intopia

Headings, Tables, Landmarks

Engineers - Session 1

My name is Russ Weakley:

- 1995: Web Design and HTML
- 2003: Accessibility
- 2012: Component libraries and Design systems

We also have the amazing **Kiya Doig** in the meeting, from the Accessibility team!

Is everyone ok if we **record this session**?

For all three topics, we will run through some slides, and then **review some existing examples from within your site**.

Your input would be really valuable during these reviews - **questions, comments, suggestions**.

Some tools

- Chrome extension: Accessibility Insights for Web
- Chrome extension: Skip to Landmarks and Headings
- Bookmarklet: Easy Checks headings checker

Part 1: Headings

Why are headings important for accessibility?

Headings communicate **how the content on web pages is structured** for sighted users and assistive technology users.

For people with **limited or no vision**, headings help:

- Understand the overall page structure
- Navigate to content with minimal effort
- Understand the content they are reading

JAWS and NVDA

Key/function	Purpose
`H`	Jump to next heading
`SHIFT + H`	Jump to previous heading
`1, 2, 3, 4, 5, 6`	Jump to next heading of that type
`SHIFT + 1, 2, 3, 4, 5, 6`	Jump to previous heading of that type
Elements list popup	View all headings

For people with **limited movement**, heading help:

- Navigate to content with minimal effort

For people with **limited learning or cognition**, headings help:

- Skim content quickly and effectively

- Build mental models to understand the content

Heading hierarchy

Headings should be **correctly nested in order**, starting with the most important heading - `<h1>`, down to the least important heading - `<h6>`.

```
<h1>Site name or Page title</h1>
    <h2>Section</h2>
        <h3>Subsection</h3>
        <h3>Subsection</h3>
    <h2>Section</h2>
        <h3>Subsection</h3>
```

Skipping heading ranks can confuse some users and **should be avoided where possible**.

```
<h1>Site name or Page title</h1>
    ~~<h5>Subsection</h5>~~
    ~~<h5>Subsection</h5>~~
    ~~<h5>Subsection</h5>~~
```

While heading levels should be a design problem, it often **falls to engineers to resolve within layouts**.

You can use your Heading classes to change the visual appearance of headings, while **maintaining heading hierarchy**.

```
<h3 class="xui-heading-2xlarge">...</h3>
```

What should you do when heading levels are **defined within components**, like the XUI Accordion?

In these cases, you just have to **work around these heading** as best you can until they are resolved within the system.

Site or page headings

Every page should have an `<h1>` **near the start of the page**.

This `<h1>` should be reserved for **the most important information on the page** - such as the site name of primary page heading.

While it is not considered a WCAG failure to have more than one `<h1>` per page, it is a good practice to **limit pages to a single** `<h1>`.

Avoid ‘fake’ headings

While the following markup may produce content that visually looks like a heading, it **lacks meaningful semantic structure**.

```
<p class="xui-heading-xlarge">Fake page heading</p>
```

For many people who rely on actual heading structure, this type of markup **would be ignored completely**.

Always **try to use actual headings** throughout your layouts.

Some examples

Some example pages

- Business: Short-term cash flow projection - 2 x H1
- Business: Business snapshot - multiple H1
- Accounting: Reports - 2 x H1 and no other headings (Favourites, Financial performance, Financial statements etc)

Part 2: Tables

A quick overview of basic table markup

You have a nice table component that uses all of the correct elements.

List	Purpose
`<table>`	Presents tabular data in a two-dimensional table
`<thead>`	Defines a set of rows defining the head of the columns of the table.
`<tbody>`	Defines a set of rows, indicating that they comprise the body of the table.
`<tr>`	Defines a row of cells in a table
`<td>`	Defines a cell of a table that contains data

The <th> element

The <th> element defines **a cell as the header of a group of table cells** (table header).

Each column header **must be defined** as a <th> element.

```
<table>
  <thead>
    <tr>
      <th></th>
      <th></th>
      <th></th>
    </tr>
  </thead>
...
```

Header cell	Header cell	Header cell
Table cell	Table cell	Table cell
Table cell	Table cell	Table cell
Table cell	Table cell	Table cell

The <th> can also be used at the left side of the table for cells that **define the entire row**.

```
...
  </thead>
  <tbody>
    <tr>
      <th></th>
      <td></td>
      <td></td>
    </tr>
  </tbody>
</table>
```

Header cell	**Header cell**	**Header cell**	**Header cell**
Header cell	Table cell	Table cell	Table cell
Header cell	Table cell	Table cell	Table cell
Header cell	Table cell	Table cell	Table cell

The `<th>` is important for screen readers as it **allows users to orientate header cells to table cells** within the table.

Local bus times

	205 Bus	**207 Bus**	**212 Bus**
Duloc St	8.20am	12:00pm	4:00pm
Maple St	8:30am	12:10pm	4:10pm
Drury Lane	8:30am	12:20pm	4:20pm

The <caption> element

The <caption> element provides the table with:

- A visible label - for sighted users.
- An acccessible name - for assistive technologies.

The <caption> element must come directly after the <table> element's start tag.

```
<table>
  <caption>Table caption</caption>
</table>
```

Heading elements can be placed inside the <caption> to **provide additional styling and semantics**.

```
<table>
  <caption>
    <h3>Table caption</h3>
  </caption>
</table>
```

The `aria-label` and `aria-labelledby` can be used to create **accessible names for your tables**.

```
<table aria-label="Business updates">
  ...
</table>
```

```
<p id="aaa">A table caption</p>
<table aria-labelledby="aaa">
  ...
</table>
```

Let's look at a demo codepen.

Some examples

Some example pages

- Business snapshot: Largest operating expenses - a good example
- Invoices - lack of TH, dropdown arrow, empty cells
- Purchase orders - not a table
- Products and services - role=button on TH

Using ARIA to create table structure

ARIA can also be used to create entire tables via **ARIA table roles**.

Native	ARIA
`<table>`	`<div role="table">`
`<thead>`	`<div role="rowgroup">`
`<tbody>`	`<div role="rowgroup">`
`<tr>`	`<div role="row">`
`<th>`	`<span role="columnheader">`
`<td>`	`<span role="cell">`

Part 3: Landmarks

Why are landmarks important for accessibility?

Just like headings, landmarks communicate **how the content on web pages is structured** for sighted users and assistive technology users.

For people with **limited or no vision**, landmarks help:

- Understand the overall page structure
- Navigate to content with minimal effort
- Understand the content they are reading

For people with **limited movement**, landmarks help:

- Navigate to content with minimal effort

HTML and ARIA landmarks

ARIA introduced a range of landmark roles to **identify page regions for assistive technologies**.

```
<div role="navigation">
  <ul>
    <li>Home</li>
    <li>About</li>
  </ul>
</div>
```

Some time later, HTML5 introduce new elements to **define page regions**.

```
<nav>
  <ul>
    <li>Home</li>
    <li>About</li>
```

```
  </ul>
</nav>
```

There are now **two ways to identify landmarks**, HTML elements and ARIA roles.

Landmark	ARIA role	HTML element
Banner	`role="banner"`	`<header>`
ContentInfo	`role="contentinfo"`	`<footer>`
Navigation	`role="navigation"`	`<nav>`
Main	`role="main"`	`<main>`
Complementary	`role="complementary"`	`<aside>`
Region	`role="region"`	`<section>`
Article	`role="article"`	`<article>`
Search	`role="search"`	N/A

Use HTML elements for landmarks rather than ARIA roles, as they have **built-in semantics and behaviour**.

An activity

Let's look at a page to see how all key landmarks work **with the accessibility tree**.

Some examples

Some example pages

- Contacts - Good example! - banner, 2x navigation (with labels), main
- Dashboard - Overuse of banner landmark - solvable with use of main