The ACCESS Project

- Universal Design for Learning (UDL)
- Funded by U.S. Dept. of Education, Office of Postsecondary Ed.
- Winner of 2008 N. Preston Davis Group Award for Instructional Innovation
- Partnering with The Institute for Learning and Teaching (TILT), College of Natural Sciences, CVMBS

Defining UDL

"Universal Design for Learning is a set of principles and techniques for teaching, creating inclusive classroom instruction, technology, and accessible course materials."
Creating Accessible Documents

- Accessibility starts with the “native editor”:
  - Word processor
  - Spreadsheet
  - Desktop Publishing software
- Accessibility features transfer to other formats:
  - Adobe PDF
  - HTML
  - Daisy Talking Books

Content, Structure, and Presentation

- Content
  - The actual information you are providing in a document. This can include Text, Images, Videos, or Multimedia.
- Structure
  - The organization of content is structure. This includes headings, lists, tables, emphasis, etc.
- Appearance
  - One can add style rules to structural elements to give documents a particular appearance.

HTML: Content, Structure, and Appearance

- Create more flexible web pages with multiple presentations views:
  - Standard
  - Print
  - Small device
  - RSS
HTML: Structure

- **HTML tags define structure**
  - Headings: `<h1>` to `<h6>`
  - Paragraphs: `<p>`
  - Divisions: `<div>`
  - Lists: `<ul>`, `<ol>`, `<dl>`, and `<li>`
  - Images: `<img>`
  - Table cells and headers: `<td>` and `<th>`

- Example in Dreamweaver...

HTML: Appearance

- **Cascading Style Sheets (CSS)** control the appearance, position and behavior of elements:
  - `<h1>` { color: red; }
  - `#logo` { position: absolute; left: 0px; top: 0px; }
  - `#menu li a:hover` { background-color: red; }

HTML: Images and Alt Text

- The “alt” attribute of the `<img>` tag should capture the meaning of an image:
  ```html
  <img src="icecream.jpg" alt="The girls celebrated Julie's birthday over large bowls of ice cream.">
  ```

- Purely decorative images receive the “null” or “empty” alt attribute. Text-to-speech software will skip such images.
  Example: `<img src="bullet.gif" alt="">`
HTML: Tables

- Use tables for tabular data, not as a framework for page layout.
- Keep tables as simple as possible. Avoid nesting one table inside another.
- In header row, replace `<td>` (table data) with `<th>` (table header) tags.

```
<table>
<thead>
<tr>
<th>Costs</th>
<th>May</th>
<th>June</th>
<th>July</th>
</tr>
</thead>
<tbody>
<tr>
<td>$12</td>
<td>$20</td>
<td>$25</td>
<td></td>
</tr>
<tr>
<td>Sales</td>
<td>$15</td>
<td>$22</td>
<td>$25</td>
</tr>
<tr>
<td>Profit</td>
<td>$3</td>
<td>$2</td>
<td>$0</td>
</tr>
</tbody>
</table>
```

Scope attribute of `<th>` tag:
- Indicates one logical row and/or column of headers.

Caption:
- The `<caption>` tag serves as a label and is bound to the table data.

HTML: Simple Tables

```
<table>
<thead>
<tr>
<th>Costs</th>
<th>May</th>
<th>June</th>
<th>July</th>
</tr>
</thead>
<tbody>
<tr>
<td>$12</td>
<td>$20</td>
<td>$25</td>
<td></td>
</tr>
<tr>
<td>Sales</td>
<td>$15</td>
<td>$22</td>
<td>$25</td>
</tr>
<tr>
<td>Profit</td>
<td>$3</td>
<td>$2</td>
<td>$0</td>
</tr>
</tbody>
</table>
```

HTML: Complex Tables

```
<table>
<thead>
<tr>
<th>Costs</th>
<th>May</th>
<th>June</th>
<th>July</th>
</tr>
</thead>
<tbody>
<tr>
<td>$12</td>
<td>$20</td>
<td>$27</td>
<td>$28</td>
</tr>
<tr>
<td>Sales</td>
<td>$15</td>
<td>$22</td>
<td>$25</td>
</tr>
<tr>
<td>Profit</td>
<td>$3</td>
<td>$2</td>
<td>$0</td>
</tr>
</tbody>
</table>
```

- Complex Tables: two or more logical rows, and/or columns of headers.
- Must use 'id' and 'headers' attribute.
- Complex tables may require the “summary” attribute.
### HTML: Forms

- Group form elements using `<fieldset>` and `<legend>` tags.
- Add labels to form elements using the `for` attribute of the `<label>` tag. This attribute keeps labels and form elements linked, regardless of where they appear on the page.

```html
<fieldset>
    <legend>Credit Card Selection</legend>
    <input type="radio" name="credit" id="visa">
    <label for="visa">Visa</label>
    <input type="radio" name="credit" id="mc">
    <label for="mc">Master Card</label>
    <input type="radio" name="credit" id="amex">
    <label for="amex">American Express</label>
</fieldset>
```

### Example: fieldset, legend, ‘for’ attribute

```html
<fieldset>
    <legend>Credit Card Selection</legend>
    <input type="radio" name="credit" id="visa">
    <label for="visa">Visa</label>
    <input type="radio" name="credit" id="mc">
    <label for="mc">Master Card</label>
    <input type="radio" name="credit" id="amex">
    <label for="amex">American Express</label>
</fieldset>
```
HTML: Traversing a Form

- Make sure forms can be navigated from the keyboard using the Tab key.
- Avoid JavaScript behaviors that move the focus (cursor position) without notifying the user.

HTML: Hyperlinks

- Hyperlinks should describe where they will take the user.
- Links should make sense when read out of context.
- Include destination URL if page is likely to be printed.
  - Examples:
    - Poor: Click here to visit the Colorado State University home page.
    - Better: Visit the Colorado State University home page (http://www.colostate.edu/).
- Allow users to skip over repetitive navigation links and go straight to the main content.

ACCESS Project Website
http://accessproject.colostate.edu/udl/modules/

- Colorblind testing tools for your website
- Usability studies
  - effective layouts and fonts
  - eye tracking studies
  - effects of line length on reading ability
  - and more
- Validate HTML and CSS code
- And much more…
Thank you!

The ACCESS Project
accessproject@colostate.edu