WEB ACCESSIBILITY

by Paul Collins

“The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect”.

Tim Berners-Lee, W3C Director and inventor of the World Wide Web

What we'll cover

- A look at ARIA (Accessible Rich Internet Applications)
- Accessibility evaluation tools
- Some accessibility issues and solutions on the ANZ KYC project

Please jump in with a question anytime if you have one!

USING ARIA

Extending your markup using ARIA

- ARIA is a set of attributes you can add to your markup to meet Accessibility requirements.
- These attributes communicate role, state and property semantics to assistive technologies via the accessibility APIs implemented in browsers.

Browser support for ARIA

http://caniuse.com/#feat=wai-aria

ARIA Core components

Roles

Define what an element is or does.

Properties

Properties or meaning of elements. EG: <input aria-required=“true”>

States

Define the current condition of an element.

EG: <input aria-disabled=“true”>

ARIA Landmark roles

Designed to help developers meet accessibility requirements while browsers catch up to HTML5 support. Assistive software provides shortcut keys to navigate to these elements.

http://www.html5accessibility.com/

- **banner:** Site content, such as page title, log
- **navigation:** area that contains the navigation for the site
- **main:** The main or central content of the document
- **search:** The section contains the search functionality of the site
- **article:** Stand alone content that makes sense on it's own
- **complementary:** Supporting content info for the document.
- **contentinfo:** Informal child content, such as footnotes, copyright, etc.

- EG: <ul role="navigation">, <div role="main">, <form role="search">

ARIA States and Properties

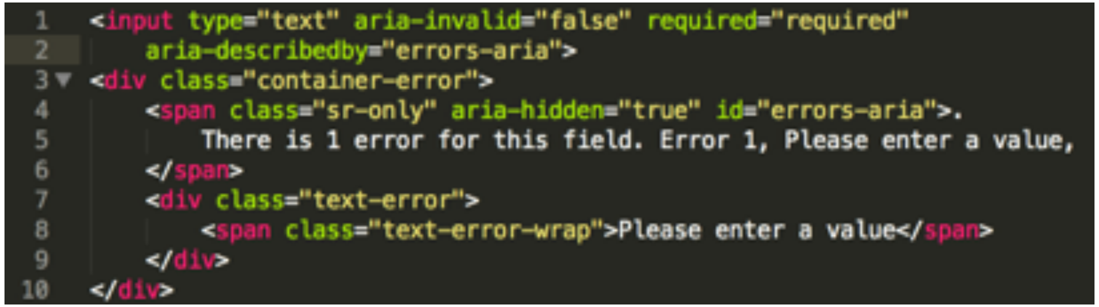
- A state is used for content that changes dynamically, it explains the current state. Some examples are aria-hidden="true", aria-expanded="true", aria-invalid="true"

- A property is used to describe the meaning of an element. Used to extend native HTML for elements that aren't allowed. Some examples are aria-required="true" aria-described-by="…".

- Also worth noting, you can add tabindex="0" to any element on a page and ARIA will add it to the normal tab flow of the document.

Definitive list of states and properties

https://www.w3.org/TR/wai-aria/states_and_properties



```
<input type="text" aria-invalid="false" required="required"
    aria-describedby="errors-aria">
<div class="container-error">
    <span class="sr-only" aria-hidden="true" id="errors-aria">.
        There is 1 error for this field. Error 1, Please enter a value,
    </span>
    <div class="text-error">
        <span class="text-error-wrap">Please enter a value</span>
    </div>
</div>
```

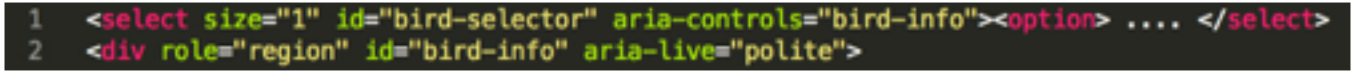
ARIA Live regions

Designed to alert the user to a change in content. If the HTML element has the aria-live attribute, with a value of off, polite, assertive or rude, the screen reader will react accordingly.

- **aria-live="off"** tells the screen reader not to announce the update
- **aria-live="polite"** will notify the user of the content change once they are done with the current task.
- **aria-live="assertive"** will notify the user of the content change as soon as it appears.
- **aria-live="rude"** will interrupt the user, should be used sparingly.

Live regions intro

http://mzl.la/1JseeN9



```
<select size="1" id="bird-selector" aria-controls="bird-info"><option> .... </select>
<div role="region" id="bird-info" aria-live="polite">
```

TESTING TOOLS

These should compliment “human” testing

HTML Code Sniffer bookmark

http://squizlabs.github.io/HTML_CodeSniffer/

Cynthia Says

Generate a page report. (Site must be live and public facing though).

http://www.cynthiasays.com/

Colour contrast analyser

https://www.paciellogroup.com/resources/contrastanalyser/

Chrome Vox

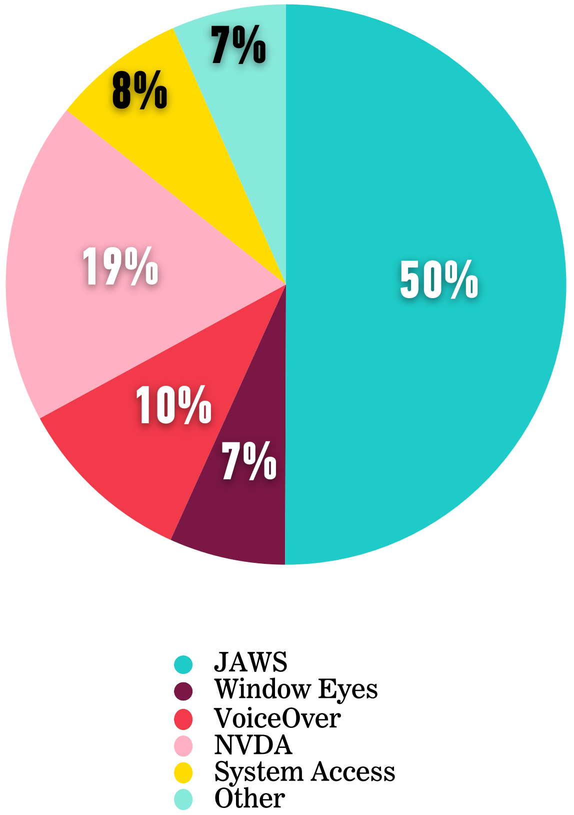
Chrome screen reader extension

http://www.chromevox.com/ Tutorial: http://www.chromevox.com/tutorial/index.html

Screen reader statistics:

Full report
http://webaim.org/projects/screenreadersurvey5/

(Chrome Vox is 0.4% of users)



Chrome Vox keyboard commands

- TAB = Next element
- SHIFT + TAB = Previous element
- ENTER = Select current element (eg: link or button)
- SPACEBAR = Select radio button

ACCESSIBILITY ISSUES ON KYC

Two scenarios

- Accessibility on hamburger menu
- Maintaining keyboard focus on modals

* Keyboard access is also a requirement for non-disabled users - e.g. data entry at branches.

CHEERS!