

# Accessibility Markup Review

Websites and web applications have the opportunity to provide additional information to persons with disabilities to help them understand and navigate more easily. Some of this information is visible to all users and can enhance their experience. Some of it is transparent to sighted users, but provides context for persons with visual, auditory or motor limitations.

In an accessibility markup (code) review, the focus of the inspection is not the interface as the user sees it, but an *under-the-covers* examination of the code used to implement it – often HTML.

The person conducting the review must be familiar with the implementation language and know what standards of quality and correctness apply. Code reviews are an important part of an accessibility assessment since there are a number of established conventions that must be followed.

## Under-the-covers Questions

- Is content separated from presentation through CSS?
  - Are buttons and other controls text-based?
  - Are tables used for data and not layout?
  - Does page content linearize?
- Do pages have frames?
  - Are they necessary? If so, do they have meaningful titles?
- Are headings hierarchical?
  - Is punctuation provided in headings to control screen readers?
- Are tables properly structured?
  - Do they contain headers and summaries?
  - Are table cells associated with headers?
  - Are they used for data and not presentation?
  - If tables are necessary for presentation, are they identified as such? E.g.: `role="wairole:presentation"`
- Are forms properly structured?
- Are forms organized by subject area?
- Are labels associated with input boxes?
- Are XHTML navigation landmarks taken from XHTML2? See: [The XHTML Role Attribute](#) (Don't use 'skip to main content' - these roles are much more comprehensive.)
- Has a navigation section been identified on each page (`role="navigation"`)?
- Do all navigation landmarks have a title including customer ARIA regions?
- Do Rich Internet Application widgets properly support ARIA and follow specified navigation rules?
- Do Rich Internet Application widgets follow the *tab to container and arrow* convention?
- Are keyboard mnemonic conventions consistent for context menus and a limited set of UI widgets?
- Are application and document areas in web pages properly defined to assist assistive technology with keyboard navigation? (A screen reader should not be in browse mode in a web application.)
- Does UI componentry provide rich hint, help, or tooltip text by establishing a relationship between the UI widget (could be a form control) and help information? For example, ARIA provides a `describedBy` and `Tooltip` relationship for these situations.
- Do all forms support extended ARIA form information, such as *required* and *invalid*?
- Are ARIA states and Properties synchronized with CSS style properties?
- Do UI components supporting multiple selections or options repeat option text?

## Accessibility Review Protocols

The Fluid UX Accessibility Working Group has created a set of protocols for assessing accessibility. Code review is one aspect of an accessibility review. If you'd like to do a deeper, more thorough evaluation, check out these additional resources. These include inspecting markup as well as inspecting the behaviour of interfaces.

- [Simple Accessibility Review Protocol](#):

These are a simple set of heuristics for evaluating the general accessibility of a web application without need for complex assistive technologies or a lot of "under the covers" assessment of the HTML markup and so on.

- [Comprehensive Accessibility Review Protocol for PC](#): This is a PC-based protocol for in-depth accessibility testing. It contains three accessibility test protocols for Fluid: Internet Explorer 7 with the AIS Toolbar Extension, the JAWS Screen Reader and the Web Accessibility Checker tool. A form for entering results and a completed form for reference are contained in attachments.
- [Comprehensive Accessibility Review Protocol for Macintosh](#): This is a Macintosh version of the PC-based protocol.

Another useful guide for determining whether a site addresses accessibility through properly crafted HTML is the [IBM Human Ability and Accessibility Center: Web Accessibility Checklist](#).