Georgia Tech Center for Inclusive Design and Innovation

The Basics of WAI-ARIA

Presenter:

- John Toles

Center for Inclusive Design & Innovation

Georgia Tech Center for Inclusive Design and Innovation

- Center for Inclusive Design & Innovation (CIDI) provides practical solutions for challenges faced daily by individuals with disabilities.
- We focus on solutions that offer utility, usability, and durability.
- CIDI offers services including disability compliance consultation, braille, captioning, audio description, accessibility evaluation of digital content, training and technical assistance.

NORTH
THE SEAL OF THE GEORGIA INSTITUTE OF TECHNOLOGY
1885
Georgia Institute of Technology
A Unit of the University System of Georgia

Find more information at cidi.gatech.edu (https://cidi.gatech.edu)

Presenter Bio

Georgia Tech Center for Inclusive Design and Innovation

I started with CIDI in 2016 providing support for a research study. Since then I have expanded my role to include development for the Student Accommodation Manager application and providing web accessibility evaluations for the Web Access team, as well as QuickBase and Drupal development for various other projects.

John Toles
Assistive Technology
Support Specialist

Basics of ARIA

Georgia Tech Center for Inclusive Design and Innovation

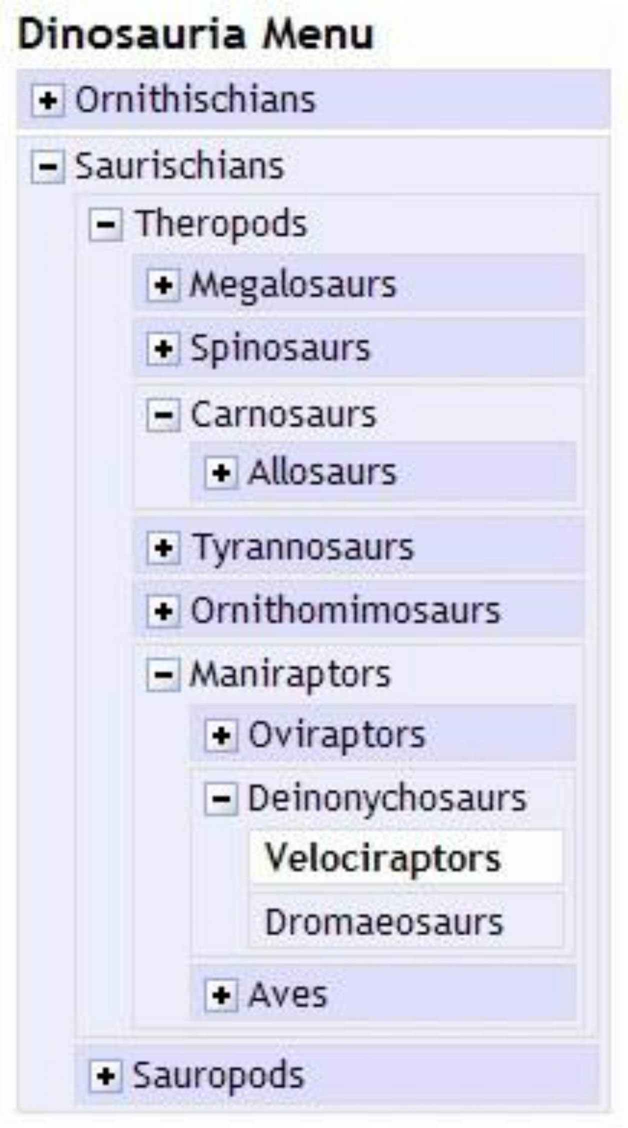
What is ARIA?

From The W3C (World Wide Web Consortium):

Web Accessibility Initiative's **A**ccessible **R**ich **I**nternet **A**pplications (**ARIA**) Suite, defines a way to make Web content and Web applications more accessible to people with disabilities. It especially helps with dynamic content and advanced user interface controls developed with Ajax, HTML, JavaScript, and related technologies.*

*w3.org/WAI/standards-guidelines/aria/

WAI-ARIA and ARIA are synonymous, but it's most often referred to simply as ARIA.



Basics of ARIA

Georgia Tech Center for Inclusive Design and Innovation

WAI-ARIA 1.1 Specifications

(https://www.w3.org/TR/wai-aria-1.1)

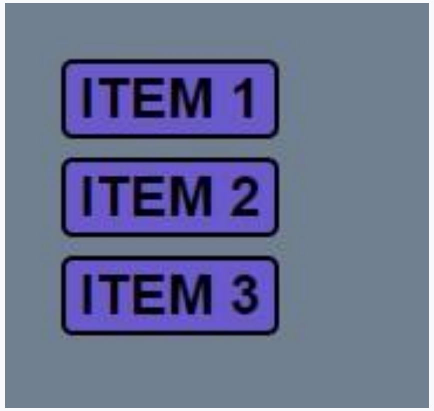
The ARIA specification consists of roles and states and properties.

- Roles provide information about the purpose of an element (“menu”, “treeitem”, “slider”, and “progressmeter”)
- States and properties indicate the condition of an element and whether or not it has changed ("checked" or "haspopup")

Put simply, roles describe what a component is or does. While state and properties describe how it can be interacted with.

Basics of ARIA

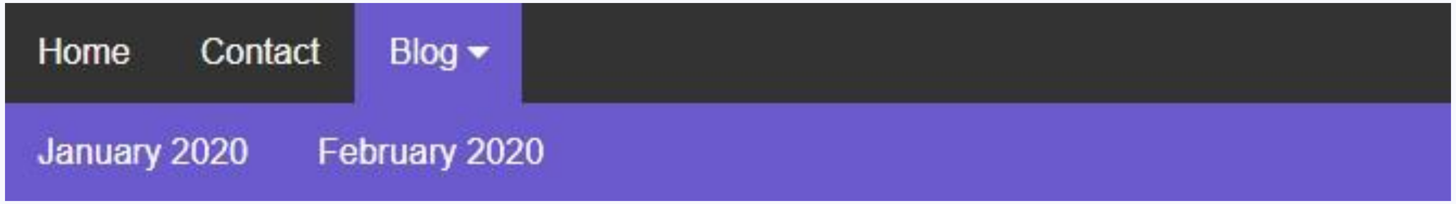
Roles Simple menu example:



```
<div role="menu"> <!--the role is on a <div> element-->

   <ul> <!--no role has been assigned to the <ul> tag-->

      <!--roles have been applied to the <a> tags in the list-->

      <li>< button role="menuitem" onclick=“function1()">Item 1</button ></li>

      <li>< button role="menuitem" onclick=“function2()">Item 2</button ></li>

      <li><button role="menuitem" onclick=“function3()">Item 3</button ></li>

   </ul>

</div>
```

Basics of ARIA

State and properties Simple navigation menu example:



```
<div role="navigation">
    <ul>
        <li><a href=“#”>Home</a></li>
        <li><a href=“uri”>Contact</a>
        <!--the <a> tag on holds the aria describing the popup menu-->
        <li><a aria-haspopup="true“ aria-expanded="false“ href=“#“>Blog</a>
        <ul role="menu" aria-label="Blog">
            <li role="none"> <!--notice roles are used to override semantic html-->
                <a role="menuitem" href="/blog-Jan2020.html“>January 2020</a></li>
            <li role="none">
                <a role="menuitem" href="/blog-Feb2020.html“ February 2020</a></li>
    </ul>
<div>
```

ARIA Roles Demonstration

Basics of ARIA

What does ARIA do?

WAI-ARIA addresses accessibility challenges by defining how information about component functionality to assistive technology. With WAI-ARIA, an advanced Web application can be made accessible and usable to people with disabilities. *w3.org/WAI/standards-guidelines/aria/

...and not do?

ARIA is not a coding method for making things accessible. It is a supplement to HTML. Developers use it to provide information about the purpose and function of web components not to provide additional features.

ARIA doesn't do anything on its own. It only tells assistive technology the intention behind a component; which is then conveyed to the user.

For ARIA to function properly it must be applied to components coded to meet HTML specification. Applying ARIA to bad code only makes things worse for the user.

Basics of ARIA

Georgia Tech Center for Inclusive Design and Innovation

When should you use ARIA?

ARIA is useful for identifying custom components and remediating older code.

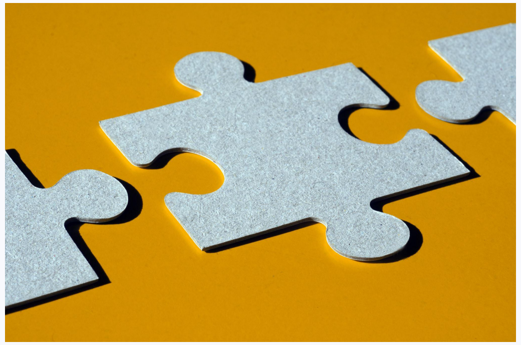
Before adding ARIA, however, developers should consider whether updating code would benefit a project before attempting to remediate old code. Developers should also test thoroughly before changing code to identify areas where it lacks accessibility and add ARIA to enhance those areas.

... and not use it?

In most cases developers do not need to add ARIA to their code. Modern screen readers and browsers work together well, and most tags are interpreted correctly. Developers should rely on semantic HTML wherever possible and implement custom components only when necessary.

Basics of ARIA

The Principles of ARIA?



The two essential principles of ARIA:

1. A role is a promise -

   When an ARIA role is used to identify a component, that component will behave the same as an equivalent HTML component.

2. ARIA can both block and enhance -

   When ARIA is applied correctly it can provide essential information about a component for assistive technology but applied incorrectly it can also cloak information about a component.

Unofficial third Principle:

No ARIA is better than Bad ARIA!

Basics of ARIA

How should you use ARIA?

Semantic markup is better for accessibility than including extensive ARIA. In most cases screen readers interpret HTML and identify components correctly.

The W3C maintains a catalog of tutorials for explaining how to implement ARIA in common HTML components.

WAI-ARIA Authoring Practices (https://www.w3.org/TR/wai-aria-practices-1.1/)

Basics of ARIA

Bad ARIA example:

```
<ul role="navigation">
  <li><a href="uri1">nav link 1</a></li>
  <li><a href="uri2">nav link 2</a></li>
</ul>
```

In implementing this component using a list but placing the role="navigation" attribute on the <ul> tag, the component will be identified as a navigation region not as a list. The ARIA is overriding the semantic HTML and the user is losing information.

Corrected example:

```
<nav> <!--notice this example has no ARIA-->
  <ul>
     <li><a href="uri1">nav link 1</a></li>
     <li><a href="uri2">nav link 2</a></li>
  </ul>
</nav>
```

Basics of ARIA

Georgia Tech Center for Inclusive Design and Innovation

ARIA Role and HTML Equivalents:

ARIA	HTML
role="navigation"	<nav>
role="main"	<main>
role="banner"	<header>
role="contentinfo"	<footer>
role="complementary"	<aside>
role="region"	<section>*
role="form"	<form>
role="search"	None*

*webpages often contain multiple <section> tags. An aria-label attribute should be applied to identify individual sections

*there is no HTML tag equivalent to role="search". A search bar is a good example of a custom component that can be enhanced with ARIA.

Basics of ARIA

Georgia Tech Center for Inclusive Design and Innovation

Final Takeaways:

1. ARIA can help you inform users about accessibility of your web components.
2. ARIA cannot make an inaccessible component accessible.
3. No ARIA is better than bad ARIA.

Resources:

4. W3C Web Accessibility Initiative (https://www.w3.org/WAI/)
5. WAI-ARIA Specification (https://www.w3.org/TR/wai-aria-1.1)
6. WAI-ARIA Authoring Practices (https://www.w3.org/TR/wai-aria-practices-1.1/)

Questions?