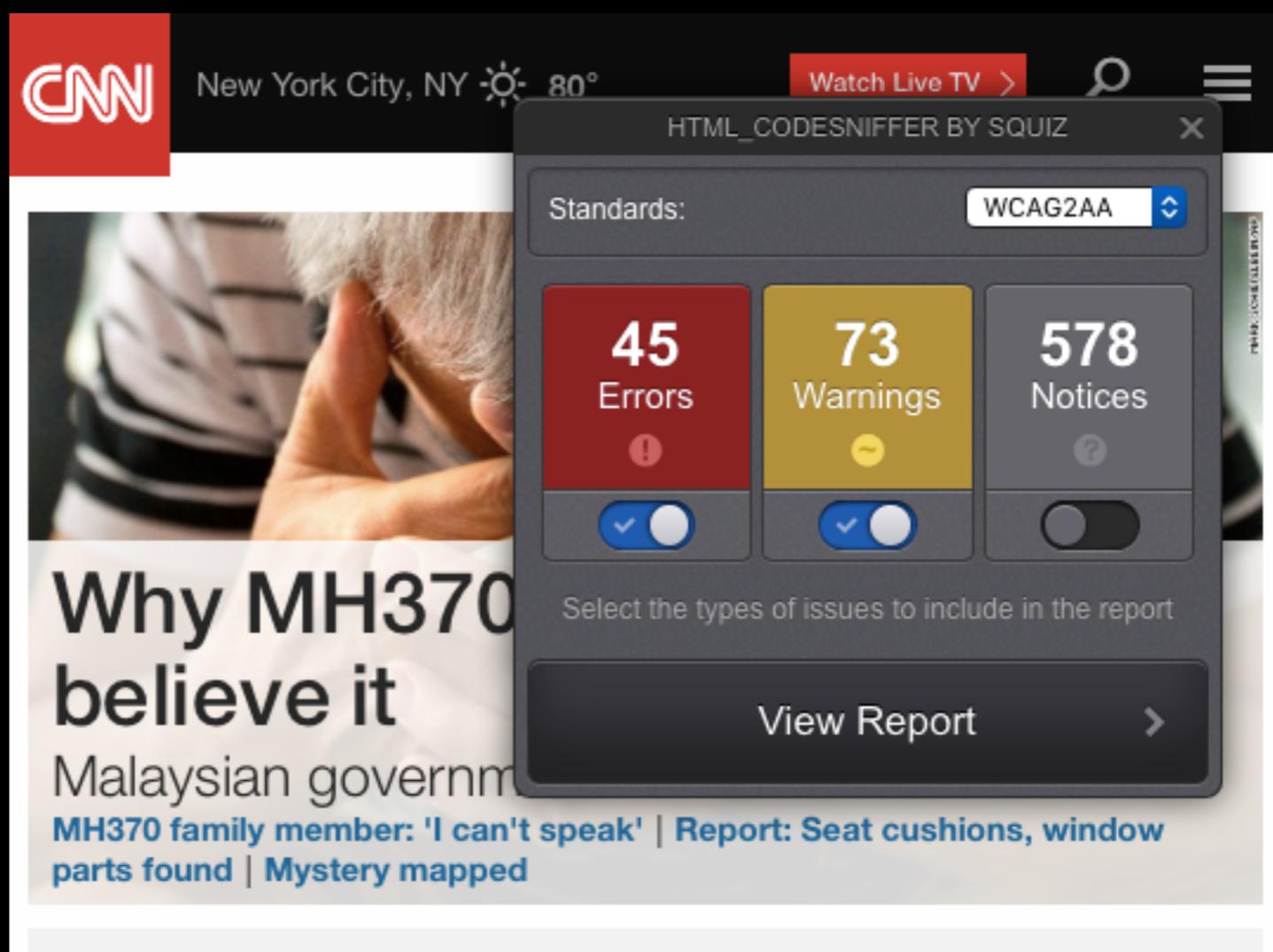
Mobile User Agents & Web Inspecting

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



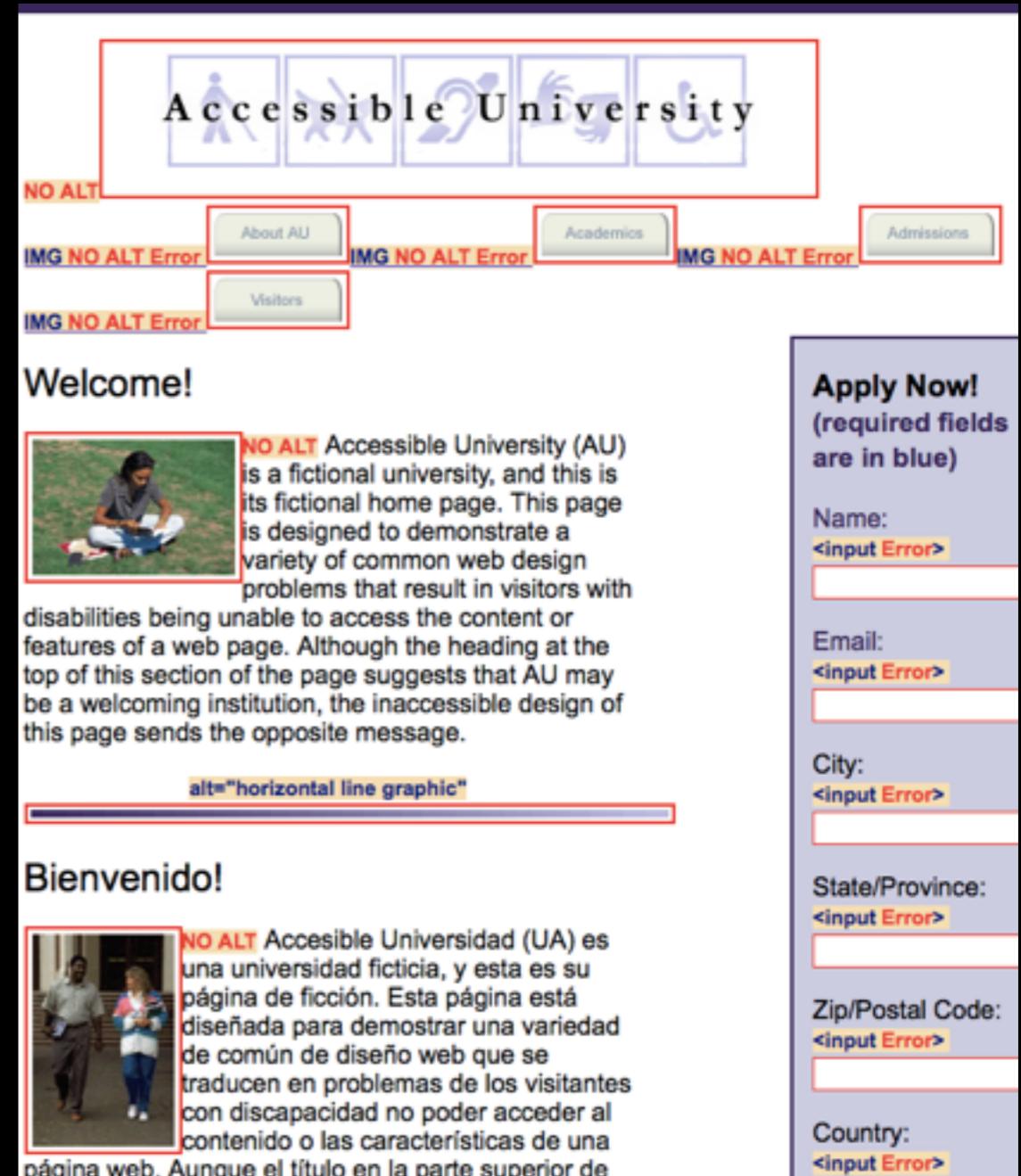
JavaScript Bookmarklets

- **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**

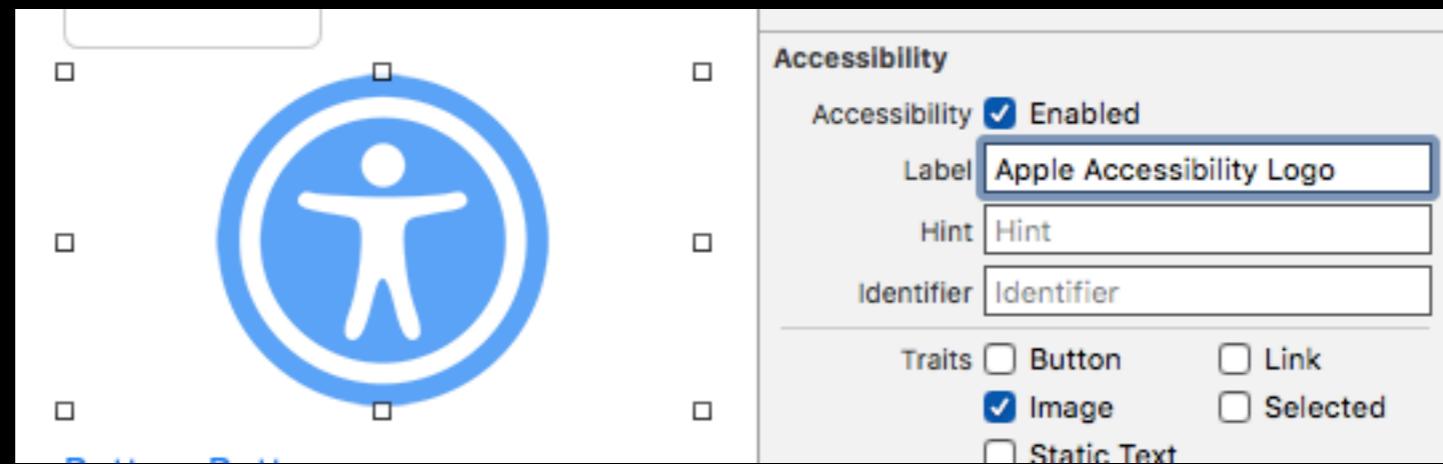


Hands-On Activity Test Before & After Demos with Testing Tools

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



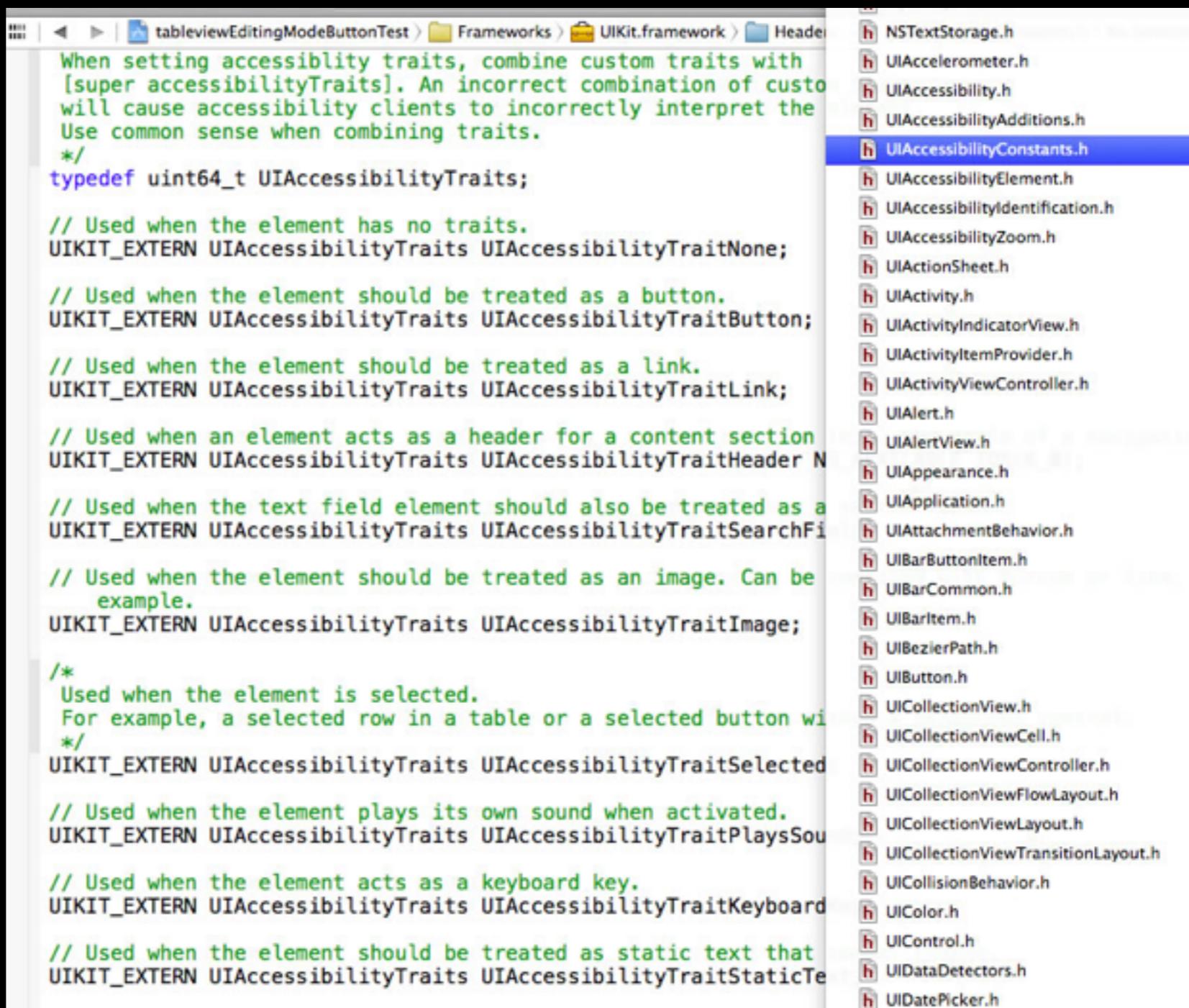
Native iOS App Accessibility



- developer.apple.com
- [https://developer.apple.com/
accessibility/ios/](https://developer.apple.com/accessibility/ios/)

UIAccessibility API

- (Label, Hint, Trait, etc.)
- <http://pauljadam.com/iosvsandroid11y/#native>



The screenshot shows the Xcode interface with the file `UIAccessibilityConstants.h` open. The file contains various `UIKIT_EXTERN` declarations for `UIAccessibilityTraits` constants. The file path in the top bar is `Frameworks > UIKit.framework > Headers`. The right side of the screen shows a list of other header files in the project.

```
When setting accessibility traits, combine custom traits with [super accessibilityTraits]. An incorrect combination of custom traits will cause accessibility clients to incorrectly interpret the element. 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;

// Used when the text field element should also be treated as a search field.
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitSearchField;

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

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

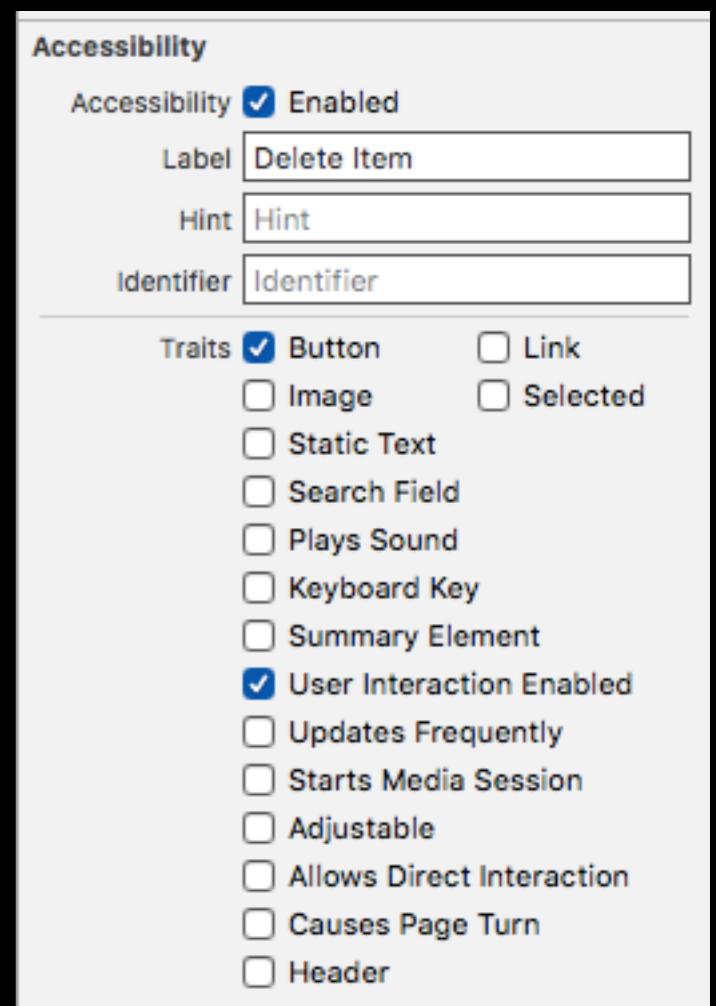
// Used when the element plays its own sound when activated.
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitPlaysSound;

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

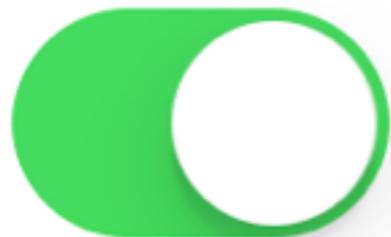
// Used when the element should be treated as static text that is not intended to be modified by the user.
UIKIT_EXTERN UIAccessibilityTraits UIAccessibilityTraitStaticText;
```

Xcode Overview

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



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

```
/* [REDACTED]
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.

```
*/ [REDACTED]
UIKIT_EXTERN void UIAccessibilityPostNotification(UIAccessibilityNotifications notification, id argument);
```

Easy Modal Dialog!

```
/* [REDACTED]
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.
```

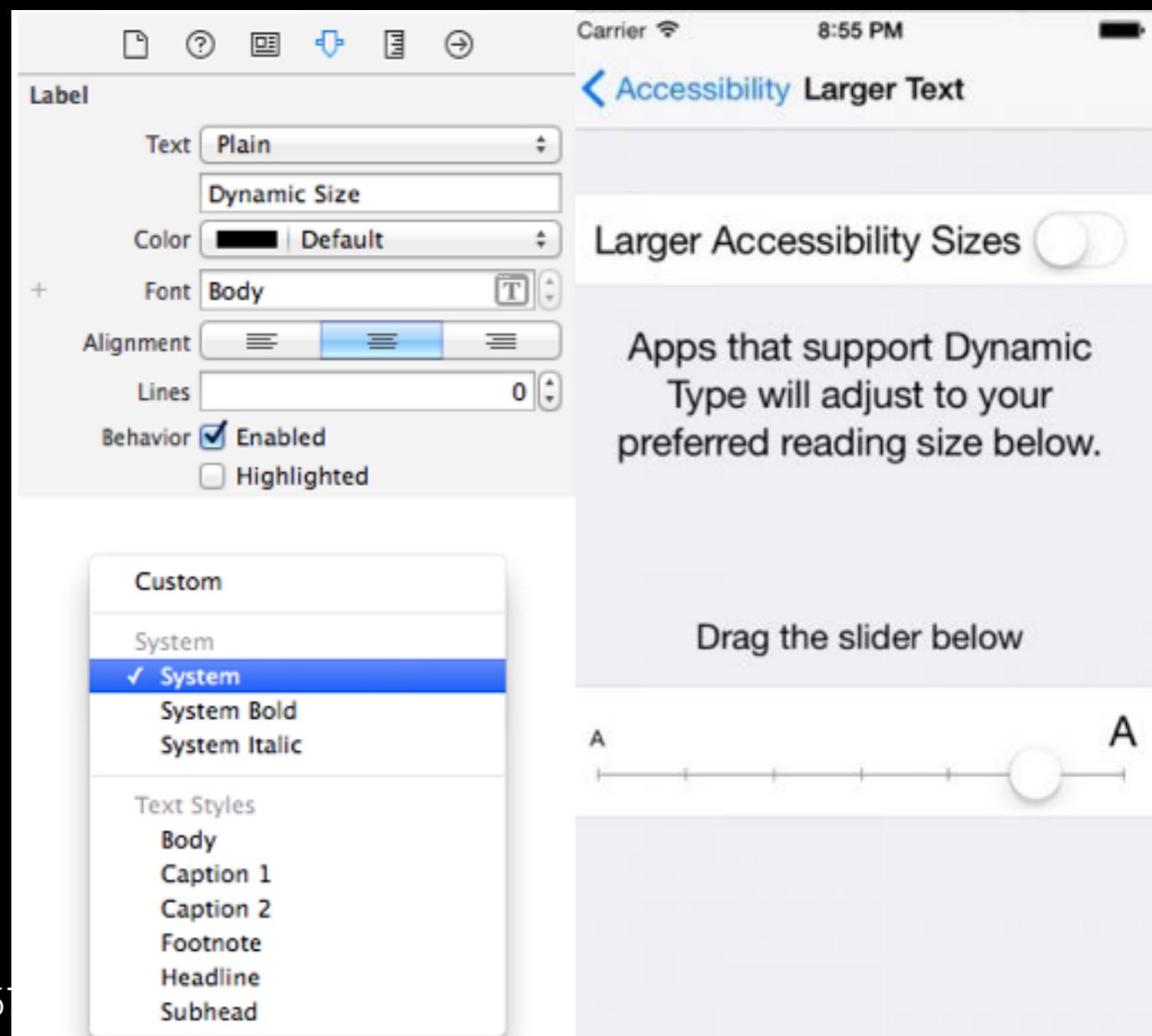
default == NO

```
*/ [REDACTED]
@property(nonatomic) BOOL accessibilityViewIsModal NS_AVAILABLE_IOS(5_0);
```

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

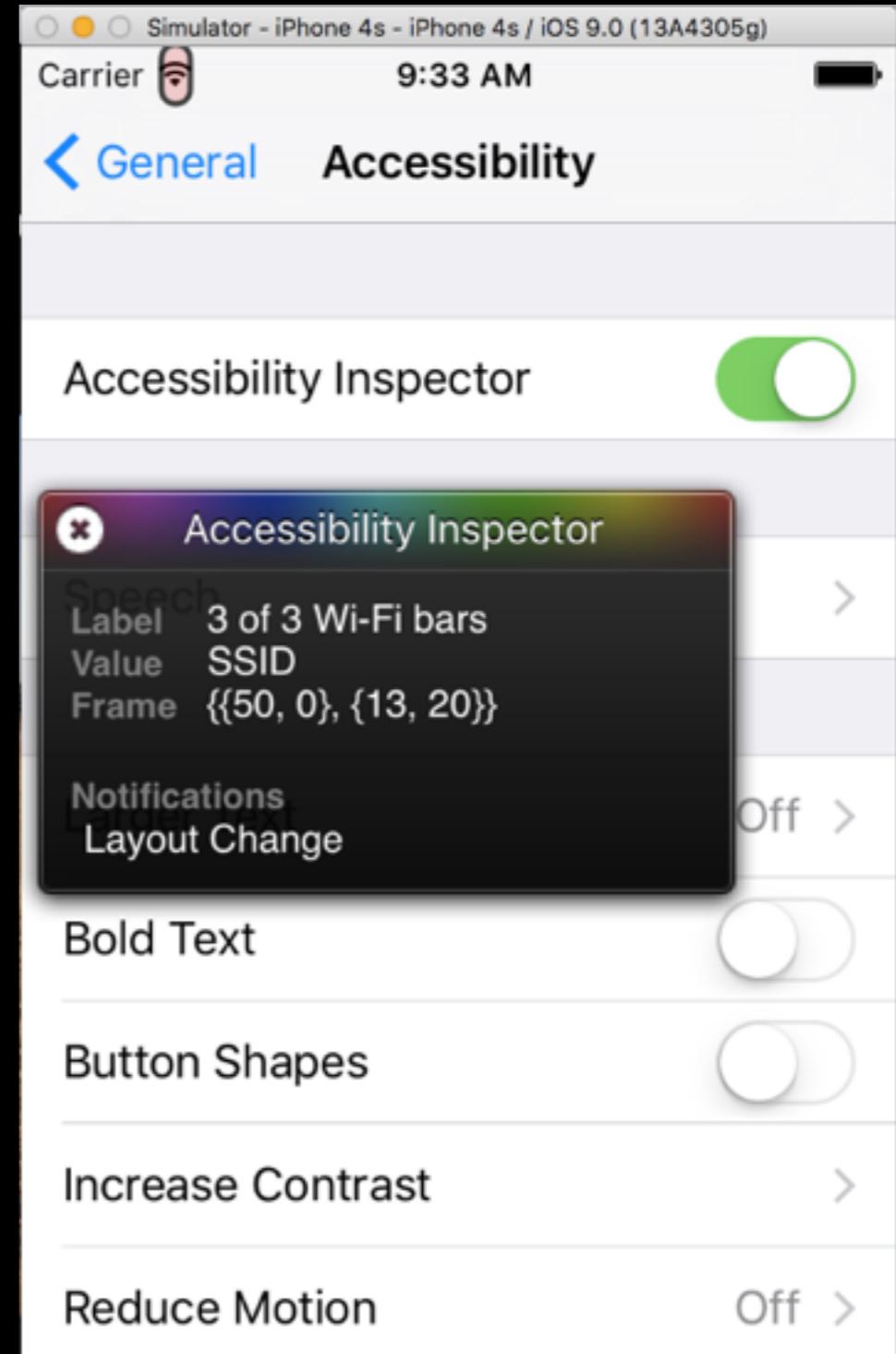
Dynamic Type

- <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

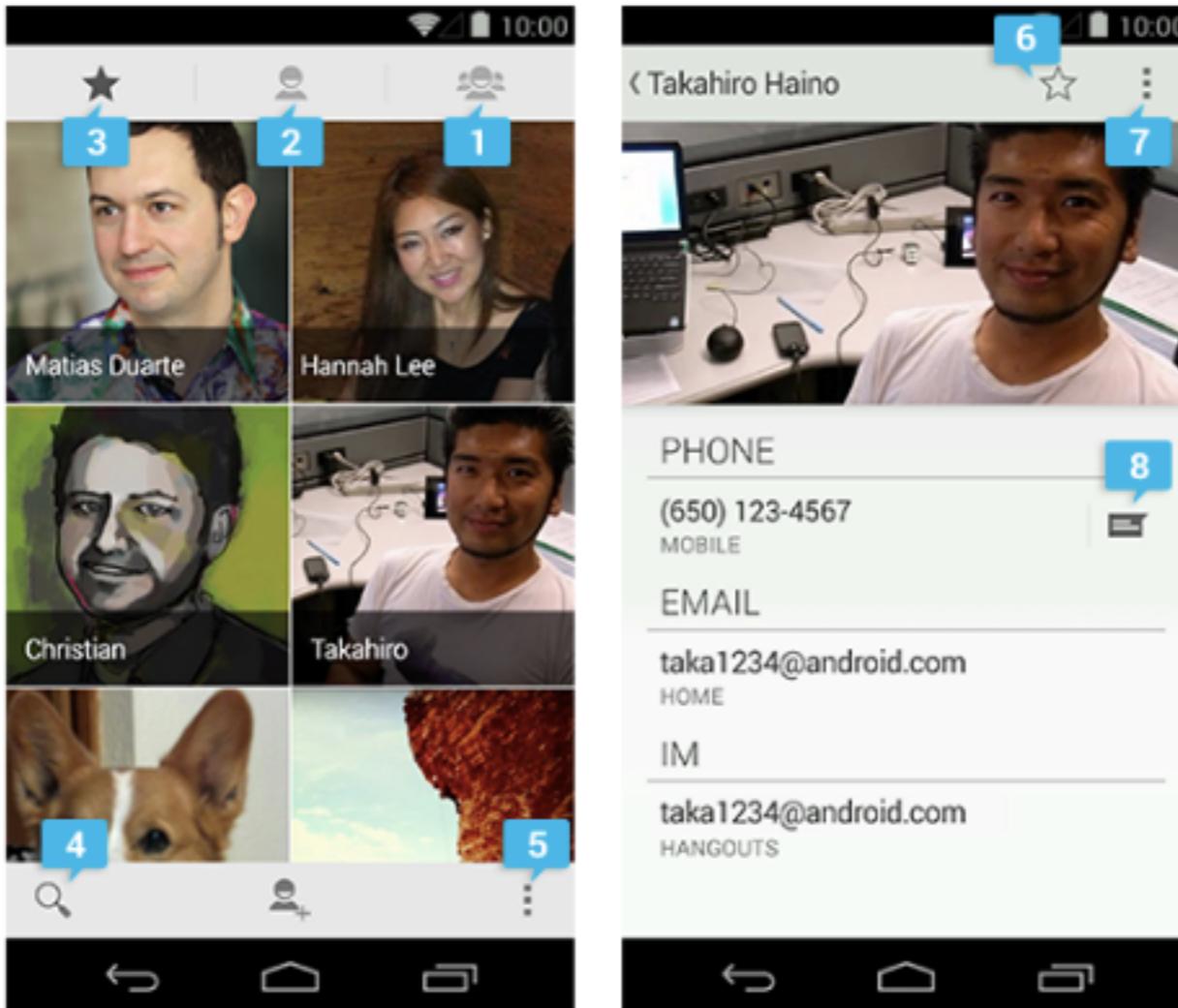
- [https://www.apple.com/
accessibility/watch/](https://www.apple.com/accessibility/watch/)



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.



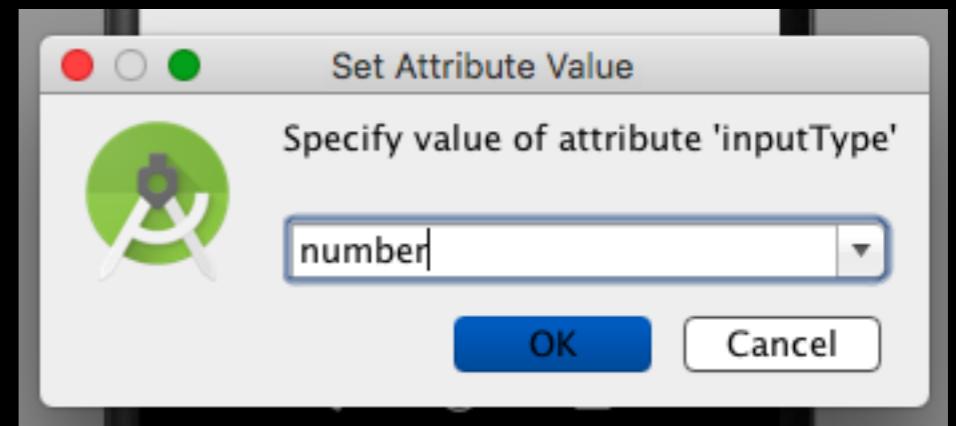
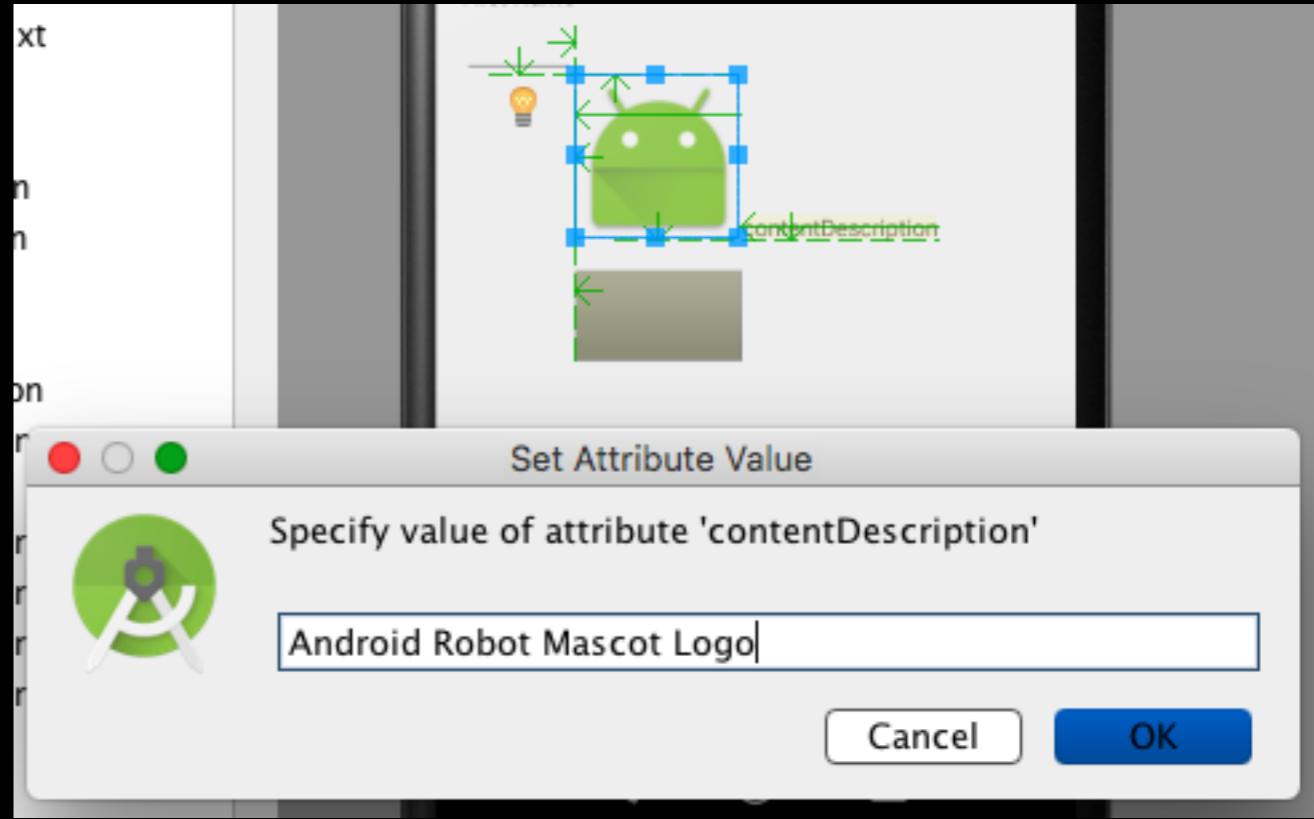
1. group
2. all contacts
3. favorites
4. search
5. action overflow button
6. when starred: remove from favorites
when not starred: add to favorites
7. action overflow button
8. text message

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

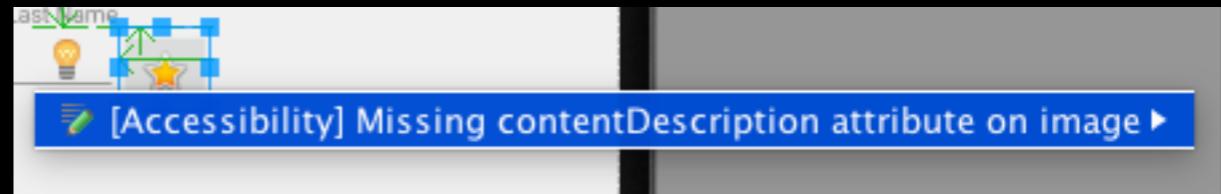
- **Fixing Accessibility Issues in Design View, XML View, Java Code View**
- **Android Lint Accessibility Tooltip Errors**
 - **Add missing accessibility properties via error dialogs.**



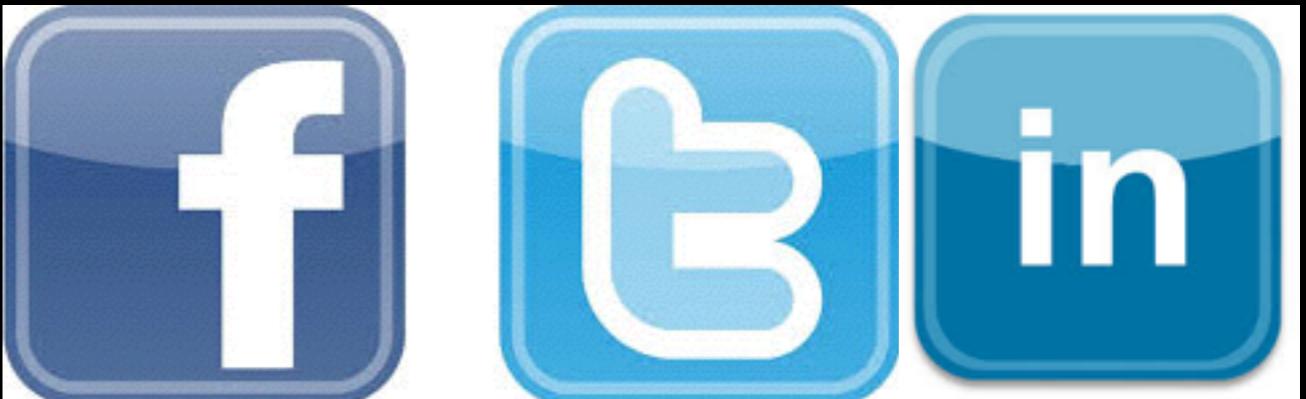
Android Lint Accessibility Errors & Solutions

- **<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**

- **@dequesystems**



- **LinkedIn**

- **Deque Systems**



- **Web**

- **www.deque.com**



- **Email**

- **info@deque.com**

