Girl Develop It Web Accessibility Workshop

Instructor Guide

# Introduction

This is the instructor guide to the Girl Develop It Web Accessibility curriculum. It was originally developed by Sylvia Richardson, with many additions by Marcy Sutton. It is intended as a two-week basic course.

The slide show is designed for maximum presentability—each slide has one picture and one sentence. There are two reasons for this. This first is that the more words you have on a slide, the less people will listen to you. The second is that it reduces your temptation to just read from the slides. There are a few text-heavy slides in the slide deck, to make it more useful as a reference for students after the class. I skip those slides when I present, and I suggest you do too. Don’t worry, this guide has speaker notes.

This guide describes the class as I taught it, but the course is yours to modify to fit your needs, and the needs of your students. If you make some improvements, please fork the GitHub repository (<https://github.com/girldevelopit/girldevelopit-rdu-access>) or comment there.

Bonus note: The slide set contains a hidden navigation menu for screen reader users. Try it out!

# Checklist

## Before the class

### Planning

**Locate an appropriate venue**. To conduct the course as planned, you’ll need a room with space for everyone in the class. I recommend capping the class at about 15 students. This space will need:

* A desk/table and chair for each student
* A projector
* Speakers, for screen reader demos
* A whiteboard, or place to mount a large piece of paper
* Wifi
* Outlets for student laptops

**Set your time and date** and post the event. Begin basic promotional efforts. Here is a sample blurb:

Who can use your websites? Are they accessible to as many people as possible?

Web accessibility has traditionally described the process of enhancing websites for people with disabilities. Come to this class to learn more about developing with accessibility in mind—and building a better web experience for everyone.

This two-week class will focus on general design principles, with some hands-on exercises. It is helpful to have basic knowledge of HTML and CSS. You should also plan to bring a laptop and some headphones to work along with the class.

### One week before

**Start heavy-duty promotion**: tweet, ask other meetups to cross-post, and so on.

**Practice the presentation**. Run through the full presentation at least once, preferably in front of an audience.

**Visit the venue**, and make sure you know where everything is/how to set things up.

**Print the handouts**. Your printer will inevitably fail if you try to do it the day of.

**Gather your supplies**. You’ll need to find or buy index cards (for namecards), sharpies, the handouts, and a pen for each student. You’ll also need any cables or connectors to attach to the projector and speakers. Gather these things now, and put them in your laptop bag.

### One day before

**Send participants an introductory email**. Here is some sample text:

Hello everyone!

I look forward to seeing you at tomorrow's workshop; it will be a lot of fun. To participate in the activities, you'll need a laptop and a set of headphones. If you don't have access to a laptop, please let me know; we may be able to bring an extra for you. In class, we will be trying a screenreader simulation that requires the Adobe Shockwave plugin. To save yourself some time, you can go to <http://www.adobe.com/shockwave/welcome/> tonight and check to see if you have the plugin installed.

I also recommend you install the WAVE toolbar--it is a free Firefox and Chrome plugin you can use to do in-browser accessibility testing. Download it at <http://wave.webaim.org/toolbar/>.

I will be demonstrating the built-in Apple screenreader, VoiceOver. If you have an iDevice (iPod touch, iPhone, iPad), bring it along.

**Email/call your contact at the venue**, and confirm all your details (especially who will let you into the building!) Ask them for the wifi password.

**Download a local copy of the slides**, in case you have connection issues.

**Charge your laptop,** then put it in your laptop bag, along with all your supplies. Put this bag by the door/in your car.

### Setting Up

**Write on the board** or post clearly in room:

* Name, contact information
* Shortlink to screenreader demo: <http://bit.ly/p3JmJ2>
* Wifi ID and password
* Link to slides (Mine are at http://roenok.github.io/girldevelopit-rdu-access/)

**Have for each student**:

* Index card and marker, for making a name card
* Making Alt Text work handout

**Have for you:**

* Laptop
* Cables to connect to projector and speakers
* Presenter notes
* Optionally, an iDevice (iPhone, iPod, iPad) to demonstrate VoiceOver. You can find a good intro to VoiceOver at <http://examples.oreilly.com/9780596804299/voiceover-free-bonus-appendix.pdf>.

On your laptop, close all programs with popups and notifications. You don’t want your mom to Skype you in the middle of class. Open up a browser window with tabs for the slides and each demo. Mine were:

* The slides
* Screenreader demo (<http://webaim.org/simulations/screenreader-sim.htm>)
* Example of a site without a skipnav ([http://cnn.com](http://cnn.com/), at the time I taught the class)
* YouTube video of eye tracking software (<https://www.youtube.com/watch?v=p83wLuq3USA>)
* Example of a moving clickable element (<http://photojojo.com/store/awesomeness/iphone-boom-mic/>, video in slideshow with moving pause button)
* Example of popup signup (<http://codecanyon.net/item/subscribe-popup/full_screen_preview/2421232>)

### After class

Send a followup email to the class, with a link to the slides.

# Speaker Notes

## Class 1 Intro

### Welcome and Introductions

(3 slides)

Have everyone fill out a name card, and hang it on their laptop. Go around the room and do introductions—keep it moving.

### What is Accessibility?

(1 presentation slide, 1 notes slide)

Basic into to accessibility—about helping people use your site, interfacing with technology.

### Why Accessibility?

(2 pres, 1 notes)

Throw some numbers out. From the Census Bureau:

**54 million** people have a disability--**19 percent** of the civilian noninstitutionalized population. This increases with age.

* 5 percent of children 5 to 17 have disabilities.
* 10 percent of people 18 to 64 have disabilities.
* 38 percent of adults 65 and older have disabilities.

**1.8 million** people 15 and older are unable to see printed words.

**1 million** people 15 and older are unable to hear conversations.

**2.5 million** have difficulty having their speech understood.

**16.1 million** have limitations in cognitive functioning or who have a mental or emotional illness.

Talk about legal issues. Unclear if ADA applies to websites, regs for government agencies.

It’s a good thing to do! (to puppies)

### Benefits of Accessibility

(1 pres, 1 notes)

Ask if people recognize curb cut, then point out that while wheelchairs use them, so do strollers, people with luggage, etc. Mention how accessibility can help the non-disabled as well.

### Types of Disability

(1 pres, 1 notes)

**Ask the class** about types of disabilities, facilitate discussion. You are fishing for:

* Visual disabilities: blind or low-sight, color blind
* Hearing disabilities: deaf or hard-of-hearing
* Physical disabilities: MS, paraplegic/quadriplegic, epilepsy
* Cognitive disabilities: dyslexia, low literacy, learning disabilities

Point out how this can include non-traditional disabilities—low English proficiency, arthritis, etc.

### Accessibility is a continuum, not a checkbox

(1 pres)

There are checkers and standards, like section 508, but it really is a process.

## Visual disabilities

(1 pres, 1 activity slide)

Go over basics of how screenreaders work, then it is time for **Class Activity 1**.

### Working with screen readers

(1 pres, 2 notes)

Basics: screen readers read linearly, keep alt text short, describe function, not content, include a skipnav, test with text view or a reader, use headings.

**Demo:** Choose a site with a complex nav menu, and show how long it takes to get through with a screenreader. I used CNN.com and NVDA for this demo.

Then it is time for **Class Activity 2**.

### Other visual disabilities

(3 pres, 1 notes)

Use color carefully (color blindness), accommodate low-sight users who need good contrast, zoom. Very common among older users. **Share color contrast test tools.**

## Hearing disabilities

(1 pres, 1 notes)

Captions/transcripts, noise should not be the sole indicator of an event or function.

## Physical disabilities

(1 pres, 1 notes)

Not everyone can use a mouse! **Demo: show video on eye-tracking**, **movie with moving pause button, popup signup.**

Tiny buttons are bad!

Flickering elements can trigger seizures.

## Cognitive disabilities

(3 pres, 1 notes)

Cognitive and learning disabilities are common, but there is not too much research on usability. (Also includes LEP, low-English proficiency) Scanning is a high-level task; make primary content easy to find, provide extra time on timed elements, avoid captchas whenever possible.

## Conclusion

Finish up with **Class Activity 3**.

# Class 2 Introduction

To get everyone started, re-state GDI mission and the definition of accessibility.

## Common problems

Introduce the [WebAIM Screen Reader survey](http://webaim.org/projects/screenreadersurvey4/) and show the common problems. Marcy's version of the [infographic](http://marcysutton.com/stuff/GDI-Accessibility/img/webaim-survey-problematic-items.png) highlights topics covered in the two classes. This can be helpful to keep open in a second browser tab so it can be referenced more than once. Slide: <http://marcysutton.com/stuff/GDI-Accessibility/classslides-week2.html#/1/1>

### How do I code HTML for Accessibility?

Give overview of the HTML topics we'll cover in class: headings, form labels, tab index, and external link indicators.

## What are html headings and how should I use them?

Talk about HTML headings and the different levels (h1-h6). What are they for? How do screen readers use them? Resource: [WebAIM Semantic Structure](http://webaim.org/techniques/semanticstructure/)

## What about html5 and headings?

Talk about [screen reader testing](http://tink.co.uk/2013/02/screen-reader-support-for-html5-sections/) of HTML 5 tags & headings and the current state of the HTML5 heading algorithm (h1's that start over with new landmarks). Mention [bugs with JAWS and IE8](http://accessibleculture.org/articles/2011/10/jaws-ie-and-headings-in-html5/).

## Form labels – how do they work?

Talk about form labels. How do you link them to form inputs? **Activity 1: add a label to an HTML form**

## External links – what's the big deal?

A common problem cited in the WebAIM survey is “unexpected screen changes”. Discuss how to warn users of links that open in new windows with offscreen text and title attributes. Include a discussion about how it's really about usability in general. Recommended reading: “Don't Make Me Think” by Steve Krug

## Tab Index – What is it and how do I use it?

Explain [tab order](http://webaim.org/techniques/keyboard/tabindex). Show a demo of tabIndex=”” attributes on native and non-native elements to show how it affects the flow of a document (anchor, input, div, heading, etc.). Explain tab index values of -1, 0 and 1+ their specific purposes. **Activity 2: play with tab index on an HTML document**

### How do I code CSS for accessibility?

Give overview of how CSS can be used for accessibility. Topics covered: offscreen content, really hiding content from a screen reader & keyboard. Reference: [WebAIM Invisible Content Article](http://webaim.org/techniques/css/invisiblecontent/)

## Offscreen content: how does it work?

Discuss the various approaches for putting content offscreen with CSS: position and left properties, clip, text-indent, etc.

## Visibility( or invisibility): how to really hide content

Show CSS for hiding content from everyone: display: none and visibility: hidden. **Activity 3: play with CSS on an HTML document and see how it affects a keyboard and screen reader.**

### Resources and questions

Wrap up by answering questions and providing resources.

* TabIndex: <http://webaim.org/techniques/keyboard/tabindex>
* Headings & Semantic Structure: <http://webaim.org/techniques/semanticstructure/>
* HTML5, Headings & Screen Readers: <http://tink.co.uk/2013/02/screen-reader-support-for-html5-sections/>
* Form Labels: <http://webaim.org/techniques/forms/controls>
* Invisible Content with CSS: <http://webaim.org/techniques/css/invisiblecontent/>
* Responsive Images with CSS Backgrounds: <http://mobile.smashingmagazine.com/2013/07/22/simple-responsive-images-with-css-backgrounds/>

# Class 1 Activities

## Activity One: Use a Screenreader

Have students try the screenreader demo at <http://webaim.org/simulations/screenreader-sim.htm>. (With headphones!) You may have to help people install the Shockwave plugin. Time limit to ten minutes, and cut it short if students seem bored or frustrated.

If available, demo VoiceOver on the iPhone or iPad. Show students how they can set up the Accessibility Shortcut (triple tap home button to enable/disable VoiceOver, configurable from Settings > General > Accessibility).

## Activity Two: Making Alt Text Work

Pass out the alt text handout (the one with the kittens). Have students work in pairs or groups of three to decide on good alt text for each image. Lead a brief class discussion of what people picked and why. Keep it moving. There are no right answers—the goal is to get people thinking.

## Activity Three: Improving Existing Sites

Students can do this activity alone or in pairs, depending on the size of the class. Have each student/group visit a site they use a lot, and identify one accessibility feature that is already part of the site, and one thing they would change. If people get stuck, recommend:

* <http://www.buzzfeed.com/>
* <http://www.ebay.com/>
* [http://facebook.com](http://facebook.com/)

Depending on time, you can have each group present on what they found.

# Class 2 Activities

Demo files: <https://github.com/marcysutton/girldevelopit-rdu-access/blob/master/demo.zip>

## Activity One: add a label to a form

Students should experiment with adding a label to a form. Make sure the for attribute links the label to an input or other form element. Suggest adding labels to a group of radio buttons using the for=”” and name =”” attributes.

## Activity Two: experiment with tab index

Have students put tabIndex=”-1”, tabIndex=”0” and tabIndex=”1” (or higher numbers) to see how they affect the flow of an HTML document. Have a discussion about the purpose of each value.

## Activity THREE: play with offscreen & hidden text

Students should use CSS to put content offscreen and see how it affects the keyboard (and a screen reader, if available). They should also experiment with “really hiding” content to remove it from the flow of a document.