CS1320
Creating Modern Web and Mobile Applications

Lecture 3:
Universal Access
What does Universal Accessibility mean?

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

• Do you have a particular set of users in mind?
• Are you designing for yourself?
  o Most people do
• Do you represent all your users?
  o 6-8% of males are color blind
  o 30-70% of CS students have wrist problems
  o 65-75% of people wear glasses or contacts
  o 17% of impairments are uncorrectable
    ▪ 6% of the population
    ▪ ~50% as people get older

• **THERE IS NO TYPICAL USER**
Common Disabilities

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

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

• **Movement problems**
  o Paraplegic, wrist problems, broken arm/hand, MS, ALS

• **Difficulty in reading**
  o Dyslexia, illiterate, non-native speaker
Web Site Accessibility

Making a web site accessible primarily involves:

A. Ensuring your web site can be used effectively and efficiently with assistive devices such as screen readers
B. Making sure that 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 disabled users.
Why You Should Care

• **This only affects a small subset of potential users**
  - You should have enough users without these
  - It can be a lot of work adapting your app to 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: It isn’t really a small subset**
• **YES: Its not that hard**
• **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 all businesses make **reasonable** accommodations for all handicaps
- It applies to web and mobile applications
  - Universities have been sued for switching to Google Apps
  - Companies have been sued as well
- Anything done for the government (or Brown)
  - Must meet ADA guidelines
  - Might mean everything done in this course
So What Do You Do

• Make your site at least minimally ADA-compliant
  o This will make your web site better as well
• Easiest to do this from the start
  o While designing the user interface
  o While designing the web application
• Much harder to retrofit later on
  o Might not be simple CSS changes
  o Might require a full redesign of your site
• Need to understand what this means
  o Especially when we get to dynamic interfaces
Assistive Technologies

• You can't be expected to do everything for everyone (too many disabilities)
  o Accessibility doesn't have to be built-in
  o **But it has to be available**
  o You should know what technology people actually use

• Assistive Technologies
  o Screen readers
  o Screen magnifiers
  o Assistive display settings
  o Alternative input devices (keyboard only)
  o Video and audio captioning
Experiences

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

• **Could you use the web/applications this way?**
  o Why or why not?
How to Make Web Sites & Mobile Apps Accessible

• General rules
  o HTML provides features that can enhance accessibility
    ▪ Also features that have the opposite effect
  o You should know what helps and what hinders
    ▪ Use what helps, avoid what hinders

• The bulk of the work has been done for you
  o W3C web accessibility content accessibility guidelines
  o Understanding these makes accessibility easier
  o Particular HTML tags and fields to aid accessibility
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**
  - Guidelines apply to mobile applications as well
Guideline Examples

• **Understandability guidelines**
  - Make text readable and understandable (e.g. avoid idioms, funny fonts, …)
  - 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
  o http://wave.webaim.org
    ▪ Also available as a browser extension
  o http://cynthiasays.com
  o http://achecker.ca/checker/index.php
  o http://fae.cita.uiuc.edu
  o http://colorfilter.wickline.org :: color blind views of your page

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

• What happens with a Front-End heavy application (dynamic web pages)
  o DOM changes dynamically - Html effectively changes on the fly
  o Will the user be able to tell from a screen reader?
  o Will the user see the change if the page is highly magnified
  o This would need to be tested extensively
    ▪ Simple validators don’t necessarily help

• What about included documents
  o Word, PDF, PowerPoint, Excel

• We will keep coming back to accessibility
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 users’ backgrounds**
  - Is English their first language

- **Should you create one or multiple web sites**
  - How easy will it be to maintain
  - How many languages should you accommodate
What is Internationalization

• Creating source information that is locale independent
  o **Locale** : set of features defining the user’s region
    ▪ en-us, C, de, …
  o Facilitate customization through localization
  o Much more than simple translation

• Localization
  o Adapting a web site to a particular locale
  o Not an attempt to be everything for everybody at once

• Internationalization is really setting up for **Localization**
What Changes with Localization

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

- Translation and automatic translation
- Fonts and character sets
  - Unicode versus ASCII versus UTF-8 versus UTF-16 …
- Text direction, flow direction
- Lengths of text elements
Symbols and Design Elements
Symbols and Design Elements
Symbols and Design Elements
Currency

- Different currency signs and conventions

![Currency signs](image-url)
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**
  - Saturday is 02/01/2020 (US)
  - Saturday is 01/02/2020 (Elsewhere)

- **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)
  o Uppercase / lowercase
  o Accents can affect order

• Where do extra letters go
  o Some accented letters are actually other letters

• Special sort orders for languages
  o 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

- **Privacy Regulations**
- **Shopping (shipping, customs, ...)**
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
- Privacy requirements
- Shopping
Internationalization

- **How do you make your web site handle all these**
  - To let your web site be customized to a particular locale (localization)
  - 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/…) [or lots of somethings]
  - Easier to redo the entire web site from scratch

- **Fairly simple if done consistently from the start**
  - Get in the habit of doing it right
  - **DESIGNERS** – work on this from the start if you will need it
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?)
  o Replaceable icons and images
  o Replaceable fonts, colors, …
  o Separate CSS sheets for different locales

• Use library/browser support for text
  o Different character sets
  o Different text directions

• Use localization libraries and functions
  o Numbers, dates, currencies
  o Sorting
Avoiding Internalization Problems

• **Avoid text embedded in graphics**
  o Use text on top of a structured background
  o Good idea for accessibility as well

• **Avoid culture-dependent symbols**
  o Use envelops rather than mailboxes
  o Be wary of icons with cultural meanings (stars, crosses, …)
  o Choose icons carefully or allow them to be localized

• **Internationalize your database**
  o Store Unicode, not simple ASCII
  o Use locale-specific sort order (system sort functions)
Handling Static Text

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

• **Use a “resource file”**
  - Use a different resource file for different locales
    - Can be a directory hierarchy
  - 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**
  - Built into latest JavaScript

- **Number formatting & Sorting libraries**
  - Built into latest JavaScript

- **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("Did gyre and gimble in the wabe: \n"));
  ...
  exit(0);
}
```
Handling Dynamic Text

• **What are the problems?**
  o `The ${what} is currently unavailable`  
    ▪ $what is one of “server”, “connection”, …
  o `There are ${n} connections`  
    ▪ This can be handled bygettext

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

• **LAB on HTML/CSS**
  - There is (long) prelab to be done in preparation for the lab
  - Bring laptop to class if possible
  - Lab will be collaborative
Symbols and Design Elements
Symbols and Design Elements

- Symbols mean different things in different cultures. As can colors.

[Images of various symbols and icons related to accessibility and design elements]
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 you source files
  o Replace all translatable output with gettext(“…”)
  o Can use _(“…”)
• Run xgettext on the file
  o This yields a file of all the messages in messages.po
• Create translated versions of those messages
  o Automatically or manually in another file
• Set up a hierarchy of messages.po files
  o Organized by locale name
• gettext in php/node will read from the right file