What is HTML5 ARIA?

Asked 8 years, 11 months ago Active 2 years ago Viewed 124k times

236

What is HTML5 ARIA? I do not understand how to implement it.

html5 wai-aria

54

edited May 2 '12 at 8:20

asked Aug 13 '10 at 5:18

Subbu

1,631 4 17 36

4 Note that WAI-ARIA predates HTML5 and does not require it, although the ARIA attributes will only be considered valid either by an HTML5 validator, or when compared with an ARIA extended DTD. However, the HTML5 draft currently disallows some WAI-ARIA constructs. – Alohci Aug 13 '10 at 11:42

I have seen this in Facebook's html. – Sorter Aug 7 '15 at 14:23

5 Answers

202

WAI-ARIA is a spec defining support for accessible web apps. It defines bunch of markup extensions (mostly as attributes on HTML5 elements), which can be used by the web app developer to provide additional information about the semantics of the various elements to assistive technologies like screen readers. Of course, for ARIA to work, the HTTP user agent that interprets the markup needs to support ARIA, but the spec is created in such a way, as to allow down-level user agents to ignore the ARIA-specific markup safely without affecting the web app's functionality.

Here's an example from the ARIA spec:

```
<ul role="menubar">

  <!-- Rule 2A: "File" label via aria-labelledby -->
  <li role="menuitem" aria-haspopup="true" aria-labelledby="fileLabel"><span
id="fileLabel">File</span>
    <ul role="menu">
```

```
      <!-- Rule 2C: "New" label via Namefrom:contents -->
      <li role="menuitem" aria-haspopup="false">New</li>
      <li role="menuitem" aria-haspopup="false">Open…</li>
      ...
    </ul>
  </li>
  ...
</ul>
```

Note the `role` attribute on the outer `<ul>` element. This attribute does not affect in any way how the markup is rendered on the screen by the browser; however, browsers that support ARIA will add OS-specific accessibility information to the rendered UI element, so that the screen reader can interpret it as a menu and read it aloud with enough context for the end-user to understand (for example, an explicit "menu" audio hint) and is able to interact with it (for example, voice navigation).

edited Dec 29 '16 at 11:11

Dan Atkinson
9,104 10 74 105

answered Aug 13 '10 at 6:08

Franci Penov
67.3k 14 119 156

10 What is the difference between data attribute and aria? – JackMahoney Sep 3 '12 at 11:11

50 `aria` is specifically for accessibility, and should not be used to store random data. `data` is for random data the application needs to associate with the node. – Franci Penov Sep 4 '12 at 16:28

2 Please keep in mind: `role="menu"` and `"menuitem"` are correct for APPS (= software with menus like `File > Open` or `Edit > Copy to clipboard` . For a classical website it's probably better to stay at the usual `<ul>` (= role list by default) as you are only providing links to other webpages and no functions like "save" or "copy/paste". If you are using `role="menu"` you also have to add support to navigate the menu with arrow keys of your keyboard as usual software/app menus – Oops D'oh Apr 12 '16 at 18:26

56

ARIA stands for Accessible Rich Internet Applications.

> WAI-ARIA is an incredibly powerful technology that allows developers to easily describe the purpose, state and other functionality of visually rich user interfaces - in a way that can be understood by Assistive Technology. WAI-ARIA has finally been integrated into the current working draft of the HTML 5 specification.

And if you are wondering what WAI-ARIA is, its the same thing.

> Please note the terms WAI-ARIA and ARIA refer to the same thing. However, it is more correct to use WAI-ARIA to acknowledge its origins in WAI.

WAI = Web Accessibility Initiative

From the looks of it, ARIA is used for assistive technologies and mostly screen reading.

Most of your doubts will be cleared if you read this article

http://www.w3.org/TR/aria-in-html/

edited Jun 6 '15 at 17:08

answered Aug 13 '10 at 5:56

Ranhiru Jude Cooray
14k 16 74 118

19

What is it?

WAI-ARIA stands for *"Web Accessibility Initiative – Accessible Rich Internet Applications"*. It is a set of attributes to help enhance the semantics of a web site or web application to help assistive technologies, such as screen readers for the blind, make sense of certain things that are not native to HTML. The information exposed can range from something as simple as telling a screen reader that activating a link or button just showed or hid more items, to widgets as complex as whole menu systems or hierarchical tree views.

This is achieved by applying roles and state attributes to HTML 4.01 or later markup that has no bearing on layout or browser functionality, but provides additional information for assistive technologies.

One corner stone of WAI-ARIA is the role attribute. It tells the browser to tell the assistive technology that the HTML element used is not actually what the element name suggests, but something else. While it originally is only a div element, this div element may be the container to a list of auto-complete items, in which case a role of "listbox" would be appropriate to use. Likewise, another div that is a child of that container div, and which contains a single option item, should then get a role of "option". Two divs, but through the roles, totally different meaning. The roles are modeled after commonly used desktop application counterparts.

An exception to this are document landmark roles, which don't change the actual meaning of the element in question, but provide information about this particular place in a document.

The second corner stone are WAI-ARIA states and properties. They define the state of certain native or WAI-ARIA elements such as if something is collapsed or expanded, a form element is required, something has a popup menu attached to it or the like. These are often dynamic and change their values throughout the lifecycle of a web application, and are usually manipulated via JavaScript.

What is it not?

WAI-ARIA is not intended to influence browser behavior. Unlike a real button element, for example, a div which you pour the role of "button" onto does not give you keyboard focusability, an automatic click handler when Space or Enter are being pressed on it, and other

properties that are indiginous to a button. The browser itself does not know that a div with role of “button” is a button, only its accessibility API portion does.

As a consequence, this means that you absolutely have to implement keyboard navigation, focusability and other behavioural patterns known from desktop applications yourself. You can find some Advanced ARIA techniques Here.

When should I not use it?

Yes, that’s correct, this section comes first! Because the first rule of using WAI-ARIA is: **Don’t use it unless you absolutely have to!** The less WAI-ARIA you have, and the more you can count on using native HTML widgets, the better! There are some more rules to follow, you can check them out here.

edited Jul 28 '17 at 14:00

answered Apr 2 '16 at 9:37

Faisal Naseer

2,032 16 36

12

What is ARIA?

ARIA emerged as a way to address the accessibility problem of using a markup language intended for documents, HTML, to build user interfaces (UI). HTML includes a great many features to deal with documents (P, h3,UL,TABLE) but only basic UI elements such as A, INPUT and BUTTON. Windows and other operating systems support APIs that allow (Assistive Technology) AT to access the functionality of UI controls. Internet Explorer and other browsers map the native HTML elements to the accessibility API, but the html controls are not as rich as the controls common on desktop operating systems, and are not enough for modern web applications Custom controls can extend html elements to provide the rich UI needed for modern web applications. Before ARIA, the browser had no way to expose this extra richness to the accessibility API or AT. The classic example of this issue is adding a click handler to an image. It creates what appears to be a clickable button to a mouse user, but is still just an image to a keyboard or AT user.

The solution was to create a set of attributes that allow developers to extend HTML with UI semantics. The ARIA term for a group of HTML elements that have custom functionality and use ARIA attributes to map these functions to accessibility APIs is a “Widget. ARIA also provides a means for authors to document the role of content itself, which in turn, allows AT to construct alternate navigation mechanisms for the content that are much easier to use than reading the full text or only iterating over a list of the links.

It is important to remember that in simple cases, it is much preferred to use native HTML controls and style them rather than using ARIA. That is don’t reinvent wheels, or checkboxes, if you don’t have to.

Fortunately, ARIA markup can be added to existing sites without changing the behavior for mainstream users. This greatly reduces the cost of modifying and testing the website or application.

edited Jul 1 '15 at 9:25

answered Apr 17 '15 at 14:06

Krishna
2,817 2 21 27

6

I ran some other question regarding ARIA. But it's content looks more promising for this question. would like to share them

What is ARIA?

If you put effort into making your website accessible to users with a variety of different browsing habits and physical disabilities, you'll likely recognize the role and aria-* attributes. WAI-ARIA (Accessible Rich Internet Applications) is a method of providing ways to define your dynamic web content and applications so that people with disabilities can identify and successfully interact with it. This is done through roles that define the structure of the document or application, or through aria-* attributes defining a widget-role, relationship, state, or property.

ARIA use is recommended in the specifications to make HTML5 applications more accessible. When using semantic HTML5 elements, you should set their corresponding role.

And see this you tube video for ARIA live.

answered Jan 30 '13 at 12:01

user10
3,453 5 33 59

Small correction: When using semantic HTML5 elements, you should *only* set their corresponding role if the the semantics of the element you have chosen do not provide enough information. In other words, if you have a nav element, there is no need to add role="navigation", because the role is already clear from the element choice. But if you have an input element for searching the site, you *should* add role="search" because input elements are used for many things other than search, and this marks up that specific input as having the search role. – brennanyoung Mar 13 '18 at 12:29

protected by Josh Crozier Jun 6 '16 at 19:25

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?