Accessibility of the “title” attribute

# Background

## Browser Compatibility

When mouse is placed over the link text containing “title” attribute a tooltip pops up displaying the value of “title” attribute.

* Firefox 18 and 17 and older versions (11) fully support the “title” attribute.
* Internet Explorer 9 and 8 fully support the “title” attribute.
* Chrome 24 also supports the “title” attribute.

## Cross reference of WCAG and other resources

* <http://www.w3.org/TR/WCAG20-TECHS/H33.html>

Internet Explorer will display the alt text on mouse-over if there is no title text. The Firefox and Opera browsers only display the title text on mouse-over and do not use the alt attribute text for this purpose.

Assistive technologies provide different levels of support for speaking “title” attributes. Some do not include features that allow users to access information provided via the title attribute.

Implementing this technique with the “title” attribute is only sufficient if the “title” attribute is accessibly supported.

* <http://webaim.org/articles/gonewild/#title>

The title attribute, by definition, can be used to provide **advisory** information.

The “title” attribute should be used on small links and navigation items in order to cover the underlying item in a title tooltip.

The title attribute is not read for most elements by default in most screen readers. Exceptions are the frame element and form elements that do not have a label.

## Solving the issue around the use of redundant or non-descriptive link text



The “title” attribute along with the <aria-label> property should be used to provide additional descriptive and advisory information for the link text. This is analogous to a person wearing a belt and suspenders for the same purpose.

# Best Practice Consideration

Various assistive technologies utilize the features of link text, “title” attribute and the <aria-label> property, in different ways.

Web browsers also have different approach method for the “title” attribute and the <aria-label> property. Some support the functionality of the <aria-label> to provide advisory information (invisible label) about the link text. However, still there are features of ARIA which are not supported in some popular web browsers.

**Therefore, a combination of “title” attribute along with the <aria-label> property should be used to provide additional descriptive and advisory information for the link texts.**

Although WAI-ARIA is already supported in several browsers and assistive technologies, it is not finalized yet. Once WAI-ARIA is stable, WAI will collect and publish a list of WAI-ARIA implementations, that is, what supports WAI-ARIA. Until that time a combination of methods will provide compliant access for users of websites.

More information regarding browsers support and accessibility issues around ARIA can be found in the following URL:

<https://developer.mozilla.org/en-US/docs/Accessibility/ARIA/Web_applications_and_ARIA_FAQ>

The purpose of <aria-label> is to provide the user with a recognizable name of the object. There are cases where providing a visible label is not the desired user experience. Most host languages provide an attribute that could be used to name the element (e.g. the title attributes in HTML), this may present a browser tooltip. In the cases where a visible label or visible tooltip is undesirable, developers may set the accessible name of the element using <aria-label>.

# User Interface

Users of Assistive Technologies react differently to the link text considering the following:

* link context – as it appears on the page
* Its “title” attribute
* Its <alt> attributes

Assistive technologies prioritize the information provided in the link text, “title” or <alt> attribute differently.

Depending on what type of Assistive Technology is used, information may or may not be provided from the list above. Assistive Technologies access the code (same link texts) differently and therefore present different kinds of advisory information to the end user.

Having the same link texts with no additional feature to provide an acceptable level of support for accessibility and information distinguishing the link text causes confusion or lack of information for Assistive Technology users. This is true, especially for visually impaired users who are not able to gather information about the target of the link by the surrounding context and images.

Therefore, users’ Assistive Technologies need additional coding methods to distinguish the link text information.

Note: Developers need to focus and consider methodologies in coding that provide for alternative methods of access. Just as coding methods are developed for mobile applications, there is a need for developing coding methods that allow for other types of devices, i.e. Assistive Technologies. This is accomplished by consultation on the best coding methodologies for Assistive Technologies and interoperability standards.

# Browsers compatibility with the best practice consideration

 ****

The overlapping section of the graph above represents that the combination of “title” and <aria-label> attributes, is a compatible feature for the popular browsers. It depicts that Internet Explorer versions 8, 9, Firefox 17, 18 and Chrome 24 provide appropriate access to redundant link text.