

Sakai Accessibility Working Group Accessibility Goals Development

 This page is a workspace for the Accessibility Working Group's Accessibility Goals Development efforts.

Draft Sakai Accessibility Goals

Prompted by Eli Cochran, the Accessibility Working Group has been working on accessibility goals for Sakai.

Since Sakai 2.x is obviously in production, is designed with different technical and user experience goals than Sakai 3, and mostly developed under earlier accessibility standards, we see the accessibility goals between Sakai 2.x and Sakai 3 as needing to be separate. These goals focus on technical accessibility standards and laws, and Sakai's functional accessibility, and not the process of achieving the goals, where the goals fit into Sakai's Development Process, or where they fit into any time-line or roadmap documents. The plan for meeting these goals, and the resulting process related issues need to be addressed in separate documents that discuss where accessibility fits into the Sakai Development Process. It is our intent that those documents will be written or discovered, and then linked to from the accessibility goals if/when they are adopted and posted.

Draft Sakai 2.x Accessibility Goals

- Well formed, semantically correct markup - valid where possible except where exceptions are necessary to make Sakai more accessible or functional
- Section 508 §1194.22 (a-p)
- WCAG 1.0 Priorities One and Two (except Checkpoints 3.2 and 6.3)
- Tested for functional accessibility to ensure the best user experience possible for persons using adaptive technology

Draft Sakai 3.x Accessibility Goals

- Well formed, semantically correct markup - valid where possible except where exceptions are necessary to make Sakai more accessible or functional
- WCAG 2.0 Level A and AA
- WAI-ARIA feature usage according to best practice guidelines
- Consistent and standard keyboard functionality following a style guide similar to the DHTML Style Guide Working Group's DHTML Style Guide
- Authoring Tool Accessibility Guidelines (ATAG)
- Tested for functional accessibility to ensure the best user experience possible for persons using adaptive technology

Reasoning Used in Determining the Above Draft Accessibility Goals

Reasoning for the Draft Sakai 2.x Accessibility Goals

These goals, while obviously not met 100% by Sakai 2.x, appear realistic to what could be achieved with Sakai 2.x and should be strived for with any new development.

Well formed, semantically correct markup - valid where possible except where exceptions are necessary to make Sakai more accessible or functional

The need for strictly valid markup was excluded to allow for technologies like ARIA which work, but are not included in current (X) HTML standards, and to allow for other non standard tags to be used where needed to improve accessibility or functionality (like the embed tag as needed for displaying video in some browsers, etc.).

Besides keeping coding/markup as valid as possible, well formed code is needed in any case where xml is concerned to help provide consistent results in the various parsers since recovery and error handling techniques vary.

Semantically correct markup is very important for adaptive technology and the disabled user. Adaptive technology usually ignores the CSS and other presentational information and uses the semantic/structural markup to make contextual sense of the information being marked up. If CSS is used to, say, simply tell the rendering engine to make a paragraph look like a heading (such as `<p class="heading">`), then the page structure will be lost for users of adaptive technology. Understanding of the page, navigational features, and the ability to target headings in custom user style sheets would all be lost.

Section 508 §1194.22 (a-p)

Goals "Section 508" and "WCAG 1.0 Priorities One and Two" are stated as goals in existing Sakai 2.x documentation (like the Accessibility Help Document). It is assuming here that they simply meant the Web Accessibility parts of Section 508, hence the "§1194.22 (a-p)" specification.

WCAG 1.0 Priorities One and Two (except Checkpoints 3.2 and 6.3)

WCAG 1.0 checkpoint 3.2 (valid markup) was excluded to allow for ARIA and non standard tags to be used where needed to improve accessibility or functionality (WCAG 2.0 favors accessibility over strict adherence to valid markup see WCAG 2.0 Checkpoint 4.1.1). Semantically correct markup is important to provide adaptive technology (and its users) proper contextual information and navigation features.

WCAG 1.0 Checkpoint 6.3 was left out because it requires pages to be usable with JavaScript turned off (this is not a requirement of Section 508 or WCAG 2.0 - the output and behavior of scripting is required to be accessible).

Tested for functional accessibility to ensure the best user experience possible for persons using adaptive technology

JavaScript on the client/browser end and coding on the server side, is used heavily in today's web applications to control the behavior of a web "page" and the controls found on it. Where the behavior of HTML controls used to be determined by the browser, the behavior of many HTML controls, including HTML elements not originally thought of as controls, is now in the hands of the programmer. Since JavaScript and CSS are used heavily in modern sites to make traditional HTML elements behave in non-traditional manners (such as slider controls, spinner controls, pop-up calendar date pickers, etc.), the standards don't (and can't) cover all the roles and behaviors a programmer can devise for HTML elements. The standards can't describe how every new and future creation should behave.

The only way to know it is truly usable by users of adaptive technology is to test it.

Reasoning for the Draft Sakai 3.x Accessibility Goals

Sakai 3 developers should be reaching for these goals.

Well formed, semantically correct markup - valid where possible except where exceptions are necessary to make Sakai more accessible or functional

WCAG 2.0 favors accessibility over strict adherence to valid markup see WCAG 2.0 Checkpoint 4.1.1. The need for strictly valid markup was excluded here, to allow for technologies like ARIA which work, but are not included in current (X)HTML standards, and to allow for other non standard tags to be used where needed to improve accessibility or functionality (like the embed tag as needed for displaying video in some browsers, etc.).

Besides keeping coding/markup as valid as possible, well formed code is needed in any case where xml is concerned to help provide consistent results in the various parsers since recovery and error handling techniques vary between browsers.

Semantically correct markup is very important for adaptive technology and the disabled user. Adaptive technology usually ignores the CSS and other presentational information and uses the semantic/structural markup to make contextual sense of the information being marked up. If CSS is used to, say, simply tell the rendering engine to make a paragraph look like a heading (such as <p class="heading">), then the page structure will be lost for users of adaptive technology. Understanding of the page, navigational features, and the ability to target headings in custom user style sheets would all be lost.

WCAG 2.0 Level A and AA

Many international laws either directly state requiring WCAG compliance or at least appear based on WCAG standards - this is especially true of WCAG 1.0. WCAG 2.0 is in many ways backwards compatible with WCAG 1.0 and offers features that make sense for Sakai 3 (Allows JavaScript and WAI-ARIA, covers many accessibility issues caused by using JavaScript to modify the behavior of standard HTML controls as found in today's web applications (and covers much of Section 508 §1194.21)).

WAI-ARIA

WAI-ARIA is included as a goal for its features which help make web applications accessible and usable, and as a technology to use in meeting WCAG 2.0 4.1.2, Section 508 §1194.21, and proposed new rules in Section 508 §1194.22 (not an exhaustive list).

Consistent and standard keyboard functionality following the DHTML Style Guide Working Group's DHTML Style Guide

JavaScript coding, and even server side programming, can affect much of the keyboard functionality and other behaviors of a web application such as Sakai 3. Accessible navigation and functional accessibility in general, benefit greatly by consistent and standard behavior from web applications. Conformance to the DHTML Style Guide Working Group's DHTML Style Guide (http://dev.aol.com/dhtml_style_guide) would work to achieve this. The DHTML Style Guide Working Group had a wide range of participants who contributed to the style guide (it wasn't just AOL). We haven't found a similar, or better, standard / guidelines document elsewhere.

Authoring Tool Accessibility Guidelines (ATAG) 1.0

Authoring Tool Accessibility Guidelines (ATAG) 1.0 are included to insure Sakai 3's content authoring features are accessible and that they allow users to create accessible content.

Tested for functional accessibility to ensure the best user experience possible for persons using adaptive technology

Especially in sites like Sakai 3, JavaScript on the client/browser end and coding on the server side, is used heavily to control the behavior of a web "page" and the controls found on it. Where the behavior of HTML controls used to be determined by the browser, the behavior of many HTML controls, including HTML elements not originally thought of as controls, is now in the hands of the programmer. Since JavaScript and CSS are used heavily in modern sites to make traditional HTML elements behave in non-traditional manners (such as slider controls, spinner controls, pop-up calendar date pickers, etc.), the standards don't (and can't) cover all the roles and behaviors a programmer can devise for HTML elements. The standards can't describe how every new and future creation should behave.