UNIVERSITY OF CRETE
FACULTY OF SCIENCES AND ENGINEERING
COMPUTER SCIENCE DEPARTMENT

COURSE CS-564 (OPTIONAL)

ADVANCED TOPICS

IN HUMAN – COMPUTER INTERACTION

Course Convenor: Constantine Stephanidis

Web Accessibility

Introduction to Web Accessibility

What is Web Accessibility?

- Web accessibility means that people with disabilities can use the Web
  - More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web
- Currently most Web sites and Web software have accessibility barriers that make it difficult or impossible for many people with disabilities to use the Web
- Web accessibility also benefits others, including older people with changing abilities due to aging
- Web accessibility also benefits people without disabilities
  - For example, a key principle of Web accessibility is designing Web sites and software that are flexible to meet different user needs, preferences, and situations
  - This flexibility also benefits people without disabilities in certain situations, such as people using a slow Internet connection, people with "temporary disabilities" such as a broken arm, and people with changing abilities due to aging

Why is Web Accessibility Important?

- The Web is an increasingly important resource in many aspects of life: education, employment, government, commerce, health care, recreation, and more
- It is essential that the Web be accessible in order to provide equal access and equal opportunity to people with disabilities
- An accessible Web can also help people with disabilities more actively participate in society
- The Web offers the possibility of unprecedented access to information and interaction for many people with disabilities. That is, the accessibility barriers to print, audio, and visual media can be much more easily overcome through Web technologies
- Another important consideration for organizations is that Web accessibility is required by laws and policies in some cases

Common Accessibility Barriers

- Images without alternative text
- Misleading use of structural elements on pages
- Uncaptioned audio or undescribed video
- Lack of alternative information for users who cannot access frames, applets or scripts
- Tables that are difficult to decipher when linearized
- Sites with poor color contrast

Web Accessibility is a cross-disability issue

- Examples of design requirements for people with different kinds of disabilities include:
  - Visual
    - described graphics or video
    - well marked-up tables or frames
    - keyboard support, screen reader compatibility
  - Hearing
    - captioning for audio, supplemental illustration
  - Physical, Speech
    - keyboard or single-switch support
    - alternatives for speech input on voice portals
  - Cognitive, Neurological
    - consistent navigation, appropriate language level
    - illustration; no flickering or strobing designs

Accessibility Contributes to Universal Access (1/2)

- Accessible Web design contributes to better design for other users:
  - Multi-modality (support for visual, auditory, tactile access) benefits users of:
    - mobile phones with small display screens
    - Web-TV
    - kiosks
  - Multi-modality increases usability of Web sites in different situations:
    - low bandwidth (images are slow to download)
    - noisy environments (difficult to hear the audio)
    - screen-glare (difficult to see the screen)
    - driving (eyes and hands are "busy")

Accessibility Contributes to Universal Access (2/2)

- Redundant text/audio/video can support:
  - different learning styles
  - low literacy levels
  - second-language access
- Style sheets can support:
  - more efficient page transmission and site maintenance
- Captioning of audio files supports:
  - better machine indexing of content
  - faster searching of content

Essential Components of Web Accessiblity

How the components relate

The Implementation Cycle

- It is essential that several different components of Web development and interaction work together in order for the Web to be accessible to people with disabilities. These components include:
  - **content** - the information in a Web page or Web application, including:
    - natural information such as text, images, and sounds
    - code or markup that defines structure, presentation, etc.
  - **Web browsers, media players**, and other "user agents“
  - **assistive technology**, in some cases - screen readers, alternative keyboards, switches, scanning software, etc.
  - **users'** knowledge, experiences, and in some cases, adaptive strategies using the Web
  - **developers** - designers, coders, authors, etc., including developers with disabilities and users who contribute content
  - **authoring tools** - software that creates Web sites
  - **evaluation tools** - Web accessibility evaluation tools, HTML validators, CSS validators, etc.

World Wide Web Consortium

- The World Wide Web Consortium (W3C) is an international community where member organizations, a full-time staff, and the public work together to develop Web standards
- Led by Web inventor Tim Berners-Lee and CEO Jeffrey Jaffe, W3C's mission is to lead the Web to its full potential
- Principles
  - Web for All: One of W3C's primary goals is to make the benefits of web available to all people, whatever their hardware, software, network infrastructure, native language, culture, geographical location, or physical or mental ability
  - Web on Everything: The number of different kinds of devices that can access the Web has grown immensely. Mobile phones, smart phones, personal digital assistants, interactive television systems, voice response systems, kiosks and even certain domestic appliances can all access the Web

Web Accessibility Initiative

- The Web Accessibility Initiative (WAI) works with organizations around the world to develop strategies, guidelines, and resources to help make the Web accessible to people with disabilities
- In more detail, pursues accessibility of the Web through five primary activities:
  - ensuring that core technologies of the Web support accessibility
  - developing guidelines for Web content, user agents, and authoring tools
  - facilitating development of evaluation and repair tools for accessibility
  - conducting education and outreach
  - coordinating with research and development that can affect future accessibility of the Web

Web Content Accessibility Guidelines

- The Web Content Accessibility Guidelines (WCAG) documents explain how to make Web content accessible to people with disabilities
- Web "content" generally refers to the information in a Web page or Web application, including text, images, forms, sounds, and such
- WCAG Versions
  - Web Content Accessibility Guidelines 1.0 was published in May 1999
  - WCAG 2.0 was published on 11 December 2008. WCAG 2.0 applies broadly to more advanced technologies
  - WCAG 2.0 has 12 guidelines that are organized under four principles: perceivable, operable, understandable, and robust
  - For each guideline, there are testable success criteria, which are at three levels: A, AA, and AAA

WCAG 2.0 Conformance (1/2)

- There are five requirements that must be met in order for content to be classified as 'conforming' to WCAG 2.0
  1. **Conformance Level**: One of the following levels of conformance is met in full.
     - Level A: For Level A conformance (the minimum level of conformance), the Web page satisfies all the Level A Success Criteria, or a conforming alternate version is provided
     - Level AA: For Level AA conformance, the Web page satisfies all the Level A and Level AA Success Criteria, or a Level AA conforming alternate version is provided.
     - Level AAA: For Level AAA conformance, the Web page satisfies all the Level A, Level AA and Level AAA Success Criteria, or a Level AAA conforming alternate version is provided
  2. **Full pages**: Conformance (and conformance level) is for full Web page(s) only, and cannot be achieved if part of a Web page is excluded
  3. **Complete processes**: When a Web page is one of a series of Web pages presenting a process (i.e., a sequence of steps that need to be completed in order to accomplish an activity), all Web pages in the process conform at the specified level or better. (Conformance is not possible at a particular level if any page in the process does not conform at that level or better.)

WCAG 2.0 Conformance (2/2)

4. **Only Accessibility-Supported Ways of Using Technologies**: Only accessibility-supported ways of using technologies are relied upon to satisfy the success criteria. Any information or functionality that is provided in a way that is not accessibility supported is also available in a way that is accessibility supported
5. **Non-Interference**: If technologies are used in a way that is not accessibility supported, or if they are used in a non-conforming way, then they do not block the ability of users to access the rest of the page. In addition, the Web page as a whole continues to meet the conformance requirements under each of the following conditions:

   1. when any technology that is not relied upon is turned on in a user agent,
   2. when any technology that is not relied upon is turned off in a user agent, and
   3. when any technology that is not relied upon is not supported by a user agent

   In addition, the following success criteria apply to all content on the page, including content that is not otherwise relied upon to meet conformance, because failure to meet them could interfere with any use of the page: *Audio Control*, *No Keyboard Trap*, *Three Flashes or Below Threshold*, and *Pause, Stop, Hide*.

WCAG 2.0

WCAG 2.0 Principles

- The 4 principles of Web accessibility in WCAG 2.0 are:
  - Perceivable - Information and user interface components must be presentable to users in ways they can perceive. This means that users must be able to perceive the information being presented; it can't be invisible to all of their senses.
  - Operable - User interface components and navigation must be operable by users. This means that users must be able to operate the interface; the interface can't require interaction that the user can not perform.
  - Understandable - Information and the operation of the user interface must be understandable by users. This means that users must be able to understand the information as well as the operation of the user interface; the content or operation cannot be beyond their understanding.
  - Robust - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies. This means that users must be able to access the content as technologies advance; as technologies and user agents evolve, the content should remain accessible.

WCAG 2.0 Guidelines (1/2)

1. Perceivable
   - 1.1 Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language
   - 1.2 Provide alternatives for time-based media
   - 1.3 Create content that can be presented in different ways (for example simpler layout) without losing information or structure
   - 1.4 Make it easier for users to see and hear content including separating foreground from background
2. Operable
   - 2.1 Make all functionality available from a keyboard
   - 2.2 Provide users enough time to read and use content
   - 2.3 Do not design content in a way that is known to cause seizures
   - 2.4 Provide ways to help users navigate, find content, and determine where they are
3. Understandable
   - 3.1 Make text content readable and understandable
   - 3.2 Make Web pages appear and operate in predictable ways
   - 3.3 Help users avoid and correct mistakes
4. Robust
   - 4.1 Maximize compatibility with current and future user agents, including assistive technologies

WCAG 2.0 Guidelines (2/2)

- **Principles** - At the top are four principles that provide the foundation for Web accessibility: *perceivable, operable, understandable, and robust*
- **Guidelines** - Under the principles are guidelines. The 12 guidelines provide the basic goals that authors should work toward in order to make content more accessible to users with different disabilities. The guidelines are not testable, but provide the framework and overall objectives to help authors understand the success criteria and better implement the techniques
- **Success Criteria** - For each guideline, testable success criteria are provided to allow WCAG 2.0 to be used where requirements and conformance testing are necessary. In order to meet the needs of different groups and different situations, three levels of conformance are defined: A (lowest), AA, and AAA (highest)
  - In the context of this course only A and AA success criteria are being discussed
- **Sufficient and Advisory Techniques** - For each of the *guidelines* and *success criteria* in the WCAG 2.0 document itself, a wide variety of *techniques have been documented*. The techniques are informative and fall into two categories: those that are *sufficient* for meeting the success criteria and those that are *advisory*

1. Perceivable

GUIDELINES:

1.1 Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

1.2 Provide alternatives for time-based media.

1.3 Create content that can be presented in different ways (for example simpler layout) without losing information or structure.

1.4 Make it easier for users to see and hear content including separating foreground from background.

1.1 Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language

- (SC 1.1.1) Non-text Content: All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below (Level A)
  - **Controls, Input**: If non-text content is a control or accepts user input, then it has a name that describes its purpose
  - **Time-Based Media**: If non-text content is time-based media, then text alternatives at least provide descriptive identification of the non-text content
  - **Test**: If non-text content is a test or exercise that would be invalid if presented in text, then text alternatives at least provide descriptive identification of the non-text content
  - **Sensory**: If non-text content is primarily intended to create a specific sensory experience, then text alternatives at least provide descriptive identification of the non-text content
  - **CAPTCHA**: If the purpose of non-text content is to confirm that content is being accessed by a person rather than a computer, then text alternatives that identify and describe the purpose of the non-text content are provided, and alternative forms of CAPTCHA using output modes for different types of sensory perception are provided to accommodate different disabilities
  - **Decoration, Formatting, Invisible**: If non-text content is pure decoration, is used only for visual formatting, or is not presented to users, then it is implemented in a way that it can be ignored by assistive technology

SC 1.1.1 Examples

- **A data chart:** A bar chart compares how many widgets were sold in June, July, and August. The short label says, "Figure one - Sales in June, July and August." The longer description identifies the type of chart, provides a high-level summary of the data, trends and implications comparable to those available from the chart. Where possible and practical, the actual data is provided in a table.
- **An audio recording of a speech** : The link to an audio clip says, "Chairman's speech to the assembly." A link to a text transcript is provided immediately after the link to the audio clip.
- **An animation that illustrates how a car engine works**: An animation shows how a car engine works. There is no audio and the animation is part of a tutorial that describes how an engine works. Since the text of the tutorial already provides a full explanation, the image is an alternative for text and the text alternative includes only a brief description of the animation and refers to the tutorial text for more information.
- **A traffic Web camera:** A Web site allows users to select from a variety of Web cameras positioned throughout a major city. After a camera is selected, the image updates every two minutes. A short text alternative identifies the Web camera as "traffic Web camera." The site also provides a table of travel times for each of the routes covered by the Web cameras. The table is also updated every two minutes.
- **A photograph of an historic event in a news story**: A photograph of two world leaders shaking hands accompanies a news story about an international summit meeting. The text alternative says, "President X of Country X shakes hands with Prime Minister Y of country Y."
- **An e-learning application:** An e-learning application uses sound effects to indicate whether or not the answers are correct. The chime sound indicates that the answer is correct and the beep sound indicates that the answer is incorrect. A text description is also included so that people who can't hear or understand the sound understand whether the answer is correct or incorrect.

Sufficient Techniques for SC 1.1.1 (1/10)

- Situation A: If a short description can serve the same purpose and present the same information as the non-text content
  - Provide short text alternative for non-text content that serves the same purpose and presents the same information as the non-text content *(G94)*

In deciding what text to include in the alternative, it is a often a good idea to consider the following questions:

- Why is this non-text content here?
- What information is it presenting?
- What purpose does it fulfill?
- If I could not use the non-text content, what words would I use to convey the same function and/or information?

The alternative text should be "search" and not "magnifying glass"

The alternative text should describe how to tie the knot and not what the picture looks like

For an animation that shows how to change a tire, the short text alternative should describe what the animation is about, while the long text alternative should describe how to change a tire.

Sufficient Techniques for SC 1.1.1 (2/10)

- Situation B (1/2)
- If a short description can not serve the same purpose and present the same information as the non-text content (e.g., a chart or diagram):
  1. Provide short text alternatives that provide a brief description of the non-text content *(G95)* AND
  2. One of the following for long description:
     a) Provide long description for non-text content that serves the same purpose and presents the same information using a long text alternative *(G92)*
     b) Provide a long description in text near the non-text content, with a reference to the location of the long description in the short description *(G74)*
     c) Provide a long description in another location with a link to it that is immediately adjacent to the non-text content *(G73)*
- *Example: Consider a chart showing the top ten retail vehicle sales in the US market, by comparing two months: October 2007 and October 2008*

	Ford F - Series PU	Chevrolet Silverado PU	Toyota Camry / Solara	Toyota Corolla / Matrix	Chevrolet Impala	Honda Accord	Honda Civic	Nissan Altima	Dodge Ram PU	Honda CR-V
Oct-08	43,324	31,689	30,556	27,386	22,107	19,783	18,582	17,753	17,626	13,169
Oct-07	51,741	48,716	33,728	25,815	20,791	30,936	23,809	21,778	24,711	16,780

Sufficient Techniques for SC 1.1.1 (3/10)

- Situation B (2/2)
  1. The short text should be “Comparison of US Market Top 10 Retail Vehicle Sales: October 2008 versus October 2007”. The long description would read “Bar chart showing the comparison of US Top 10 Retail Vehicle Sales between October 2008 and October 2007. There are 10 car brands compared. The Ford F Series PU has marked the highest sales across both years, with the sales of 2007 being higher at 50.000 …”
     - <**img src**="chart.gif" **alt**=“Comparison of US Market Top 10 Retail Vehicle Sales: October 2008 versus October 2007" **longdesc**="Bar char showing the comparison of US Top 10 Retail Vehicle Sales between October 2008 and October 2007. There are 10 car brands compared…"/>
  2. The short text alternative says: " Comparison of US Market Top 10 Retail Vehicle Sales: October 2008 versus October 2007. Details in text following the chart:”. The following is in the paragraph immediately below the chart. Bar chart showing the comparison of US Top 10 Retail Vehicle Sales between October 2008 and October 2007. There are 10 car brands compared. The Ford F Series PU has marked the highest sales across both years, with the sales of 2007 being higher at 50.000 …"
     - **<img src**="chart.gif" **alt**=“Comparison of US Market Top 10 Retail Vehicle Sales: October 2008 versus October 2007. Details in text following the chart:”**/> **Bar char showing the comparison of US Top 10 Retail Vehicle Sales between October 2008 and October 2007. There are 10 car brands compared…** **
  3. The short text alternative says " Comparison of US Market Top 10 Retail Vehicle Sales: October 2008 versus October 2007”. Immediately after the non-text content is a small image denoting a long description. The alternate text for the image is "Long description of chart". The image links to the bottom of the page where there is a section titled "Description of charts on this page". The link points to this specific description: “Bar chart showing the comparison of US Top 10 Retail Vehicle Sales between October 2008 and October 2007. There are 10 car brands compared. The Ford F Series PU has marked the highest sales across both years, with the sales of 2007 being higher at 50.000 …[end of description]"

Sufficient Techniques for SC 1.1.1 (4/10)

- Situation C (1/3)
- If non-text content is a control or accepts user input:
  1. Provide a text alternative that identifies the purpose of the non-text content *(G82)*
     - Sometimes, a text alternative cannot serve the same purpose as the original non-text content (for example an applet meant to develop two dimensional rapid targeting skills and eye hand coordination.) In these cases this technique is used. With this technique a description of the purpose of the non-text content is provided.
     - An eye-hand coordination development applet has the following text alternative "Applet that uses the mouse and moving targets to develop eye-hand coordination"
  2. Use label elements to associate text labels with form controls (HTML) *(H44)*
     - The HTML and XHTML specifications allow both implicit and explicit labels. However, some assistive technologies do not correctly handle implicit labels
       - for example, <label>First name <input type="text" name="firstname"></label>
     - The objective of this technique is to use the label element to explicitly associate a form control with a label
  3. Use the title attribute to identify form controls when the label element cannot be used (HTML) *(H65)*
     - The objective of this technique is to use the title attribute to label form controls when the visual design cannot accommodate the label
     - For example, if there is no text on the screen that can be identified as a label, or where it might be confusing to display a label. User agents, including assistive technology, can speak the title attribute.

Sufficient Techniques for SC 1.1.1 (5/10)

- Situation C (2/3)
  - H44: Use label elements to associate text labels with form controls
    - A **label** is attached to a specific form control through the use of the **for** attribute. The value of the **for** attribute must be the same as the value of the **id** attribute of the form control
      - The **id** attribute may have the same value as the **name** attribute, but both must be provided, and the **id** must be unique in the Web page
    - Elements that use explicitly associated labels are (input type): "text", "checkbox", "radio", "file", "password", textarea, select
      - NOTE: the **label** is positioned after **input** elements of **type="checkbox"** and **type="radio"**
    - The label element is not used for the following because labels for these elements are provided via the value attribute (for Submit and Reset buttons), the alt attribute (for image buttons), or element content itself (button): Submit and Reset buttons (input type="submit" or input type="reset"), Image buttons (input type="image"), Hidden input fields (input type="hidden"), Script buttons (button elements or <input type="button">)

Example 1: A text input field

```
<label for="firstname">First name:</label>
<input type="text" name="firstname" id="firstname" />
```

Example 2: A checkbox

```
<input type="checkbox" id="markuplang" name="computerskills" checked="checked">
<label for="markuplang">HTML</label>
```

Sufficient Techniques for SC 1.1.1 (6/10)

- Situation C (3/3)
  - H65: Use the title attribute to identify form controls when the label element cannot be used
    - Example 1: A search form uses a pulldown menu to limit the scope of the search. The pulldown menu is immediately adjacent to the text field used to enter the search term. The relationship between the search field and the pulldown menu is clear to users who can see the visual design, which does not have room for a visible label. The title attribute is used to identify the select menu. The title attribute can be spoken by screen readers or displayed as a tool tip for people using screen magnifiers.

```
<label for="searchTerm">Search for:</label>
<input id="searchTerm" type="text" size="30" value="" name="searchTerm">
 <select title="Search in" id="scope">
…
</select>
```

    - Example 2: A Web page contains controls for entering a phone number in the United States, with three fields for area code, exchange, and last four digits

```
<fieldset>
        <legend>Phone number</legend>
        <input id="areaCode" name="areaCode" title="Area Code" type="text" size="3" value="" >
        <input id="exchange" name="exchange" title="First three digits of phone number" type="text" size="3" value="" >
        <input id="lastDigits" name="lastDigits" title="Last four digits of phone number" type="text" size="4" value="" >
 </fieldset>
```

Sufficient Techniques for SC 1.1.1 (7/10)

- Situation D: If non-text content is time-based media (including live video-only and live audio-only); a test or exercise that would be invalid if presented in text; or primarily intended to create a specific sensory experience:
  1. Provide a descriptive label using a short text alternative technique
  2. Provide a descriptive label that describes the purpose of live audio-only and live video-only content using a short text alternative technique *(G68)*
     - The purpose of this technique is to ensure that the user can determine what the non-text content is - even if they cannot access it
     - *Example: A live audio feed of the Mississippi House of Representatives has the following descriptive label "Live audio from the microphones in the Mississippi House of Representatives"*
  3. Provide the accepted name or a descriptive name of the non-text content using a short text alternative technique (G100)
     - The objective of this technique is to allow users to identify the non-text content even if the non-text content is intended to provide a specific sensory experience
     - For example, a deaf person may want to know what an audio instrumental file is - even if they cannot hear it
     - Similarly, a blind person may want to know what the subject of a visual image is - even if they cannot see it
     - *Example: A painting of the Mona Lisa has an alternate text of "Mona Lisa, by Leonardo da Vinci"*

Sufficient Techniques for SC 1.1.1 (8/10)

- Situation E: If non-text content is a CAPTCHA:
  1. Provide a text alternative that describes the purpose of the CAPTCHA *(G143)*
     - Such tests often involve asking the user to type in text that is presented in an obscured image or audio file. From the text alternative, the user can tell that the CAPTCHA requires completing a task and what type of task it is
     - When an alternate version of a CAPTCHA is available, the text alternative should include instructions about how to find the alternate version
     - *Example: A CAPTCHA test asks the user to type in text that is played in an audio file. The text alternative is "Type the letters spoken in the audio"*
  2. **AND** Ensure that the Web Page contains another CAPTCHA serving the same purpose using a different modality *(G144)*
     - The purpose of this technique is to reduce occasions in which a user with a disability cannot complete a CAPTCHA task
     - *Example: A blog comment form includes a visual CAPTCHA that must be completed before a user can submit comments. In addition to the visual CAPTCHA, it includes a CAPTCHA with a form field that asks, "What is two plus seven?" with a text entry field that allows users to enter the correct answer*

Sufficient Techniques for SC 1.1.1 (9/10)

- Situation F (1/2)
- If the non-text content should be ignored by assistive technology: Implement or mark the non-text content so that it will be ignored by assistive technology by:
  - Using null alt text and no title attribute on img elements for images that AT should ignore (HTML) *(H67)*
    - If no title attribute is used, and the alt text is set to null (i.e. alt="") it indicates to assistive technology that the image can be safely ignored.
    - Note: Although alt=" " is also valid, alt="" is recommended.
  - Using CSS to include decorative images (CSS) *(C9)*
    - Provides a mechanism to add purely decorative images and images used for visual formatting to Web content without requiring additional markup within the content. This makes it possible for assistive technologies to ignore the non-text content. Some user agents can ignore or turn off CSS at the user's request, so that background images included with CSS simply "disappear" and do not interfere with display settings such as enlarged fonts or high contrast settings.
    - Background images can be included with the following CSS properties:
      - background,
      - background-image,
      - content, combined with the :before and :after pseudo-elements,
      - list-style-image

Sufficient Techniques for SC 1.1.1 (10/10)

- Situation F (2/2)
  - H67: Use null alt text and no title attribute on img elements for images that AT should ignore
    - Example 1: The following image is used to insert a decorative image on a Web page

```
<img src="squiggle.gif" width="20" height="20" alt="" />
```

  - C9: Use CSS to include decorative images
    - Example 1: Background image for an HTML page

```
…
  <style type="text/css">
    body { background: #ffe url('/images/home-bg.jpg') repeat; }
  </style>
</head>
<body>
...
```

    - Example 2: Background image with CSS for image rollovers (creates a decorative rollover effect when the user hovers the mouse pointer over a link)

```
a:hover { background: #ffe url('/images/hover.gif') repeat; color: #000;
  text-decoration: none;
}
```

Short Text Alternative Techniques (1/5)

- Use alt attributes on images used as submit buttons (H36)
  - For input elements of type 'image', the alt attribute of the input element is used to provide a functional label
  - This label indicates the button's function, but does not attempt to describe the image
    - The label is especially important if there are multiple submit buttons on the page that each lead to different results
  - The input element is used to create many kinds of form controls. Although the HTML and XHTML DTDs permits the alt attribute on all of these, it should be used only on image submit buttons
    - User agent support for this attribute on other types of form controls is not well defined, and other mechanisms are used to label these controls
  - Example 1: An input element with an alt attribute

```
<form action="http://example.com/prog/text-read" method="post">
  <input type="image" name="submit" src="button.gif" alt="Submit" />
</form>
```

Short Text Alternative Techniques (2/5)

- Combine adjacent image and text links for the same resource (H2)
  - This objective of this technique is to avoid unnecessary duplication that occurs when adjacent text and iconic versions of a link are contained in a document
  - Many kinds of links have both a text and iconic link adjacent to each other
    - Visually they appear to be the same link, but they are experienced by many people as two identical links and this can be confusing
    - The preferred method to address this is to put the text and image together in one link, and provide null alternative text on the image to eliminate duplication of text
  - Example: A link contains text and an icon, and the icon provides additional information about the target

Woodend Music Festival Program

```
<a href="WMFP.pdf">
Woodend Music Festival Program
<img src="pdficon.gif" alt="PDF format"/>
</a>>
```

Short Text Alternative Techniques (3/5)

- Use alt attributes on img elements (H37)
  - When an image contains words that are important to understanding the content, the alt text should include those words
  - This will allow the alt text to play the same function on the page as the image
  - Note that it does not necessarily describe the visual characteristics of the image itself but must convey the same meaning as the image
- Provide text alternatives on applet elements (H35)
  - Provide a text alternative for an applet by using the alt attribute to label an applet and providing the text alternative in the body of the applet element
  - In this technique, both mechanisms are required due to the varying support of the alt attribute and applet body text by user agents
  - Example: An applet to play the tic-tac-toe gane

```
<applet code="tictactoe.class" width="250" height="250" alt="tic-tac-toe game">
  tic-tac-toe game
</applet>
```

Short Text Alternative Techniques (4/5)

- Use the body of the object element (H53)
  - The objective of this technique is to provide a text alternative for content rendered using the object element
  - The body of the object element can be used to provide a complete text alternative for the object or may contain additional non-text content with text alternatives
  - Example 1: An object includes a long description that describes it

```
<object classid="http://www.example.com/analogclock.py">
  <p>Here is some text that describes the object and its operation.</p>
</object>
```

  - Example 2: An object includes non-text content with a text alternative

```
<object classid="http://www.example.com/animatedlogo.py">
  <img src="staticlogo.gif" alt="Company Name" />
</object>
```

Short Text Alternative Techniques (5/5)

- Use a text alternative on one item within a group of images that describes all items in the group (G196)
- The objective of this technique is to avoid unnecessary duplication that occurs when a grouping of adjacent non-text content is used to present information or functionality
- Example: A rating system
  - In the following example, a rating is shown as three filled stars and two empty stars. While a text alternative could have been provided for each of the five images, the author has instead provided the rating in the form "3 out of 5 stars" for the first image and has marked the others using null alt text

Rating:

```
<p>Rating:
  <img src="star1.gif" alt="3 out of 5 stars">
  <img src=" star1.gif" alt="">
  <img src=" star1.gif" alt="">
  <img src="star2.gif" alt="">
  <img src="star2.gif" alt="">
 </p>
```

1.2 Provide alternatives for time-based media

Audio-only and Video-only (Pre-recorded)	Captions (Pre-recorded)	Audio Description or Media Alternative (Pre-recorded)
• For prerecorded audio-only and prerecorded video-only media an alternative for time-based media is provided or an audio track is provided that presents equivalent information for prerecorded video-only content	• Captions are provided for all prerecorded audio content in synchronized media, except when the media is a media alternative for text and is clearly labeled as such	• An alternative for time-based media or audio description of the prerecorded video content is provided for synchronized media, except when the media is a media alternative for text and is clearly labeled as such

SC 1.2.1 Audio-only and Video-only (Prerecorded)

- For prerecorded audio-only and prerecorded video-only media, the following are true, except when the audio or video is a media alternative for text and is clearly labeled as such: (Level A)
  - Prerecorded Audio-only: An alternative for time-based media is provided that presents equivalent information for prerecorded audio-only content
  - Prerecorded Video-only: Either an alternative for time-based media or an audio track is provided that presents equivalent information for prerecorded video-only content
  - Benefits:
    - It helps people who have difficulty perceiving visual content
    - Assistive technology can read text alternatives aloud, present them visually, or convert them to braille
    - Alternatives for timed-based media that are text based may help some people who have difficulty understanding the meaning of prerecorded video content
    - People who are deaf, are hard of hearing, or who are having trouble understanding audio information for any reason can read the text presentation

Example: An audio recording of a press conference

A Web page includes a link to an audio recording of a press conference that identifies the audio recording. The page also links to a text transcript of the press conference. The transcript includes a verbatim record of everything the speakers say. It identifies who is speaking as well as noting other significant sounds that are part of the recording, such as applause, laughter, questions from the audience, and so on

SC 1.2.2 Captions (Prerecorded)

- Captions are provided for all prerecorded audio content in synchronized media, except when the media is a media alternative for text and is clearly labeled as such (Level A)
  - Captions provide the part of the content available via the audio track
    - They not only include dialogue, but identify who is speaking and include non-speech information conveyed through sound, including meaningful sound effects
    - Captions are not needed when the synchronized media is, itself, an alternate presentation of information that is also presented via text on the Web page
  - Example: A captioned tutorial
    - A video clip shows how to tie a knot. The captions read,
    - "(music)
    - Using rope to tie knots was an important skill
    - for the likes of sailors, soldiers and woodsmen.."
  - Note: Captions can be open or closed
    - Open Captioning is encoded as part of a movie. Open Captioning is very similar to subtitles on foreign films and they cannot be turned off
    - The term "closed" in closed captioning indicates that not all viewers see the captions—only those who choose to decode or activate them. This distinguishes from "open captions" (sometimes called "burned-in" or "hardcoded" captions), which are visible to all viewers

SC 1.2.3 Audio Description or Media Alternative (Prerecorded)

- An alternative for time-based media or audio description of the prerecorded video content is provided for synchronized media, except when the media is a media alternative for text and is clearly labeled as such (Level A)
  - One approach is to provide audio description of the video content
    - The audio description augments the audio portion of the presentation with the information needed when the video portion is not available
    - During existing pauses in dialogue, audio description provides information about actions, characters, scene changes, and on-screen text that are important and are not described or spoken in the main sound track
  - The second approach involves providing all of the information in the synchronized media (both visual and auditory) in text form
    - The alternative for time-based media reads something like a screenplay or book
    - Unlike audio description, the description of the video portion is not constrained to just the pauses in the existing dialogue. Full descriptions are provided of all visual information, including visual context, actions and expressions of actors, and any other visual material. In addition, non-speech sounds (laughter, off-screen voices, etc.) are described, and transcripts of all dialogue are included
    - If there is any interaction as part of the synchronized media presentation (e.g., "press now to answer the question") then the alternative for time-based media would provide hyperlinks or whatever is needed to provide the same functionality

1.3 Create content that can be presented in different ways (for example simpler layout) without losing information or structure

Info and Relationships	Meaningful Sequence	Sensory Characteristics
• Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text	• When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined	• Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, size, visual location, orientation, or sound

SC 1.3.1 Info and Relationships (1/2)

- Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text (Level A)
  - The intent of this SC is to ensure that information and relationships that are implied by visual or auditory formatting are preserved when the presentation format changes
    - For example, the presentation format changes when the content is read by a screen reader or when a user style sheet is substituted for the style sheet provided by the author
  - Sighted users perceive structure through various visual cues
    - For example: headings are often in a larger, bold font separated from paragraphs by blank lines; list items are preceded by a bullet and perhaps indented; paragraphs are separated by a blank line; etc.
  - Auditory cues may be used as well
    - For example, a chime might indicate the beginning of a new section; a change in voice pitch or speech rate may be used to emphasize important information or to indicate quoted text; etc.
  - Example: A form that uses color and text to indicate required fields
  - A form contains both required and optional fields. Instructions at the top of the form explain that required fields are labeled with red text and also with an icon whose text alternative says, "Required." Both the red text and the icon are programmatically associated with the appropriate form fields so that assistive technology users can determine the required fields

SC 1.3.1 Info and Relationships (2/2)

- Example techniques
  - G115: Use semantic elements to mark up structure AND H49: Use semantic markup to mark emphasized or special text (HTML)
  - G140: Separate information and structure from presentation to enable different presentations
  - H39: Using caption elements to associate data table captions with data tables (HTML)
  - H73: Using the summary attribute of the table element to give an overview of data tables (HTML)
  - H43: Using id and headers attributes to associate data cells with header cells in data tables (HTML)
  - H44: Using label elements to associate text labels with form controls (HTML)
    - See previous examples
  - H65: Using the title attribute to identify form controls when the label element cannot be used (HTML)
    - See previous examples s
  - H71: Providing a description for groups of form controls using fieldset and legend elements (HTML)
  - H85: Using OPTGROUP to group OPTION elements inside a SELECT (HTML)
  - H48: Using ol, ul and dl for lists or groups of links (HTML)
  - H42: Using h1-h6 to identify headings (HTML)

G115: Use semantic elements to mark up structure (1/2)

- The objective of this technique is to mark up the structure of the Web content using the appropriate semantic elements
- **In other words, the elements are used according to their meaning, not because of the way they appear visually**
- The nature of a piece of content as a paragraph, header, emphasized text, table, etc. can all be indicated in this way
- In some cases, the relationships between units of content should also be indicated, such as between headings and subheadings, or amongst the cells of a table
- The user agent can then make the structure perceivable to the user, for example using a different visual presentation for different types of structures or by using a different voice or pitch in an auditory presentation

G115: Use semantic elements to mark up structure (2/2)

- This example shows how to use the em and strong elements to emphasize text

How to start the engine

Before starting the engine, **make sure the gear is in neutral**. Next, turn the key in the ignition. The engine should start.

```
<h1>How to start the engine</h1>
<p>
Before starting the engine, <strong>make
sure the gear is in neutral</strong>. Next, turn
the key in the ignition.  The engine should
start.
</p>
```

- Using highlighting and background color to visually and semantically identify important information

.....

New vocabulary words are emphasized and highlighted with a cyan background

The scathing review of the play seemed a bit too harsh. ....

```
<style type="text/css">
.vocab {
background-color:cyan;
font-style:normal;
}
</style>
.......
<p>New vocabulary words are emphasized and
highlighted  with a cyan background</p>
<p>The <em class="vocab">scathing </em>
review of the play seemed a bit too harsh. .... </p>
```

H49: Use semantic markup to mark emphasized or special text (1/2)

- The objective of this technique is to demonstrate how semantic markup can be used to mark emphasized or special text so that it can be programmatically determined
- Using semantic markup to mark emphasized or special text also provides structure to the document
- Assistive Technology Support Notes
  - Some semantic elements are not supported well by assistive technologies. Elements and attributes that are known to have limited support include code, del, dfn, ins, kbd, s, sub, sup, tt, and q
    - For these elements, authors are encouraged to consider whether they are using them in a way that would require users to be able to access the semantic meaning of the markup in order to understand the content and, where understanding the semantics is required, to provide this information in text
  - Most screen readers do not provide automatic notification about em, strong, b, or i
  - Firefox 1.0 (Windows) and higher, Opera 7.54 (Windows) and higher, Mozilla 1.7.3 (Windows) and higher automatically generate quotes around q elements, but Internet Explorer 6 for Windows does not

H49: Use semantic markup to mark emphasized or special text (2/2)

- This example shows how to use the em and strong elements to emphasize text

What she *really* meant to say was, "This is not ok, it is **excellent**!"

```
<p>
  What she <em>really</em> meant to say
was, "This is not ok, it is
<strong>excellent</strong>!"
</p>
```

- Superscripts and subscripts are created using the sup and sub elements

Beth received 1st place in the 9th grade science competition.

The chemical notation for water is H_2O.

```
<p>Beth received 1<sup>st</sup> place in
the 9<sup>th</sup> grade science
competition.</p>
<p>The chemical notation for water is
H<sub>2</sub>O.</p>
```

- Here is the use of the q element to mark up a shorter quotation. CSS rules to suppress automatic generation of quotes are provided for those user agents that do support the q element, to prevent double-quoted content

Helen Keller said, "Self-pity is our worst enemy and if we yield to it, we can never do anything good in the world."

```
<style type="text/css">
q:before { content: ""; }
q:after { content: ""; }
</style>
<p>Helen Keller said, "<q>Self-pity is our worst
enemy and if we yield to it,
we can never do anything good in the world.</q>"
</p>
```

G140: Separate information and structure from presentation to enable different presentations

- The objective of this technique is to facilitate the interaction of assistive technology with content by logically separating the content's structural encoding from the presentational encoding
  - Structural encoding is the indication of elements such as headings, paragraphs, lists, tables, etc., and is done by using technology features reserved for the purpose
  - Presentational encoding is the indication of formatting effects, such as typeface, color, size, position, borders, etc., and is also supported by technology features
- Providing separate structure, functionality, and presentation layers allows the semantics implied by the formatting to become programmatically determined via the structure layer
- Example: HTML with CSS
  - An HTML document uses the structural features of HTML, such as paragraphs, lists, headings, etc., and avoids presentational features such as font changes, layout hints, etc. CSS is used to format the document based on its structural properties. Well-crafted "class" attributes in the HTML extend the semantics of the structural markup if needed to allow more flexible formatting with CSS. Assistive technologies can substitute or extend the CSS to modify presentation, or ignore the CSS and interact directly with the structural encoding

Table markup

- H39: Use caption elements to associate data table captions with data tables (HTML)
  - The caption for a table is a table identifier and acts like a title or heading for the table
- H73: Use the summary attribute of the table element to give an overview of data tables (HTML)
  - It is not visually presented. . If caption and summary are used, information should not be duplicated
- H43: Use id and headers attributes to associate data cells with header cells in data tables (HTML)
  - It is used when data cells are associated with more than one row and/or one column header
  - This allows screen readers to speak the headers associated with each data cell

Grade percentages for the final course grade

Homework	Exams			Projects		
	1	2	Final	1	2	Final
15%	15%	15%	20%	10%	10%	15%

```
<table summary="Select the category of your interest, then navigate down to listen to the subcategories and further down to listen to the corresponding percentage">
<caption>Grade percentages for the final course grade</caption>
  <tr>
    <th rowspan="2" id="h">Homework</th>
    <th colspan="3" id="e">Exams</th>
    <th colspan="3" id="p">Projects</th>
  </tr>
  <tr>
    <th id="e1" headers="e">1</th>
    <th id="e2" headers="e">2</th>
    <th id="ef" headers="e">Final</th>
    <th id="p1" headers="p">1</th>
    <th id="p2" headers="p">2</th>
    <th id="pf" headers="p">Final</th>
  </tr>
  <tr>
   <td headers="h">15%</td>
   <td headers="e e1">15%</td>
   <td headers="e e2">15%</td>
   <td headers="e ef">20%</td>
   <td headers="p p1">10%</td>
   <td headers="p p2">10%</td>
   <td headers="p pf">15%</td>
  </tr>
 </table>
```

H71: Provide a description for groups of form controls using fieldset and legend elements

- The objective of this technique is to provide a semantic grouping for related form controls. This allows users to understand the relationship of the controls and interact with the form more quickly and effectively
- Grouping controls is most important for related radio buttons and checkboxes
- Example: a test item with one question and five possible answers

The play *Hamlet* was written by:

- William Shakespeare
- Rudyard Kipling
- George Bernard Shaw
- Ernest Hemingway
- Charles Dickens

```
<fieldset>
  <legend>The play <cite>Hamlet</cite> was written by:</legend>
  <input type="radio" id="shakesp" name="hamlet" checked="checked" value="a">
  <label for="shakesp">William Shakespeare</label><br />
  <input type="radio" id="kipling" name="hamlet" value="b">
  <label for="kipling">Rudyard Kipling</label><br />
  <input type="radio" id="gbshaw" name="hamlet" value="c">
  <label for="gbshaw">George Bernard Shaw</label><br />
  <input type="radio" id="hem" name="hamlet" value="d">
  <label for="hem">Ernest Hemingway</label><br />
  <input type="radio" id="dickens" name="hamlet" value="e">
  <label for="dickens">Charles Dickens</label>
</fieldset>
```

H85: Use OPTGROUP to group OPTION elements inside a SELECT

- The following combo box collects data about favorite foods. Grouping by type allows users to select their preference more quickly

```
<form action="http://example.com/prog/someprog" method="post">
  <label for="food">What is your favorite food?</label>
  <select id="food" name="food">
    <optgroup label="Fruits">
      <option value="1">Apples</option>
      <option value="3">Bananas</option>
      <option value="4">Peaches</option>
      <option value="5">...</option>
    </optgroup>
    <optgroup label="Vegetables">
      <option value="2">Carrots</option>
      <option value="6">Cucumbers</option>
      <option value="7">...</option>
    </optgroup>
    <optgroup label="Baked Goods">
      <option value="8">Apple Pie</option>
      <option value="9">Chocolate Cake</option>
      <option value="10">...</option>
    </optgroup>
  </select>
</form>
```

H48: Use ol, ul and dl for lists or groups of links

- The objective of this technique is to create lists of related items using list elements appropriate for their purposes
  - The ol element is used when the list is ordered
  - The ul element is used when the list is unordered
- Definition lists (dl) are used to group terms with their definitions
- When markup is used that visually formats items as a list but does not indicate the list relationship, users may have difficulty in navigating the information. An example of such visual formatting is including asterisks in the content at the beginning of each list item and using <br\> elements to separate the list items.
- Some assistive technologies allow users to navigate from list to list or item to item
- Style sheets can be used to change the presentation of the lists while preserving their integrity
- The list structure (ul/ol) is also useful to group hyperlinks. When this is done, it helps screen reader users to navigate from the first item in a list to the end of the list or jump to the next list. This helps them to bypass groups of links if they choose to

H42: Use h1-h6 to identify headings

- Use HTML and XHTML heading markup to provide semantic code for headings in the content
- Heading markup will allow assistive technologies to present the heading status of text to a user
  - A screen reader can recognize the code and announce the text as a heading with its level, beep or provide some other auditory indicator
  - Screen readers are also able to navigate heading markup which can be an effective way for screen reader users to more quickly find the content of interest
- Example

Plant Foods that Humans Eat

There are an abundant number of plants that humans eat...

Fruit

A fruit is a structure of a plant that contains its seeds...

Apple

The apple is the pomaceous fruit of the apple tree...

Orange

The orange is a hybrid of ancient cultivated origin...

Vegetables

A vegetable is an edible plant or part of a plant other than a sweet fruit ...

Broccoli

Broccoli is a plant of the mustard/cabbage family ...

Brussels sprouts

The Brussels sprout of the Brassicaceae family, is a Cultivar group of wild cabbage ...

```
<h1 style="color:red">Plant Foods that Humans Eat</h1>
<p>There are an abundant number of plants that humans eat...</p>
<h2 style="color:maroon">Fruit</h2>
<p> A fruit is a structure of a plant that contains its seeds...</p>
<h3>Apple</h3>
<p>The apple is the pomaceous fruit of the apple tree...</p>
<h3>Orange</h3>
<p>The orange is a hybrid of ancient cultivated origin...</p>

<h2 style="color:maroon">Vegetables</h2>
<p>A vegetable is an edible plant or part of a plant other than a sweet fruit ...</p>
<h3>Broccoli</h3>
<p>Broccoli is a plant of the mustard/cabbage family ... </p>
<h3>Brussels sprouts</h3>
<p>The Brussels sprout of the Brassicaceae family, is a Cultivar group of wild
cabbage ...</p>
```

SC 1.3.2 Meaningful Sequence (1/2)

- When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined (Level A)
  - The intent of this Success Criterion is to enable a user agent to provide an alternative presentation of content while preserving the reading order needed to understand the meaning
  - A sequence is meaningful if the order of content in the sequence cannot be changed without affecting its meaning
    - For example, if a page contains two independent articles, the relative order of the articles may not affect their meaning, as long as they are not interleaved. In such a situation, the articles themselves may have meaningful sequence, but the container that contains the articles may not have a meaningful sequence
  - The semantics of some elements define whether or not their content is a meaningful sequence
    - For instance, in HTML, text is always a meaningful sequence. Tables and ordered lists are meaningful sequences, but unordered lists are not
  - Example : CSS is used to position a navigation bar, the main story on a page, and a side story. The visual presentation of the sections does not match the programmatically determined order, but the meaning of the page does not depend on the order of the sections

SC 1.3.2 Meaningful Sequence (2/2)

- Techniques:
  - Order the content in a meaningful sequence for all the content in the Web page (G57)
  - Use a Unicode right-to-left mark (RLM) or left-to-right mark (LRM) to mix text direction inline
    - For example, when mixing languages with different directionality in HTML, the bidirectional algorithm may position punctuation in the wrong location in the visual rendering
  - Use the dir attribute on an inline element to resolve problems with nested directional runs
    - A nested directional run is a run of text that includes mixed directional text, for example, a paragraph in English containing a quoted Hebrew sentence which in turn includes an English phrase
  - Position content based on structural markup
    - Using the positioning properties of CSS2, content may be displayed at any position on the user's viewport. Using structural elements ensures that the meaning of the content can still be determined when styling is not available
  - Use CSS letter-spacing to control spacing within a word
    - The CSS letter-spacing property helps developers control the amount of white space between characters. This is recommended over adding blank characters to control the spacing, since the blank characters can change the meaning and pronunciation of the word
  - Make the DOM order match the visual order
    - The objective of this technique is to ensure that the order of content in the source code is the same as the visual presentation of the content. The order of content in the source code can be changed by the author to any number of visual presentations with CSS. When the source order matches the visual order, everyone will read the content and interact with it in the same (correct) order

G57: Order the content in a meaningful sequence for all the content in the Web page (1/2)

- The objective of this technique is to ensure that the order of content presented to assistive technologies allows the user to make sense of the content
  - Some techniques permit the content to be rendered visually in a meaningful sequence even if this is different from the order in which the content is encoded in the underlying source file
  - Note: controlling the placement of blocks of content in an HTML document using layout tables may produce a rendering in which related information is positioned together visually, but separated in the content stream. Since layout tables are read row by row, if the caption of an illustration is placed in the row following the illustration, it may be impossible to associate the caption with the image

G57: Order the content in a meaningful sequence for all the content in the Web page (2/2)

- Example: A Web page from a museum exhibition contains a navigation bar containing a long list of links
  - The page also contains an image of one of the pictures from the exhibition, a heading for the picture, and a detailed description of the picture
  - The links in the navigation bar form a meaningful sequence
  - The heading, image, and text of the description also form a meaningful sequence
  - CSS is used to position the elements on the page

Without CSS

Markup:

```
<h1>My Museum Page</h1>
<ul id="nav">
                <li><a href="#">Link 1</a></li> ...
                <li><a href="#">Link 10</a></li>
</ul>
<div id="description">
<h2>Mona Lisa</h2>
<p> <img src="img.png" alt="Mona Lisa"> </p>
<p>...detailed description of the picture...</p>
</div>
```

CSS:

```
ul#nav {
                float: left;
                width: 9em;
                list-style-type: none;
                margin: 0;
                padding: 0.5em;
                color: #fff;
                background-color: #063;
}
ul#nav a {
                display: block;
                width: 100%;
                text-decoration: none;
                color: #fff;
                background-color: #063;
}
div#description {  margin-left: 11em; }
```

SC 1.3.3 Sensory Characteristics

- Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, size, visual location, orientation, or sound (Level A)
  - Some content relies on knowledge of the shape or position of objects that are not available from the structure of the content (for example, "round button" or "button to the right“). Some users with disabilities are not able to perceive shape or position due to the nature of the assistive technologies they use. This Success Criterion requires that additional information be provided to clarify anything that is dependent on this kind of information
    - Providing information using shape and/or location, however, is an effective method for many users including those with cognitive limitations. This provision should not discourage those types of cues as long as the information is also provided in other ways
  - Example : An on-line multi-page survey
    - An on-line multi-page survey uses a link implemented as a green arrow icon placed in the lower right hand corner of the content to move from one survey page to the next. The arrow is clearly labeled with "Next" and the instructions state, "To move to the next section of the survey, select the green arrow icon labeled 'Next' in the lower right below the last survey question."
    - This example uses both positioning, color and labeling to help identify the icon
  - Techniques
    - Providing textual identification of items that otherwise rely only on sensory information to be understood (G96)

G96: Provide textual identification of items that otherwise rely only on sensory information to be understood

- The objective of this technique is to ensure that items within a Web page are referenced in the content not only by shape, size, sound or location, but also in ways that do not depend on that sensory perception
  - Example 1
    - A round button is provided on a form to submit the form and move onto the next step in a progression. The button is labeled with the text "go." The instructions state, "to submit the form press the round button labeled go”
    - This includes both shape and textual information to locate the button
  - Example 2
    - The following layout places a button in the lower right corner and indicates it by position. An indication of the text label clarifies which button to use for users who access a linearized version in which the position is not meaningful

Related source code:
http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/G96

1.4 Distinguishable: Make it easier for users to see and hear content including separating foreground from background

Use of color

- Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element

Audio Control

- If any audio on a Web page plays automatically for more than 3 seconds, either a mechanism is available to pause or stop the audio, or a mechanism is available to control audio volume independently from the overall system volume level

Contrast (Minimum)

- The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for the following
  - Large Text: Large-scale text and images of large-scale text have a contrast ratio of at least 3:1
  - Incidental: Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement
  - Logotypes: Text that is part of a logo or brand name has no minimum contrast requirement

Resize text

- Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality

Images of Text

- If the technologies being used can achieve the visual presentation, text is used to convey information rather than images of text except for the following:
  - Customizable: The image of text can be visually customized to the user's requirements
  - Essential: A particular presentation of text is essential to the information being conveyed

SC 1.4.1 Use of Color (1/2)

- Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element. (Level A)
  - All users should be able to access information that is conveyed by color differences, that is, by the use of color where each color has a meaning assigned to it
  - If the information is conveyed through color differences in an image (or other non-text format), the color may not be seen by users with color deficiencies. In this case, providing the information conveyed with color through another visual means ensures users who cannot see color can still perceive the information
  - Color is an important asset in design of Web content, enhancing its aesthetic appeal, its usability, and its accessibility. However, some users have difficulty perceiving color. People with partial sight often experience limited color vision, and many older users do not see color well. In addition, people using text-only, limited-color or monochrome displays and browsers will be unable to access information that is presented only in color.
  - Examples of information conveyed by color differences: “required fields are red", “error is shown in red", and “Mary's sales are in red, Tom's are in blue“
  - Examples of indications of an action include: using color to indicate that a link will open in a new window or that a database entry has been updated successfully
  - An example of prompting a response would be: using highlighting on form fields to indicate that a required field had been left blank

SC 1.4.1 Use of Color (2/2)

- Techniques
  - Situation A: If the color of particular words, backgrounds, or other content is used to indicate information:
    - Ensuring that information conveyed by color differences is also available in text (G14)
    - Including a text cue for colored form control labels (H92)
    - Ensuring that additional visual cues are available when text color differences are used to convey information (G182)
    - Using a contrast ratio of 3:1 with surrounding text and providing additional visual cues on focus for links or controls where color alone is used to identify them (G183)
  - Situation B: If color is used within an image to convey information:
    - Using color and pattern (G111)
    - Ensuring that information conveyed by color differences is also available in text (G14)

H92: Include a text cue for colored form control labels

- Example: A form with required fields
- The instructions for an online form say, "Required fields are shown in red and marked with (required)." The cue "(required)" is included within the label element

Last name (required):

```
<label for="lastname" class="required">Last name (required): </label>
<input id="lastname" type="text" size="25" value=""/>
<style type="text/css">
  .required {
    color=red;
  }
</style>
```

SC 1.4.2 Audio Control

- If any audio on a Web page plays automatically for more than 3 seconds, either a mechanism is available to pause or stop the audio, or a mechanism is available to control audio volume independently from the overall system volume level (Level A)
  - Individuals who use screen reading software can find it hard to hear the speech output if there is other audio playing at the same time. This difficulty is exacerbated when the screen reader's speech output is software based (as most are today) and is controlled via the same volume control as the sound
  - Therefore, it is important that the user be able to turn off the background sound
  - Note: Playing audio automatically when landing on a page may affect a screen reader user's ability to find the mechanism to stop it because they navigate by listening and automatically started sounds might interfere with that navigation
    - Therefore, the practice of automatically starting sounds (especially if they last more than 3 seconds) is discouraged
    - Instead it is encouraged that the sound be started by an action initiated by the user after they reach the page, rather than requiring that the sound be stopped by an action of the user after they land on the page

SC 1.4.3 Contrast (Minimum) (1/2)

- Contrast (Minimum): The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for the following: (Level AA)
  - Large Text: Large-scale text and images of large-scale text have a contrast ratio of at least 3:1;
  - Incidental: Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement
  - Logotypes: Text that is part of a logo or brand name has no minimum contrast requirement

SC 1.4.3 Contrast (Minimum) (2/2)

- Example Techniques
  - Situation A: text is less than 18 point if not bold and less than 14 point if bold
    - G18: Ensuring that a contrast ratio of at least 4.5:1 exists between text (and images of text) and background behind the text
  - Situation B: text is at least 18 point if not bold and at least 14 point if bold
    - G145: Ensuring that a contrast ratio of at least 3:1 exists between text (and images of text) and background behind the text
- Useful Resources:
  - Contrast Analyser
    - http://www.paciellogroup.com/resources/contrast-analyser.html

SC 1.4.4 Resize text

- Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality
  - The intent of this Success Criterion is to ensure that visually rendered text, including text-based controls (text characters that have been displayed so that they can be seen [vs. text characters that are still in data form such as ASCII]) can be scaled successfully so that it can be read directly by people with mild visual disabilities, without requiring the use of assistive technology such as a screen magnifier
- Example Techniques:
  - Ensure that text containers resize when the text resizes AND use measurements that are relative to other measurements in the content by using one or more of the following techniques:
  - Techniques for relative measurements
    - C12: Use percent for font sizes (CSS)
      e.g. strong {font-size: 120%}
    - C13: Use named font sizes (CSS)
      e.g. strong {font-size: larger}
    - C14: Use em units for font sizes (CSS)
      e.g. strong {font-size: 1.6em}
  - Techniques for text container resizing
    - G146: Use liquid layout
      - Example: liquidLayoutExample.html
      - http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/G146

2. Operable

GUIDELINES:

2.1 Make all functionality available from a keyboard

2.2 Provide users enough time to read and use content

2.3 Do not design content in a way that is known to cause seizures

2.4 Provide ways to help users navigate, find content, and determine where they are

Make all functionality available from a keyboard

Keyboard

- All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints

No Keyboard Trap

- If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving focus away.

SC 2.1.1 Keyboard (1/3)

- All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints (Level A)
  - This exception relates to the underlying function, not the input technique. For example, if using handwriting to enter text, the input technique (handwriting) requires path-dependent input but the underlying function (text input) does not
  - This does not forbid and should not discourage providing mouse input or other input methods in addition to keyboard operation
  - When content can be operated through a keyboard or alternate keyboard, it is operable by people with no vision (who cannot use devices such as mice that require eye-hand coordination) as well as by people who must use alternate keyboards or input devices that act as keyboard emulators
  - Most actions carried out by a pointing device can also be done from the keyboard (for example, clicking, selecting, moving, sizing)
    - However, there is a small class of input that is done with a pointing device that cannot be done from the keyboard in any known fashion without requiring an inordinate number of keystrokes. Free hand drawing, watercolor painting, and flying a helicopter through an obstacle course are all examples of functions that require path dependent input
    - Drawing straight lines, regular geometric shapes, re-sizing windows and dragging objects to a location (when the path to that location is not relevant) do not require path dependent input

SC 2.1.1 Keyboard (2/3)

- Examples
  - A drawing Program
    - A drawing program allows users to create, size, position and rotate objects from the keyboard
  - Moving between and connecting discrete points.
    - A connect-the-dots program allows the user to move between dots on a screen and use the spacebar to connect the current dot to the previous one
  - Exception - Painting Program
    - A watercolor painting program passes as an exception because the brush strokes vary depending on the speed and duration of the movements
  - Exception - Model helicopter flight training simulator.
    - A model helicopter flight training simulator passes as an exception because the nature of the simulator is to teach real-time behavior of a model helicopter
  - A PDA with an optional keyboard
    - A PDA device that is usually operated via a stylus has an optional keyboard that can be attached. The keyboard allows full Web browsing in standard fashion. The Web content is operable because it was designed to work with keyboard-only access

SC 2.1.1 Keyboard (3/3)

- Example Techniques:
  1. G202: Ensuring keyboard control for all functionality
     - Example: A page with images used as links changes when the user hovers over the image with a mouse. To provide keyboard users with a similar experience, the image is also changed when a user tabs to it.

```
<html>
  <head>
    <style type="text/css">
      a:hover img, a:focus img {
        border: 2px solid blue;
      }
    </style>
  </head>
  <body>
    <a href="http://go.somewhere" >
      <img src="./carousel.gif“ alt="an artistic carousel" />
    </a>
  </body>
</html>
```

SC 2.1.2 No Keyboard Trap

- If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving focus away (Level A)
  - The intent of this Success Criterion is to ensure that that content does not "trap" keyboard focus within subsections of content on a Web page. This is a common problem when multiple formats are combined within a page and rendered using plug-ins or embedded applications
  - Examples
    - A calendar widget. A calendar widget allows users to add, remove or update items in their calendar using the keyboard. The controls in the widget are part of the tab order within the Web page, allowing users to tab through the controls in the widget as well as to any links or controls that follow
    - A puzzle applet. Once a user tabs into an applet, further tabs and other keystrokes are handled by the applet. Instructions describing the keystroke used to exit the applet are provided prior to the applet as well as within the applet itself
    - A modal dialog box. A Web application brings up a dialog box. At the bottom of the dialog are two buttons, Cancel and OK. When the dialog has been opened, focus is trapped within the dialog; tabbing from the last control in the dialog takes focus to the first control in the dialog. The dialog is dismissed by activating the Cancel button or the OK button

2.2 Enough Time: Provide users enough time to read and use content: Overview

Timing Adjustable

- **Turn off**: The user is allowed to turn off the time limit before encountering it
- **Adjust**: The user is allowed to adjust the time limit before encountering it over a wide range that is at least ten times the length of the default setting; or
- **Extend**: The user is warned before time expires and given at least 20 seconds to extend the time limit with a simple action and the user is allowed to extend the time limit at least ten times
- **Real-time Exception**: The time limit is a required part of a real-time event (for example, an auction), and no alternative to the time limit is possible
- **Essential Exception**: The time limit is essential and extending it would invalidate the activity
- **20 Hour Exception**: The time limit is longer than 20 hours.

Pause, Stop, Hide

- **Moving, blinking, scrolling**: For any moving, blinking or scrolling information that (1) starts automatically, (2) lasts more than five seconds, and (3) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it unless the movement, blinking, or scrolling is part of an activity where it is essential; and
- **Auto-updating**: For any auto-updating information that (1) starts automatically and (2) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it or to control the frequency of the update unless the auto-updating is part of an activity where it is essential.

SC 2.2.1 Timing Adjustable (1/3)

- If Web functions are time-dependent, it will be difficult for some users to perform the required action before a time limit occurs
- Providing options to disable time limits, customize the length of time limits, or request more time before a time limit occurs helps those users who require more time than expected to successfully complete tasks
  - These options are listed in the order that will be most helpful for the user: disabling time limits is better than customizing the length of time limits, which is better than requesting more time before a time limit occurs
- Any process that happens without user initiation after a set time or on a periodic basis is a time limit
  - This includes partial or full updates of content (for example, page refresh), changes to content, or the expiration of a window of opportunity for a user to react to a request for input
- It also includes content that is advancing or updating at a rate beyond the user's ability to read and/or understand it
  - In other words, animated, moving or scrolling content introduces a time limit on a users ability to read content

SC 2.2.1 Timing Adjustable (2/3)

- Examples
  - A Web site uses a client side time limit to help protect users who may step away from their computer. After a period of inactivity the Web page asks if the user needs more time. If it doesn't get a response – it times out
  - A Web page has a field that automatically updates with the latest headlines in a rotating fashion. There is an interactive control that allows the user to extend the length of time between each update to as much as ten times the default. The control can be operated with either a mouse or a keyboard
  - In an auction, there is a time limit on the amount of time a user has to submit a bid. Since the time limit applies to all users who want to bid on a particular item, it would be unfair to extend the time limit for any one particular user. Therefore, a time limit is required for this type of activity and no extension, adjustment, or deactivation of the time limit is required by this Success Criteria
  - A ticket-purchasing site allows the user two minutes to confirm purchase of selected seats, but warns the user when their time is almost out and allows the user to extend this time limit some number of times with a simple action such as clicking a "Extend time limit" button

SC 2.2.1 Timing Adjustable (3/3)

- Example Techniques:
  1. G133: Providing a checkbox on the first page of a multipart form that allows users to ask for longer session time limit or no session time limit
     - Minimize the risk that users with disabilities will lose their work by providing a checkbox to request additional time to complete multipart forms
     - Example: *A checkbox for requesting a specific extension*. A Web page contains the first part of a five-part form. Immediately following the general instructions for completing the form is a checkbox with the label, "Allow an additional 15 minutes to complete each part of this form."
  2. G198: Providing a way for the user to turn the time limit off
     - Provide a mechanism for people who cannot complete tasks within a specified time limit to turn off the time limit
     - Example: A page has a listing of news headlines that automatically update every minute. At the top of the page is a link that turns off the updating.

SC 2.2.2 Pause, Stop, Hide (1/2)

- The intent of this Success Criterion is to avoid distracting users during their interaction with a Web page
- Content that is paused can either resume in real-time or continue playing from the point in the presentation where the user left off.
  1. Pausing and resuming where the user left off is best for users who want to pause to read content and works best when the content is not associated with a real-time event or status
  2. Pausing and jumping to current display (when pause is released) is better for information that is real-time or "status" in nature. For example, weather radar, a stock ticker, a traffic camera, or an auction timer, would present misleading information if a pause caused it to display old information when the content was restarted
- Examples
  - A Web site helps users understand 'how things work' through animations that demonstrate processes. Animations have "pause" and "restart" buttons
  - A stock ticker has "pause" and "restart" buttons. Pausing the ticker causes it to pause on the currently displayed stock. Restarting causes the ticker to resume from the stopped point but with a notice that the display is delayed. Since the intent of the stock ticker is usually to provide realtime information, there might also be a button that would advance the ticker to the most recently traded stock
  - An animation runs in the upper portion of the page but has a "freeze animation" button near the bottom of the animation

SC 2.2.2 Pause, Stop, Hide (2/2)

- Example Techniques:
  1. G4: Allowing the content to be paused and restarted from where it was paused
     - Provide a way to pause movement or scrolling of content
     - Example: *A site contains a scrolling news banner at the top of the page*. Users who need more time to read it can press the Escape key to pause the scrolling. Pressing Escape again restarts it.
  2. G187: Using a technology to include blinking content that can be turned off via the user agent
     - Ensure that blinking content can be turned off using user agent features
     - Example: *A page contains a blinking banner intended to draw the user's attention to it*. The banner is an animated gif image which repeats indefinitely. The user presses the "escape" key, which causes the user agent to stop the animation of all animated gif images on the page.
  3. G191: Providing a link, button, or other mechanism that reloads the page without any blinking
     - Allow people who cannot use a page with blinking content to turn the blinking content off.
     - Example: A page has blinking text at the top warning users that they should not submit the page without first registering. A link at the very top of the page reloads the page with the blinking text replaced with text that is styled to be highly visible but does not blink.

2.3 Seizures: Do not design content in a way that is known to cause seizures

- (SC 2.3.1) Three Flashes or Below Threshold: Web pages do not contain anything that flashes more than three times in any one second period, or the flash is below the general flash and red flash thresholds (Level A)
  - The intent of this Success Criterion is to allow users to access the full content of a site without inducing seizures due to photosensitivity
  - Individuals who have photosensitive seizure disorders can have a seizure triggered by content that flashes at certain frequencies for more than a few flashes
    - People are even more sensitive to red flashing than to other colors, so a special test is provided for saturated red flashing
  - These guidelines are based on guidelines for the broadcasting industry as adapted for computer screens, where content is viewed from a closer distance

2.4 Navigable: Provide ways to help users navigate, find content, and determine where they are

Bypass Blocks

- A mechanism is available to bypass blocks of content that are repeated on multiple Web pages

Page Titled

- Web pages have titles that describe topic or purpose

Focus Order

- If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability

Link Purpose (In Context)

- The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general

Multiple Ways

- More than one way is available to locate a Web page within a set of Web pages except where the Web Page is the result of, or a step in, a process.

Headings and Labels

- Headings and labels describe topic or purpose.

Focus Visible

- Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible

SC 2.4.1 Bypass Blocks (1/3)

- A mechanism is available to bypass blocks of content that are repeated on multiple Web pages
  - The intent of this Success Criterion is to allow people who navigate sequentially through content more direct access to the primary content of the Web page
  - Web pages and applications often have content that appears on other pages or screens
    - Examples of repeated blocks of content include but are not limited to navigation links, heading graphics, and advertising frames
  - Techniques
    1. Creating links to skip blocks of repeated material using one of the following techniques:
       - Adding a link at the top of each page that goes directly to the main content area (G1)
       - Adding a link at the beginning of a block of repeated content to go to the end of the block (G123)
       - Adding links at the top of the page to each area of the content (G124)
    2. Grouping blocks of repeated material in a way that can be skipped, using one of the following techniques:
       - Providing heading elements at the beginning of each section of content (HTML) (H69)
       - Using map to group links (HTML) (H50)
       - Using frame elements to group blocks of repeated material (HTML) AND H64: Using the title attribute of the frame and iframe elements (HTML) (H70)

SC 2.4.1 Bypass Blocks (2/3)

- Example Techniques (1/2):
  1. G1: Adding a link at the top of each page that goes directly to the main content area
     - Provide a mechanism to bypass blocks of material that are repeated on multiple Web pages by skipping directly to the main content of the Web page
     - Example: *An online newspaper*. An on-line newspaper contains many sections of information: a search function, a corporate banner, sidebars, minor stories, etc. The lead story is located in the middle of the page. The first link that the user reaches when tabbing through the page is titled "Skip to Lead Story". Activating the link moves visual focus to the story. Pressing tab again takes the user to the first link in the main story.
  2. G124: Adding links at the top of the page to each area of the content
     - Provide a mechanism to bypass blocks of material by providing a list of links to the different sections of the content
     - Example: The Web pages on a site all start with three links that navigate to the main content of that Web page, the search field, and the navigation bar

SC 2.4.1 Bypass Blocks (3/3)

- Example Techniques (2/2):
  3. H69: Providing heading elements at the beginning of each section of content
     - The objective of this technique is to use section headings to convey the structure of the content. Heading markup can be used:
       - to indicate start of main content
       - to mark up section headings within the main content area
       - to demarcate different navigational sections like top or main navigation, left or secondary navigation and footer navigation
       - to mark up images of text that are used as headings

```
<h1>Search Technical Periodicals</h1>
  <h2>Search</h2>
    <form action="search.php">
    <p>
      <label for="searchInput">
        Enter search topic:
      </label>
      <input type="text" size="30" id="searchInput">
      <input type="submit" value="Go">
    </p>
    </form>
  </h2>
  <h2>Available Periodicals</h2>
    <div class="jlinks">
      <a href="pcoder.com">Professional Coder</a> |
      <a href="algo.com">Algorithms</a> |
      <a href="jse.com">
        Journal of Software Engineering
      </a>
    </div>
  </h2>
  <h2>Search Results</h2>

 ... search results are returned in this section ...
```

SC 2.4.2 Page Titled

- Web pages have titles that describe topic or purpose
  - The intent of this Success Criterion is to help users find content and orient themselves within it by ensuring that each Web page has a descriptive title
  - Titles identify the current location without requiring users to read or interpret page content
  - When titles appear in site maps or lists of search results, users can more quickly identify the content they need
  - User agents make the title of the page easily available to the user for identifying the page
    - For instance, a user agent may display the page title in the window title bar or as the name of the tab containing the page
  - Techniques:
    - Provide descriptive titles for Web pages AND
    - Associate a title with a Web page using the title element

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>The World Wide Web Consortium</title>
  </head>
  <body> ... </body>
</html>
```

SC 2.4.3 Focus Order (1/3)

- If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability (Level A)
  - The intent of this Success Criterion is to ensure that when users navigate sequentially through content, they encounter information in an order that is consistent with the meaning of the content and can be operated from the keyboard
  - This reduces confusion by letting users form a consistent mental model of the content
  - Example:
    - A Web page implements modeless dialogs via scripting. When the trigger button is activated, a dialog opens. The interactive elements in the dialog are inserted in the focus order immediately after the button. When the dialog is open, the focus order goes from the button to the elements of the dialog, then to the interactive element following the button. When the dialog is closed, the focus order goes from the button to the following element
  - Counter-example:
    - A company's Web site includes a form that collects marketing data and allows users to subscribe to several newsletters published by the company. The section of the form for collecting marketing data includes fields such as name, street address, city, state or province, and postal code. Another section of the form includes several checkboxes so that users can indicate newsletters they want to receive. However, the tab order for the form skips between fields in different sections of the form, so that focus moves from the name field to a checkbox, then to the street address, then to another checkbox

SC 2.4.3 Focus Order (2/3)

- Techniques (1/2):
  1. G59: Placing the interactive elements in an order that follows sequences and relationships
     - Ensure that interactive elements receive focus in an order that follows sequences and relationships in the content
     - Example: *A form contains two, side-by-side sections of information*. One section contains information about an applicant; the other section contains information about the applicant's spouse. All the interactive elements in the applicant section receive focus before any of the elements in the spouse section. The elements in each section receive focus in the reading order of that section.
  2. C27: Making the DOM order match the visual order
     - Ensure that the order of content in the source code is the same as the visual presentation of the content
     - Example: An online newspaper has placed a navigation bar visually in the top left corner of the page directly below its initial logo. In the source code, the navigation elements appear after the elements encoding the logo.

SC 2.4.3 Focus Order (3/3)

- Techniques (2/2):
  1. H4: Creating a logical tab order through links, form controls, and objects
     - Provide a logical tab order when the default tab order does not suffice

Search for marriage records

Search criteria	Groom	Bride
First name	1	4
Last name	2	5
Place of birth	3	6

```
<form action="#" method="post"><table summary="the first column
contains the search criteria of the groom, the second column the search
criteria of the bride"><caption>Search for marriage records</caption>
    <tr>
      <th>Search criteria</th>
      <th>Groom</th>
      <th>Bride</th>
    </tr>
    <tr>
    <th>First name</th>
    <td>
      <input type="text" size="30" value="" name="groomfirst"
        title="First name of the groom" tabindex="1">
    </td>
    <td>
      <input type="text" size="30" value="" name="bridefirst"
        title="First name of the bride" tabindex="4">
    </td>
  </tr>
  <tr>
    <th>Last name</th>
    <td>
      <input type="text" size="30" value="" name="groomlast"
        title="Last name of the groom" tabindex="2">
    </td>
    <td>
      <input type="text" size="30" value="" name="bridelast"
        title="Last name of the bride" tabindex="5">
    </td>
  </tr>
  <tr>
    <th>Place of birth</th>
    <td>
      <input type="text" size="30" value="" name="groombirth"
        title="Place of birth of the groom" tabindex="3">
    </td>
     <td>
      <input type="text" size="30" value="" name="bridebirth"
        title="Place of birth of the bride" tabindex="6">
    </td>
  </tr>
</table></form>
```

SC 2.4.4 Link Purpose (1/4)

- The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general (Level A)
  - The intent of this Success Criterion is to help users understand the purpose of each link so they can decide whether they want to follow the link
  - Whenever possible, provide link text that identifies the purpose of the link without needing additional context
  - Assistive technology has the ability to provide users with a list of links that are on the Web page
  - Link text that is as meaningful as possible will aid users who want to choose from this list of links
  - Meaningful link text also helps those who wish to tab from link to link. Meaningful links help users choose which links to follow without requiring complicated strategies to understand the page
  - Links with the same destination should have the same descriptions, but links with different purposes and destinations should have different descriptions

SC 2.4.4 Link Purpose (2/4)

- Examples
  - A link contains text that gives a description of the information at that URI
    - A page contains the sentence "There was much bloodshed during the Medieval period of history." Where "Medieval period of history" is a link
  - A link is preceded by a text description of the information at that URI
    - A page contains the sentence "Learn more about the Government of Ireland's Commission on Electronic Voting at Go Vote!" where "Go Vote!" is a link
  - Both an icon and text are included in the same link
    - An icon of a voting machine and the text "Government of Ireland's Commission of Electronic Voting" are combined to make a single link. The alt text for the icon is null, since the purpose of the link is already described by the text of the link next to the icon
  - A list of book titles
    - A list of books is available in three formats: HTML, PDF, and mp3 (a recording of a person reading the book). To avoid hearing the title of each book three times (once for each format), the first link for each book is the title of the book, the second link says "PDF" and the third says, "mp3“

SC 2.4.4 Link Purpose (3/4)

- Example Techniques (1/2):
  1. G91: Providing link text that describes the purpose of a link
     - Describe the purpose of a link in the text of the link
     - Example: Describing the purpose of a link in HTML in the text content of the a element.

```
<a href="routes.html"> Current routes at Boulders Climbing Gym </a>
```

  2. H78: Identifying the purpose of a link using link text combined with its enclosing paragraph
     - Identify the purpose of a link from the link in its paragraph context
     - Example: *Identifying the purpose of a link in HTML from the link in its paragraph context*.

```
<h3>The final 15</h3>
<p>Coming soon to a town near you...the final 15 in the National Folk Festival lineup. <a href="final15.html">[Read more...]</a></p>

<h3>Folk artists get awards</h3>
<p>Performers from the upcoming National Folk Festival receive National Heritage Fellowships. <a href="nheritage.html">[Read more...]</a> </p>
…
```

The final 15

Coming soon to a town near you...the final 15 in the National Folk Festival lineup. [Read more...]

Folk artists get awards

Performers from the upcoming National Folk Festival receive National Heritage Fellowships. [Read more...]

…

SC 2.4.4 Link Purpose (4/4)

- Example Techniques (2/2):
  3. G189: Providing a control near the beginning of the Web page that changes the link text
     - Provide the user with a control near the beginning of the page that takes the user to a conforming alternate version of the Web page where the link text alone of each link is sufficient to determine its purpose out of context
     - Example: *Providing a Link to another Version*.

Version with short link text:

```
<h1>Books for download</h1>
 <p><a href="books-full-links.html" >Full link Version</a></p>
 <ul>
  <li>The History of the Web:
   <a href="history.docx" class="hist">Word</a>,
   <a href="history.pdf" class="hist">PDF</a>,
   <a href="history.html" class="hist">HTML</a>
  </li>
 ...
 </ul>
```

Books for download

Full link Version

- The History of the Web: Word, PDF, HTML
  ...

Version with full link text:

```
<h1>Books for download</h1>
 <p><a href="books-short-links.html" >Short link Version</a></p>
 <ul>
  <li>The History of the Web:
    <a href="history.docx" class="hist">The History of the Web(Word)</a>,
    <a href="history.pdf" class="hist">The History of the Web(PDF)</a>,
    <a href="history.html" class="hist">The History of the Web(HTML)</a>
  </li>
 ...
  </ul>
```

Books for download

Short link Version

- The History of the Web: The History of the Web(Word), The History of the Web(PDF), The History of the Web(HTML)
  ...

SC 2.4.5 Multiple ways

- More than one way is available to locate a Web page within a set of Web pages except where the Web Page is the result of, or a step in, a process
  - The intent of this Success Criterion is to make it possible for users to locate content in a manner that best meets their needs. Users may find one technique easier or more comprehensible to use than another
  - Techniques: Use two or more of the following techniques
    - Providing links to navigate to related Web pages (G125)
    - Providing a Table of Contents (G64)
    - Providing a site map (G63)
    - Providing a search function to help users find content (G161)
    - Providing a list of links to all other Web pages (G126)
    - Linking to all of the pages on the site from the home page (G185)

SC 2.4.6 Headings and Labels

- Headings and labels describe topic or purpose
  - The intent of this Success Criterion is to help users understand what information is contained in Web pages and how that information is organized. When headings are clear and descriptive, users can find the information they seek more easily, and they can understand the relationships between different parts of the content more easily. Descriptive labels help users identify specific components within the content
  - Labels and headings do not need to be lengthy. A word, or even a single character, may suffice if it provides an appropriate cue to finding and navigating content
  - Techniques:
    - Providing descriptive headings (G130)
    - Providing descriptive labels (G131)

SC 2.4.7 Focus Visible (1/3)

- Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible
  - The intent of this Success Criterion is to ensure that there is at least one mode of operation where the keyboard focus indicator can be visually located
  - Techniques
    - Using user interface components that are highlighted by the user agent when they receive focus (G149)
    - Using CSS to change the presentation of a user interface component when it receives focus (CSS) (C15)
    - Using the default focus indicator for the platform so that high visibility default focus indicators will carry over (G165)
    - Using an author-supplied, highly visible focus indicator (G195)
    - Using script to change the background color or border of the element with focus (Scripting) (SCR31)

SC 2.4.7 Focus Visible (2/3)

- Example Techniques (1/2):
  1. G149: Using user interface components that are highlighted by the user agent when they receive focus
     - Ensure that the focused component can be visually identified by the user by relying on user agent support
     - Example: When html text input fields receive focus, browsers display a blinking vertical bar at the insertion point in the text field.
     - Example: When html links receive focus, they are surrounded by a dotted focus highlight rectangle.
  2. G195: Using an author-supplied, highly visible focus indicator
     - Enhance the focus indicator in the browser, by creating a highly visible one in the content
     - Example: A Web page has a dark background color and light text and links. When focus lands on a link, the link is outlined with a bright yellow line, 3 pixels wide.
     - Example: A Web page includes a form inside a table. The borders of both the table and the form elements are thin, black lines. When focus lands on a form element, the element is outlined with a 5 pixel red line that is partially transparent.

SC 2.4.7 Focus Visible (3/3)

- Example Techniques (2/2):
  3. C15: Using CSS to change the presentation of a user interface component when it receives focus
     - Demonstrate how visual appearance may be enhanced via style sheets to provide visual feedback when an interactive element has focus or when a user hovers over it using a pointing device
     - Example: Highlighting elements that receive focus

```
<html>
 <head>
  <style type="text/css">
   input.text:focus {
     background-color: #7FFF00;
     color: #000;
   }

   input[type=checkbox]:focus + label,
   input[type=radio]:focus + label {
     background-color: #FF6;
     color: #000;
   }
  </style>
 </head>
 <body>
  <form method="post" action="form.php">
   <p><label for="fname">Name: </label>
     <input class="text" type="text" name="fname"
      id="fname" />
   </p>
   <p>
     <input type="radio" name="sex" value="male"
      id="sm" /> <label for="sm">Male</label><br />
     <input type="radio" name="sex" value="female"
      id="sf" /> <label for="sf">Female</label>
   <p>
  </form>
 </body>
</html>
```

3. Understandable

GUIDELINES:

3.1 Make text content readable and understandable

3.2 Make Web pages appear and operate in predictable ways

3.3 Help users avoid and correct mistakes

3.1 Readable: Make text content readable and understandable (1/2)

- (SC 3.1.1) Language of Page: The default human language of each Web page can be programmatically determined (Level A)
  - The intent of this Success Criterion is to ensure that content developers provide information in the Web page that user agents need to present text and other linguistic content correctly
  - Both assistive technologies and conventional user agents can render text more accurately when the language of the Web page is identified: Screen readers can load the correct pronunciation rules, Visual browsers can display characters and scripts correctly, Media players can show captions correctly.
  - Example
    - A Web page produced in Germany and written in HTML includes content in both German and English, but most of the content is in German. The default human language is identified as German (de) by the lang attribute on the html element
  - Example Techniques:
    - H57: Use language attributes on the html element
      - For XHTML 1.1 compatibility use the xml:lang attribute
      - Two-letter primary codes are reserved for [ISO639] language abbreviations. Two-letter codes include fr (French), de (German), it (Italian), nl (Dutch), el (Greek), es (Spanish), pt (Portuguese), ar (Arabic), he (Hebrew), ru (Russian), zh (Chinese), ja (Japanese), hi (Hindi), ur (Urdu), and sa (Sanskrit)

3.1 Readable: Make text content readable and understandable (2/2)

- (SC 3.1.2) Language of Parts: The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text. (Level AA)
  - When no other language has been specified for a phrase or passage of text, its human language is the default human language of the Web page. So the human language of all content in single language documents can be programmatically determined
    - Individual words or phrases in one language can become part of another language. For example, "rendezvous" is a French word that has been adopted in English, appears in English dictionaries, and is properly pronounced by English screen readers
    - Most professions require frequent use of technical terms which may originate from a foreign language. Such terms are usually not translated to all languages. The universal nature of technical terms also facilitate communication between professionals. Some common examples of technical terms include: Homo sapien, Alpha Centauri, hertz, and habeas corpus
  - Example: Alternative language links
    - An HTML Web page includes links to versions of the page in other languages (e.g., Deutsch, Français, Nederlands, Castellano, etc.). The text of each link is the name of the language, in that language. The language of each link is indicated via a lang attribute

3.2 Predictable: Make Web pages appear and operate in predictable ways

On focus

- When any component receives focus, it does not initiate a change of context

On input

- Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behavior before using the component

Consistent Navigation

- Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user

Consistent Identification

- Components that have the same functionality within a set of Web pages are identified consistently

SC 3.2.1 On Focus (1/2)

- When any component receives focus, it does not initiate a change of context (Level A)
  - The intent of this Success Criterion is to ensure that functionality is predictable as visitors navigate their way through a document
  - Any component that is able to trigger an event when it receives focus must not change the context
  - Examples of changing context when a component receives focus include, but are not limited to:
    - forms submitted automatically when a component receives focus
    - new windows launched when a component receives focus
    - focus is changed to another component when that component receives focus

SC 3.2.1 On Focus (2/2)

- Example techniques:
  - G107: Using "activate" rather than "focus" as a trigger for changes of context
    - The objective of this technique is to provide a method for activating things that is predictable by the user
    - With this technique, all changes of context would be triggered only by a specific action on the part of the user
    - Further, that action would be one that usually causes changes in context, such as clicking on a link or pressing a submit button. Actions that simply move the focus to an element would not cause a change of context
    - Example: A page pops up a new window only when the user clicks (or uses spacebar) on a button rather than using onfocus to pop up a new window

SC 3.2.2 On Input (1/2)

- Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behavior before using the component (Level A)
  - The intent of this Success Criterion is to ensure that entering data or selecting a form control has predictable effects
  - Changing the setting of any user interface component is changing some state in the control that will persist when the user is no longer interacting with it
  - So checking a checkbox or entering text into a text field changes its setting, but activating a link or a button does not
  - Changes in context can confuse users who do not easily perceive the change or are easily distracted by changes. Changes of context are appropriate only when it is clear that such a change will happen in response to the user's action
  - Example: A form is provided for creating calendar entries in a Web based calendaring and scheduling application. Along with the standard fields for subject, time and location, there is a set of radio buttons to select the type of calendar entry to create. The calendar entry type can be meeting, appointment or reminder. If the user selects the radio for meeting, additional fields are displayed on the page for entering the meeting participants. Different fields appear if the reminder button is chosen. Because only parts of the entry change and the overall structure remains the same the basic context remains for the user

SC 3.2.2 On Input (2/2)

- Example techniques:
  - G80: Provide a submit button to initiate a change of context using a technology-specific technique listed below
    - H32: Provide submit buttons (HTML)
      - http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/H32
    - H84: Use a button with a select element to perform an action (HTML)
      - The objective of this technique is to allow the user to control when an action is performed, rather than having the action occur as a side effect of choosing a value for the select element
      - Example:

```
<label for="region">Region:</label>
<select name="region" id="region">
  <option value="1">Heraklion</option>
  <option value="2"> Rethymnon</option>
  <option value="3">Chania</option>
</select>
<input type="button" value="Show" onclick = "...">
```

SC 3.2.3 Consistent Navigation

- Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user (Level AA)
  - The intent of this Success Criterion is to encourage the use of consistent presentation and layout for users who interact with repeated content within a set of Web pages and need to locate specific information or functionality more than once
  - It is important to note that the use of the phrase "same order" in this section is not meant to imply that subnavigation menus cannot be used or that blocks of secondary navigation or page structure cannot be used
  - Examples:
    - **A consistently located control.** A search field is the last item on every Web page in a site. users can quickly locate the search function
    - **Consistently positioned skip navigation controls.** A "skip navigation" link is included as the first link on every page in a Web site. The link allows users to quickly bypass heading information and navigational content and begin interacting with the main content of a page
    - **Skip to navigation link.** A skip to navigation link is provided to navigational content at the end of a page. The link is consistently located at the top of each page so that keyboard users can easily locate it when needed

SC 3.2.4 Consistent Identification

- Components that have the same functionality within a set of Web pages are identified consistently (Level AA)
  - The intent of this Success Criterion is to ensure consistent identification of functional components that appear repeatedly within a set of Web pages
  - A strategy that people who use screen readers use when operating a Web site is to rely heavily on their familiarity with functions that may appear on different Web pages. If identical functions have different labels on different Web pages, the site will be considerably more difficult to use
  - It may also be confusing and increase the cognitive load for people with cognitive limitations
  - This consistency extends to the text alternatives. If icons or other non-text items have the same functionality, then their text alternatives should be consistent as well
  - Example: A check mark icon functions as "approved", on one page but as "included" on another. Since they serve different functions, they have different text alternatives
  - Counter-example: A submit "search" button on one Web page and a "find" button on another Web page both have a field to enter a term and list topics in the Web site related to the term submitted. In this case, the buttons have the same functionality but are not labeled consistently

3.3 Input Assistance: Help users avoid and correct mistakes

Error Identification

- If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text

Labels or Instructions

- Labels or instructions are provided when content requires user input

Error Suggestion

- If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content

Error Prevention (Legal, Financial, Data)

- For Web pages that cause legal commitments or financial transactions for the user to occur, that modify or delete user-controllable data in data storage systems, or that submit user test responses, at least one of the following is true:
- **Reversible**: Submissions are reversible.
- **Checked**: Data entered by the user is checked for input errors and the user is provided an opportunity to correct them.
- **Confirmed**: A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.

SC 3.3.1 Error Identification (1/2)

- If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text. (Level A)
  - The intent of this Success Criterion is to ensure that users are aware that an error has occurred and can determine what is wrong. The error message should be as specific as possible. In the case of an unsuccessful form submission, re-displaying the form and indicating the fields in error is insufficient for some users to perceive that an error has occurred. Screen reader users, for example, will not know there was an error until they encounter one of the indicators. They may abandon the form altogether before encountering the error indicator, thinking that the page simply is not functional
  - Example Techniques
    - G83: Providing text descriptions to identify required fields that were not completed
      - Notify the user when a field that must be completed has not been completed
      - Example: A user attempts to submit a form but has neglected to provide input or select a choice in one or more mandatory fields. Using server-side validation, the omission is detected and the form is re-displayed with a text description at the top informing which mandatory fields were omitted. Each omitted mandatory field is also identified using a text label so that the user does not have to return to the list at the top of the form to find the omitted fields.

SC 3.3.1 Error Identification (2/2)

- Example Techniques (cont.)
  - G84: Providing a text description when the user provides information that is not in the list of allowed values
    - When users enter input that is validated, and errors are detected, the nature of the error needs to described to the user in manner they can access
    - Example: The user inputs invalid data on a form field and submits the form. The server returns the form, with the user's data still present, and indicates clearly in text at the top of the page that there were input errors. The text describes the nature of the error(s) and clearly indicates which field had the problem so the user can easily navigate to it to fix the problem.
  - G85: Providing a text description when user input falls outside the required format or values
    - Provide assistance in correcting input errors where the information supplied by the user is not accepted
    - Example: The user inputs invalid data on a form field and submits the form. The server returns the form, with the user's data still present, and indicates clearly in text at the top of the page that there were input errors. The text describes the nature of the error(s) and clearly indicates which field had the problem so the user can easily navigate to it to fix the problem

SC 3.3.2 Labels or Instructions (1/3)

- Labels or instructions are provided when content requires user input (Level A)
  - The intent of this Success Criterion is to help users avoid making mistakes when their input is required
    - To help avoid mistakes it is good user interface design to provide simple instructions and cues for entering information
  - Some users with disabilities may be more likely to make mistakes than users without disabilities or recovery from mistakes may be more difficult, making mistake avoidance an important strategy for users with disabilities
    - People with disabilities rely on well documented forms and procedures to interact with a page
    - Blind users need to know exactly what information should be entered into form fields and what the available choices are
    - Simple instructions visually connected to form controls can assist users with cognitive disabilities or those accessing a page using a screen magnifier
  - The intent of this Success Criterion is not to clutter the page with unnecessary information but to provide important cues and instructions that will benefit people with disabilities. Too much information or instruction can be just as much of a hindrance as too little
  - Examples:
    - A field for entering a date contains initial text which indicates the correct format for the date
    - A field for entering a given name is clearly labeled with "Given Name" and the field for family name is labeled "Family Name" to avoid confusion over which name is requested

SC 3.3.2 Labels or Instructions (2/3)

- Example Techniques (1/2):
  1. G131: Providing descriptive labels
     - Ensure that the label for any interactive component within Web content makes the component's purpose clear
     - Example: *A form asking the name of the user*. A form asks the name of the user. It consists of two input fields to ask for the first and last name. The first field is labeled "First name", the second is labeled "Last name".
  2. G162: Positioning labels to maximize predictability of relationships
     - When labels for form fields are positioned where the user expects them visually, it is easier to understand complex forms and to locate specific fields

Labels above text fields

Labels to the left of text fields

Labels to the right of radio buttons

SC 3.3.2 Labels or Instructions (3/3)

- Example Techniques (2/2):

3. H44: Using label elements to associate text labels with form controls
   - Use the *label* element to explicitly associate a form control with a label
   - Example: *A text input field*.

```
<label for="firstname">First name:</label>
<input type="text" name="firstname" id="firstname" />
```

4. H71: Providing a description for groups of form controls using fieldset and legend elements
   - Provide a semantic grouping for related form controls
   - Example: *A multiple choice test*.

```
<fieldset>
  <legend>The play <cite>Hamlet</cite> was written by:</legend>
  <input type="radio" id="shakesp" name="hamlet" checked="checked" value="a">
  <label for="shakesp">William Shakespeare</label><br />
  <input type="radio" id="kipling" name="hamlet" value="b">
  <label for="kipling">Rudyard Kipling</label><br />
  <input type="radio" id="gbshaw" name="hamlet" value="c">
  <label for="gbshaw">George Bernard Shaw</label><br />
  <input type="radio" id="hem" name="hamlet" value="d">
  <label for="hem">Ernest Hemingway</label><br />
  <input type="radio" id="dickens" name="hamlet" value="e">
  <label for="dickens">Charles Dickens</label>
</fieldset>
```

The play *Hamlet* was written by:

- (●) William Shakespeare
- ( ) Rudyard Kipling
- ( ) George Bernard Shaw
- ( ) Ernest Hemingway
- ( ) Charles Dickens

SC 3.3.3 Error Suggestion

- If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content (Level AA)
  - The intent of this Success Criterion is to ensure that users receive appropriate suggestions for correction of an input error if it is possible
  - Success Criterion 3.3.1 provides for notification of errors. However, persons with cognitive limitations may find it difficult to understand how to correct the errors. People with visual disabilities may not be able to figure out exactly how to correct the error
    - In the case of an unsuccessful form submission, users may abandon the form because although they may be aware that an error has occurred, they may be unsure of how to correct the error even though they are aware that it has occurred
  - Example: Suggestions from a Limited Set of Values
    - An input field requires that a month name be entered. If the user enters "12," suggestions for correction may include: **(a)** A list of the acceptable values, e.g., "Choose one of: January, February, March, April, May, June, July, August, September, October, November, December.“, **(b)** A description of the set of values, e.g., "Please provide the name of the month.”, **(c)** The conversion of the input data interpreted as a different month format, e.g., "Do you mean 'December'?"

SC 3.3.4 Error Prevention (Legal, Financial, Data)

- For Web pages that cause legal commitments or financial transactions for the user to occur, that modify or delete user-controllable data in data storage systems, or that submit user test responses, at least one of the following is true: (Level AA)
  1. Reversible: Submissions are reversible
  2. Checked: Data entered by the user is checked for input errors and the user is provided an opportunity to correct them
  3. Confirmed: A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission
  - The intent of this Success Criterion is to help users with disabilities avoid serious consequences as the result of a mistake when performing an action that cannot be reversed. For example, purchasing non-refundable airline tickets or submitting an order to purchase stock in a brokerage account are financial transactions with serious consequences
  - Users with disabilities may be more likely to make mistakes. People with reading disabilities may transpose numbers and letters, and those with motor disabilities may hit keys by mistake. Providing the ability to reverse actions allows users to correct a mistake that could result in serious consequences. Providing the ability to review and correct information gives the user an opportunity to detect a mistake before taking an action that has serious consequences

4. Robust

GUIDELINES:

4.1 Maximize compatibility with current and future user agents, including assistive technologies

Compatible: Maximize compatibility with current and future user agents, including assistive technologies

Parsing

- In content implemented using markup languages, elements have complete start and end tags, elements are nested according to their specifications, elements do not contain duplicate attributes, and any IDs are unique, except where the specifications allow these features

Name, Role, Value

- For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies

SC 4.4.1 Parsing (1/2)

- In content implemented using markup languages, elements have complete start and end tags, elements are nested according to their specifications, elements do not contain duplicate attributes, and any IDs are unique, except where the specifications allow these features (Level A)
  - The intent of this Success Criterion is to ensure that user agents, including assistive technologies, can accurately interpret and parse content
  - If the content cannot be parsed into a data structure, then different user agents may present it differently or be completely unable to parse it
  - Some user agents use "repair techniques" to render poorly coded content
  - Since repair techniques vary among user agents, authors cannot assume that content will be accurately parsed into a data structure or that it will be rendered correctly by specialized user agents, including assistive technologies, unless the content is created according to the rules defined in the formal grammar for that technology
  - In markup languages, errors in element and attribute syntax and failure to provide properly nested start/end tags lead to errors that prevent user agents from parsing the content reliably. Therefore, the Success Criterion requires that the content can be parsed using only the rules of the formal grammar

SC 4.4.1 Parsing (2/2)

- Example Techniques:
  1. G134: Validating Web pages
     - Avoid ambiguities in Web pages that often result from code that does not validate against formal specifications
     - Example: *Validating HTML*. HTML pages include a document type declaration (sometimes referred to as !DOCTYPE statement) and are valid according to the HTML version specified by the document type declaration. The developer can use off-line or online validators to check the validity of the HTML pages.
     - Example: Validating XML. XHTML, SVG, SMIL and other XML-based documents reference a Document Type Definition (DTD) or other type of XML schema. The developer can use online or off-line validators to check the validity of the XML documents.

     - **The W3C Markup Validation Service: http://validator.w3.org/**
     - **CSS Validation Service: http://jigsaw.w3.org/css-validator/**
     - **HTML/XML Validator: http://validator.aborla.net/**
     - **XML Validator - A Document Validation Service: http://www.javaview.de/validator/**

SC 4.4.2 Name, Role, Value

- For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies (Level A)
  - Note: This success criterion is primarily for Web authors who develop or script their own user interface components. For example, standard HTML controls already meet this success criterion when used according to specification
  - A particularly important state of a user interface control is whether or not it has focus. The focus state of a control can be programmatically determined, and notifications about change of focus are sent to user agents and assistive technology
  - Other examples of user interface control state are whether or not a checkbox or radio button has been selected, or whether or not a collapsible tree or list node is expanded or collapsed

Summary

Conclusions

- Web accessibility means that people with disabilities can use the Web
- The Web Content Accessibility Guidelines (WCAG) documents explain how to make Web content accessible to people with disabilities
- There are three conformance levels with WCAG: A, AA, AAA
- The 4 principles of Web accessibility in WCAG 2.0 are:
  - Perceivable
  - Operable
  - Understandable
  - Robust

WCAG 2.0 Documents

References

- Essential Components of Web Accessibility
  - http://www.w3.org/WAI/intro/components.php
- Web Content Accessibility Guidelines (WCAG) 2.0
  - http://www.w3.org/TR/WCAG20/
- Understanding WCAG 2.0
  - http://www.w3.org/TR/UNDERSTANDING-WCAG20/
- Recommended reading:
  - Usable and Accessible Form Validation and Error Recovery
    - http://webaim.org/techniques/formvalidation/
  - Quick tips for accessible headings
    - http://webarchive.nationalarchives.gov.uk/20100418065544/http://www.rnib.org.uk/wacblog/headings/quick-tips-for-accessible-headings/