FREE eBook

LEARNING accessibility

Free unaffiliated eBook created from
Stack Overflow contributors.

#accessibilit

y

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: accessibility

It is an unofficial and free accessibility ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official accessibility.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with accessibility

Remarks

Understanding accessibility is a process of relating four main categories of abilities to software development. These broad categories are:

- visual
- hearing
- mobility
- cognitive

For each category, the needs of users needs to be considered. It must also be understood that every person has a range of abilities, and that range can very depending on things that include a person's age, their environment, and other activities they may be engaged in. Some examples of solutions to problems faced by people with abilities outside the 'normal' range include:

- Individuals who have visual impairments include abilities ranging from complete blindness to people who can't read small text without their glasses. Solutions to these problems might include things like text equivalents for images and keyboard only navigation. They may need to use high contrast color schemes or large fonts. They may not be sensitive to differences in color, requiring the software to communicate with the user using some other channel of information.

- Individuals who have hearing impairments need solutions such as closed captions and transcripts of spoken audio, or other visual means that communicate the message a sound is conveying.

- Individuals who have mobility impairments may need solutions such as voice controls or keyboard shortcuts. They almost certainly need software that does not impede or interfere with alternative access software running on their computer.

- Individuals who have cognitive impairments need solutions such as simplified terminology, example inputs, and unified page layouts.

As a software developer, there are practices that can either help or hinder the accessibility of the software you work on. For example, if you are working on desktop software, and you create custom GUI controls, then the tools that people with visual impairments use might not be able to interact with those controls unless you make extra effort to make those controls accessible. If you are developing web-based software, the structure and content of the pages can likewise either help or hinder people with disabilities that are using your site.

References

- Making a Web Form Accessible Pluralsight Course
- W3C's Getting Started Guide
- WebAim's Resources List

- NVDA - a free screen reader for Windows
- IBM Human Ability and Accessibility Center | Accessibility at IBM | Understanding accessibility
- IBM Accessibility Center | Developer and testers | IBM Accessibility Checklist for Web
- Voluntary Product Accessibility Template (VPAT) - NetBeans IDE 8.1

Examples

Installation and Setup

OSX

Implement the contract of the role-specific protocol (NSAccessibilityButton, NSAccessibilityImage, NSAccessibilityGroup, etc) within the NSAccessibility protocol that best matches the behavior of the GUI element being rendered.

Linux / BSD

For GNOME applications, the GNOME Accessibility Implementation Library (GAIL) bridges GNOME widgets and the Accessibility Toolkit (ATK). ATK bridges to the Assistive Technology Service Provider Interface (AT-SPI). AT-SPI is currently used by GTK2, Java and OpenOffice.

Windows

Microsoft Windows SDK includes all the tools necessary for MSAA and/or UI Automation. The IAccessibleEx interface the bridges between the two worlds.

References

- Windows Automation API SDK Tools – Microsoft Windows UI Automation Blog
- NSAccessibility - AppKit | Apple Developer Documentation
- Introducing ATK, AT-SPI, GAIL and GTK+

Accessibility Standards and APIs

Standards

- Accessibility/Laws - GNOME Wiki!
- Information and Communication Technology (ICT) Standards and Guidelines: Section 508
- Information and Communication Technology (ICT) Standards and Guidelines: Section 508 Refresh
- Accessible Rich Internet Applications (WAI-ARIA) 1.0 (W3C Recommendation, March 2014)
- Web Content Accessibility Guidelines (WCAG) 2.0 (W3C Recommendation, December

2008; also available as ISO/IEC 40500:2012)

- Guidance on Applying WCAG 2.0 to Non-Web Information and Communications Technologies (WCAG2ICT) (W3C Working Group Note, September 2013)
- User Agent Accessibility Guidelines (UAAG) 1.0 (W3C Recommendation, December 2002)
- User Agent Accessibility Guidelines (UAAG) 2.0 (W3C Working Group Note, December 2015)
- Authoring Tool Accessibility Guidelines (ATAG) 2.0 (W3C Recommendation, September 2015)

APIs

- ATK - Accessibility Toolkit
- IAccessible2
- Assistive Technology Service Provider Interface (AT-SPI)
- Accessibility Tools Framework | projects.eclipse.org
- Microsoft Active Accessibility (MSAA)
- Microsoft UI Automation (UIA)
- NSAccessibility - AppKit | Apple Developer Documentation

Read Getting started with accessibility online:
https://riptutorial.com/accessibility/topic/4677/getting-started-with-accessibility

Chapter 2: Developing for Screen Reader Users

Remarks

NVDA is a free screen reader for Windows, which you can use for testing.

Examples

Hiding non-interactive content from visible display, still read by screen readers

If you were to hide a link by setting `display: none` in the CSS then screen readers wouldn't find it.

Instead, we position it absolutely, with clipping.

CSS

```
.offscreen { position: absolute; clip: rect(1px 1px 1px 1px); /* for Internet Explorer */ clip: rect(1px, 1px, 1px, 1px); padding: 0; border: 0; height: 1px; width: 1px; overflow: hidden; }
```

HTML

```
<div class="offscreen">This text is hidden.</div>
```

Credit:

Steve Faulkner (Paciello Group): HTML5 Accessibility Chops: hidden and aria-hidden, 1 May 2012.

Notes by Ted Drake, on use of the off screen technique described:

> Using negative position can create long scroll bars when localizing a site for a rtl language. Also, it uses CSS properties that are commonly used and easy to accidentally over-ride.
>
> The Yahoo Accessibility Lab recommends using clip for content that should be hidden from the visual user, yet available to screen reader users. Thierry Koblentz has a great article on this technique, as well as the underlying philosophy behind using the correct CSS techniques for hiding content. Clip your hidden content for better accessibility

Read Developing for Screen Reader Users online:
https://riptutorial.com/accessibility/topic/4995/developing-for-screen-reader-users

Credits

S. No	Chapters	Contributors
1	Getting started with accessibility	Christophe Strobbe, Community, Fiona - myaccessible.website, Paul Sweatte, SteveDonie
2	Developing for Screen Reader Users	Caesar Wong, Christophe Strobbe, Fiona - myaccessible.website