Experiences

• What accessibility feature did you try?
  • Who used a screen reader?
  • Who tried a high-contrast display?
  • Who tried high-magnification?

• Could you use the web/applications this way?
  • Why or why not?
What does Accessibility mean?

• Letting everyone access your web site
• What does everyone include?
  • Non-Native speakers
  • Persons with disabilities
  • Everyone
Who Are Your Users?

- Do you have a particular set of users in mind?
- Are all your potential users “normal”?
- Are all of you “normal”
  - 6-8% of males are color blind
  - 30-70% of CS students have wrist problems
  - 65-75% of people wear glasses
  - 17% of impairments are uncorrectable
    - 6% of the population
    - ~50% as people get older

- THERE IS NO NORMAL
Typical Disabilities

- **Vision Problems**
  - Blindness, low-vision, presbyopia, color blindness

- **Hearing problems**
  - Deafness, high-frequency loss

- **Movement problems**
  - Paraplegic, wrist problems, broken arm/hand, MS

- **Difficulty in reading**
  - Dyslexia, illiterate
Web Site Accessibility

Making a web site accessible involves:

A. Ensuring your web site can be used effectively and efficiently with assistive devices such as screen readers
B. Making sure that a variety of disabled users can use your web site directly
C. Adding special HTML elements to handle various disabilities
D. Having separate web sites for the blind and other disabled groups
E. Testing your web site with a broad range of disable users.
Why Should You Care

• This is only a small subset of potential users
  • You should have enough users without these
  • It can be a lot of work adapting your app to handle all potential users
  • Is it worth it?
• YES: It is the right thing to do
  • But lots of companies (esp. startups) don’t bother
• YES: Makes your application better
  • It tends to make you application better in any case
  • Many of the things you do for accessibility help the overall look and feel and usability of the interface
• YES: Required legally
Americans with Disabilities Act

• Requires that all businesses make *reasonable* accommodations for all handicaps

• It applies to web applications
  • Universities have been sued for switching to Google Apps
    • [http://www.pcmag.com/article2/0,2817,2382033,00.asp](http://www.pcmag.com/article2/0,2817,2382033,00.asp)
  • Companies have been sued as well

• Anything done for the government
  • Must meet ADA guidelines
So What Do You Do

• Make your site at least minimally ADA-compliant
  • Using *best efforts*
  • This will make your web site better as well

• Easiest to do this from the start
  • While designing the user interface
  • While designing the web application

• Much harder to retrofit later on
  • Might not be simple CSS changes
  • Might require a full redesign of your site
Assistive Technologies

• You can’t be expected to do everything for everyone
  • Accessibility doesn’t have to be built it
  • But it has to be available
  • You should know what technology people actually use

• Assistive Technologies
  • Screen readers
  • Screen magnifiers
  • Assisitive display settings
  • Alternative input devices
Making Web Sites Accessible

• General rules
  • HTML provides features that can enhance accessibility
    • Also feature that have the opposite effect
  • You should know what helps and what hinders
    • Use what helps
    • Avoid what hinders

• The bulk of the work has been done for you
  • W3C web accessibility content accessibility guidelines
W3C Web Accessibility Initiative

• Provides simple guidelines with priorities
  • Various levels: A, AA, AAA
  • If you meet the guidelines you’re doing best effort
• The guidelines contain a lot of common sense
  • Match common user interface guidelines
    • Make your web site better
  • Guidelines apply to web sites in general
    • Go beyond accessibility and address *usability*
Guideline Examples

• **Understandability guidelines**
  • Make text readable and understandable
  • Make content appear and operate in predictable ways
  • Help users avoid and correct mistakes

• **Robustness guidelines**
  • Maximize compatibility with current assistive tools
  • Maximize compatibility with future assistive tools

• **DESIGNERS:**
  • Read and understand these guidelines
  • Before designing a web site
Guideline Checking

• Several accessibility testers exist
  • http://wave.webaim.org
  • http://cynthiasays.com
  • http://achecker.ca/checker/index.php
  • http://fae.cita.uiuc.edu
  • http://colorfilter.wickline.org :: color blind views of your page

• Test your web site with real users
  • To ensure it is accessible
  • To ensure you can handle a wide range of users
Complexities

• What happens with a Front-End heavy application
  • DOM changes dynamically
  • Will the user be able to tell from a screen reader?
  • This would need to be tested extensively

• What about included documents
  • Word, PDF, PowerPoint
Internationalization

Internationalizing a web site does not involve

A. Having separate web sites for each major country.
B. Localizing all text strings using an appropriate tool
C. Eliminating icons that have text in them
D. Avoiding culture-specific symbols
E. Using library functions for formatting time, currency, etc.
Why Internationalization

• **Where are your users**
  • Will they always be there
  • Is your software portable

• **What are your customers’ backgrounds**
  • Is English their first language

• **Creating one or multiple web sites**
  • Maintaining one or multiple web sites
What is Internationalization

• Creating source information that is locale independent
  • *Locale*: set of features defining the user’s region
  • Facilitate customization through localization
  • Much more than simple translation

• Localization
  • Customize a web site for a particular locale
  • Not an attempt to be everything for everybody

• Internationalization is really *Localization*
What Changes with Localization

• What do you think changes?
  • How many know other languages
  • How many have traveled to foreign countries
Language Changes

• Translation and automatic translation
• Fonts and character sets
  • Unicode versus ASCII versus ...
• Text direction
• Lengths of text elements
Symbols and Design Elements
Symbols and Design Elements
Symbols and Design Elements
Currency

• Different currency signs and conventions
Numbers, Dates, and Times

• Numbers are represented differently
  • US/UK: 12,345.67, France: 12 345.67,
  • Germany: 12.345,67, Asia: 1.2345,67

• Dates
  • Wednesday is 02/01/2016 (US)
  • Wednesday is 01/02/2016 (Everywhere else)

• Times
  • The meeting is at 3:30pm (US)
  • The meeting is at 15:30 (Elsewhere)

• Percentage
  • Space before percent or not; Percent before or after number
Alphabetization Order

- Unicode order is not correct (even for English)
  - Uppercase / lowercase
  - Accents affect order
- Where do extra letters go
  - Some accented letters are actually other letters
- Special sort orders for languages
  - Spanish: a,b,c,ch,d,...,l,ll,m,...
Addresses

• Postal codes
  • US has 5 (or 9) digit zip codes
  • Other countries have different codes (different length, letters and digits)

• Telephone numbers
  • Can have more or fewer digits than US
  • Country codes

• Names
  • Salutations
  • Patronymic names
  • Where to display titles, degrees, etc.
Other Changes

• Units of measure
  • Metric versus English

• Paper sizes
  • Letter/Legal versus A3/A4
  • Printer layout

• Calendars can differ
  • Gregorian
  • Julian
  • Chinese
  • Hebrew
Internationalization / Localization

- Language
- Text Direction
- Fonts
- Height and width of labels
- Character set
- Sort order
- Meaning of symbols
- Meaning of colors
- Currency
- Purchase methods
- Number representations
- Units
- Date and time
- Calendars
- Abbreviations, mnemonics, ...
- Slang or jargon, idioms
- Addresses
- Telephone numbers
- Paper sizes (printing)
- Names
Internationalization

• How do you make your web site handle all these
  • How can you let your web site be customized to a particular locale (localization)
  • How to make it easy to add new locales

• Difficult to do after the fact
  • Error-prone to retrofit
  • You’ll miss something (icon/dynamic text/...)

• Fairly simple if done consistently from the start
  • Get in the habit of doing it right
Determining User Locale

- Browser can provide Accept-Language header
- Browser provides IP address
  - Can map IP address to country
  - GeoIP extension to PHP, Node.js
- Buttons to let user set common locales
Basic Techniques for Localization

• Separate structure from presentation (sound familiar?)
  • Replaceable icons and images
  • Replaceable fonts, colors, ...
  • Separate CSS sheets for different locales
• Use library/browser support for text
  • Different character sets
  • Different text directions
• Use localization libraries
  • Numbers, dates, currencies
  • Sorting
Avoiding Internalization Problems

- Avoid text embedded in graphics
  - Use text on top of a structured background
- Avoid culture-dependent symbols
  - Use envelops rather than mailboxes
  - Be wary of icons with cultural meanings (stars, crosses, ...)
  - Choose icons carefully or allow them to be localized
- Internationalize your database
  - Store Unicode, not simple ASCII
  - Use locale-specific sort order
Handling Static Text

• Externalize all text
  • Button names, navigation terms, error messages ...
  • There should be no text in the code or html
  • Front end and back end

• Use a “resource file”
  • Use a different resource file for different locales
  • Resource file access in the program
    • Can be a simple assignment of strings to variables
    • Can be a database, Can be an array with known indices
  • Use include facilities (JavaScript, PHP, Node.JS ...)

• Problems: truncation, spacing, ...
  • Use CSS style sheets to manage localization
Packages for Internationalization

- Date and Time formatting libraries
- Number formatting libraries
- General internationalization libraries
  - gettext (GNU Project)
    - Available in PHP, Django, Ruby, Java, Node.js ...
  - Uses a directory structure of resource files
  - I18n library for node.js

```c
/* Internationalized Jabberwocky program */
#include <libintl.h>
#include <locale.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]){
    ...
    setlocale(LC_ALL, "");
    bindtextdomain("jabberwocky", "/usr/share/locale");
    textdomain("jabberwocky");
    ...
    printf(gettext("Twas brillig, and the slithy toves\n"));
    printf(gettext("Old gire and gimbie in the wabe\n"));
    ...
    exit(0);
}
```
Handling Dynamic Text

• What are the problems?
  • “The $what is currently unavailable”
    • $what is one of “server”, “connection”, ...
  • “There are $n connections”
    • This is handled by gettext

• How might you handle this
  • Use a resource file as with static text
    • Store text in a database or an array of strings
    • Complete messages
  • Generate content through a predefined translation function
    • Libraries for this purpose exist
Next Time

• LAB on HTML/CSS
  • There is homework to be done in preparation for the lab
Symbols and Design Elements
Symbols and Design Elements

Symbols mean different things in different cultures, just like colors.
Other Items

• Abbreviations, mnemonics, acronyms
• Slang or jargon, idioms
  • Motherhood and apple pie
Screen Readers

• Change the visual display into audio output
  • Scan a window and read things in the order they appear
  • Some take HTML structure into
  • Useful for blind, dyslexic, illiterate

• Braille displays
  • Provide output as Braille rather than audio
  • Useful if both blind and deaf
  • Might be faster than audio alone

• Effects of these
  • Web page is reduced entirely to text
    • Simple images are meaningless
  • Browsing is a time-based experience
    • Navigation bar at top will be read for every page
Screen Magnifiers

• **Simple solutions**
  - Increasing the font size in the browser
    - Does this work?
    - Web site might not support this (fixed images, fonts)
  - Zoom the browser
  - Decreasing screen resolution of magnifying screen

• **Large scale magnification (400+%)**
  - Might cause loss of context
  - Might make the page difficult to use
    - Especially if there is a lot of blank space
Alternative Input Devices

• Simple alternatives
  • Sticky keys, slow keys

• Uses
  • Repetitive stress injuries are common
  • Blind/low vision cannot use the mouse
  • Some can’t use keyboard, but have a mouse equivalent
    • Severe arthritis, MS, ALS, ...
  • Keyboards are difficult to use on phone/tablet

• Navigate with only the keyboard
  • Keyboard as the mouse
  • Other devices simulate keyboard input
  • Tab sequences should be logical and valid
  • How do you follow links without a mouse
Guideline Categories

• **Perceivable guidelines**
  • Provide text-alternatives for non-text content
  • Provide captions and other alternatives for multimedia
  • Create content that can be presented in different ways
    • Usable by assistive technologies

• **Operable guidelines**
  • Make all functionality available from keyboard (mouse)
  • Give users enough time to read and use content
  • Do not use content that cause seizures
  • Help user navigate and find content
Checking Accessibility

• Use existing tools for a first approximation
• There is no substitute for using real people
gettext (I18N) Usage

• Take your source files
  • Replace all translatable output with gettext("...")
  • Can use _("...")
• Run xgettext on the file
  • This yields a file of all the messages in messages.po
• Create translated versions of those messages
  • Automatically or manually in another file
• Set up a hierarchy of messages.po files
  • Organized by locale name
• gettext in php/node will read from the right file