WCAG 2.0 Reference Document

Section 1 – Perceivable

1.1 Text Alternatives – WCAG states that target sites should "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]".

1.1.1: Non-text Content (Level A) – With the exception of the instances indicated below, any non-text content that is displayed to the user has a text alternative that fulfills the same function. The exceptions are as follows: controls/input, time-based media, tests, sensory experiences, CAPTCHA, non-text content that is decorative. Some examples of non-text content could include data charts, audio recordings of speech, animations that illustrate a concept, video footage, photographs, linked thumbnails, or an image map[2].

Situation A: If a short description can serve the same purpose and present the same information as the non-text content:

Technique G94: Providing Short Text Alternative for Non-Text Content That Serves the Same Purpose and Presents the Same Information as The Non-Text Content[3]:

Examples

- A search button uses an image of a magnifying glass. The text alternative is "search" and not "magnifying glass".
- A picture shows how a knot is tied including arrows showing how the ropes go to make the knot. The text alternative describes how to tie the knot, not what the picture looks like.
- A picture shows what a toy looks like from the front. The text alternative describes a front view of the toy.
- An animation shows how to change a tire. A short text alternative describes what the animation is about. A long text alternative describes how to change a tire.
- A logo of the TechTron company appears next to each product in a list that is made by that and has a short text alternative that reads, "TechTron."
- A chart showing sales for October has an short text alternative of "October sales chart". It also has a long description that provides all of the information on the chart.
- A heading contains a picture of the words, "The History of War" in stylized text. The alt text for the picture is "The History of War".
- An image of a series of books on a shelf contains interactive areas that provide the navigation means to a Web page about the particular book. The text alternative "The books available to buy in this section. Select a book for more details about that book." describes the picture and the interactive nature.

Testing Procedure

1. Remove, hide, or mask the non-text content
2. Replace it with the text alternative
3. Check that nothing is lost (the purpose of the non-text content is met by the text alternative)
4. If the non-text content contains words that are important to understanding the content, the words are included in the text alternative

Situation B: If a short description cannot serve the same purpose and present the same information as the non-text content (e.g., a chart or diagram):

[1] https://www.w3.org/WAI/WCAG21/quickref/#non-text-content
[2] https://www.w3.org/WAI/WCAG21/Understanding/non-text-content.html
[3] https://www.w3.org/WAI/WCAG21/Techniques/general/G94.html

Technique G95: Providing Short Text Alternatives That Provide a Brief Description of The Non-Text Content[4]:

Example

- A chart showing sales for October has a short text alternative of "October sales chart". It also has a long description that provides all of the information on the chart.

Testing Procedure

1. Check for the presence of a short text alternative that provides a brief description of the non-text content.

Situation C: If non-text content is a control or accepts user input:

Technique G82: Providing A Text Alternative That Identifies the Purpose of The Non-Text Content[5]:

Examples

- An eye-hand coordination development applet has the following text alternative "Applet that uses the mouse and moving targets to develop eye-hand coordination"
- A camera applet that has a round disk where you push on the edges to control a remote camera and a slider in the middle for zooming has the following text alternative "Control for aiming and zooming remote video camera".

Testing Procedures

1. Remove, hide, or mask the non-text content
2. Replace it with the text alternative
3. Check that the purpose of the non-text content is clear - even if function is lost.

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:

Acceptable Technique Solution:

1. Providing a descriptive label[6]

Situation E: If non-text content is a CAPTCHA:

Technique G143: Providing a text alternative that describes the purpose of the CAPTCHA[7] **AND** Technique G144: Ensuring that the Web Page contains another CAPTCHA serving the same purpose using a different modality[8]

Examples

For G143

- A CAPTCHA test asks the user to type in text that is displayed in an obscured image. The text alternative is "Type the word in the image".
- 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".

For G144

- A Web page that includes a CAPTCHA test that must be completed successfully before the user can advance to the next step in a process. The page includes both a visual task, such as typing words displayed in a image, and an audio task, such as typing letters spoken in an audio file. A user with hearing disabilities who cannot pass the audio CAPTCHA may be able to pass the visual CAPTCHA.

[4] https://www.w3.org/WAI/WCAG21/Techniques/general/G95.html
[5] https://www.w3.org/WAI/WCAG21/Techniques/general/G82.html
[6] https://www.w3.org/WAI/WCAG21/quickref/?showtechniques=122%2C131%2C132%2C133%2C134%2C141%2C142%2C143%2C144%2C145%2C1410%2C1411%2C1412%2C244#non-text-content
[7] https://www.w3.org/WAI/WCAG21/Techniques/general/G143.html
[8] https://www.w3.org/WAI/WCAG21/Techniques/general/G144.html

- 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.

Testing Procedures
For G143
1. Remove, hide, or mask the CAPTCHA.
2. Replace it with the text alternative.
3. Check that the text alternative describes the purpose of the CAPTCHA.

For G144
For each CAPTCHA in a Web page:
1. Check that the Web page contains another CAPTCHA for the same purpose but using a different modality.

1.2 Time-based Media[9] – In order to reach at least AA conformance, WCAG simply required that the target site provided alternatives for any time-based media. This included pre-recorded audio and video, pre-recorded captions, audio descriptions or similar media alternatives, live-captions, and pre-recorded audio descriptions.

1.2.1 Audio-only and Video-only (Pre-Recorded) (Level A)[10] – The purpose of this requirement is to make time-based media information available through text-based alternatives since text could be displayed in any sensory modality (for instance, visual, auditory, or tactile) to suit the user's demands. Some examples of time-based media could include an audio recording of a speech, an audio recording of a press conference, an animation illustrating a concept, or a video file.

Situation A: If the content is prerecorded audio-only:

Technique G158: Providing an Alternative for Time-Based Media for Audio-only[11]

Examples
- A podcast includes a description of new features in a recent software release. It involves two speakers informally discussing the new and updated features and describing how they are used. One of the speakers works from a list of questions that is used to outline the discussion prior to recording. After the recording is complete, the outline is then edited and supplemented to match the dialogue etc. The resulting transcript is then made available on the speakers Web site along with the audio-only file. The text alternative that identifies the audio only content reads, "Episode 42: Zap Version 12 (text transcript follows)" and the link to the transcript is provided immediately following the audio-only content.

Testing Procedure
1. View the audio-only content while referring to the alternative for time-based media.
2. Check that the dialogue in the transcript matches the dialogue and information presented in the audio-only presentation.
3. If the audio includes multiple voices, check that the transcript identifies who is speaking for all dialogue.
4. Check that at least one of the following is true:
   a. The transcript itself can be programmatically determined from the text alternative for the audio-only content
   b. The transcript is referred to from the programmatically determined text alternative for the audio-only content
5. If the alternate version(s) are on a separate page, check for the availability of link(s) to allow the user to get to the other versions.

[9] https://www.w3.org/WAI/WCAG21/quickref/?showtechniques=131%2C132%2C133%2C134%2C141%2C142%2C143%2C144%2C145%2C1410%2C1411%2C1412#time-based-media
[10] https://www.w3.org/WAI/WCAG21/Understanding/audio-only-and-video-only-prerecorded.html
[11] https://www.w3.org/WAI/WCAG21/Techniques/general/G158.html

Technique SL17: Providing Static Alternative Content for Silverlight Media Playing in a MediaElement

Examples

- This example has a UI definition in XAML and interaction logic in C#. In this case the MediaElement has no visual representation itself and is 0x0 size because it plays audio only. As a simple placeholder, this example displays the text "Library of Congress Audio" to represent the media element as something visible in the UI. In addition to Play/Stop controls, this interface includes a Display Transcript button. Activating the button displays static text that represents the transcript of the audio.

Testing Procedure

1. Using a browser that supports Silverlight, open an HTML page that references a Silverlight application through an object tag. That application has audio-only media content and is expected to supply a text alternative, or has media that is expected to be replaced entirely with a transcript or similar text alternative.
2. Check for a control that indicates that activating it will supply static alternative content for the media. Activate the control.
3. Verify that the media control is replaced with alternate content, and that assistive technologies represent the change to the user interface

Situation B: If the content is prerecorded video-only:

Technique G159: Providing an Alternative for Time-Based media for Video-only Content[12]

Examples

- An animation shows how to assemble a woodworking project. There is no audio, but the animation includes a series of numbers to represent each step in the process as well as arrows and picture-in-picture highlights illustrating how the assembly is completed. It also includes short outtake animations illustrating what will happen if assembly is done incorrectly. A text alternative that identifies the video-only content reads, "Breadbox assembly video (text description follows)," and the text description of the video includes a full text description of each step in the video.

Testing Procedure

1. View the video-only content while referring to the alternative for time-based media.
2. Check that the information in the transcript includes the same information that is in the video-only presentation.
3. If the video includes multiple people or characters, check that the transcript identifies which person or character is associated with each action described.
4. Check that at least one of the following is true:
   a. The transcript itself can be programmatically determined from the text alternative for the video-only content
   b. The transcript is referred to from the programmatically determined text alternative for the video-only content
5. If the alternate version(s) are on a separate page, check for the availability of link(s) to allow the user to get to the other versions.

Technique G166: Providing Audio that Describes the Important Video Content and Describing it as Such[13]

Example

- A Web page has a link to a video-only presentation of a spaceship landing on Mars. The link to the video is a picture of a spaceship. Near the video is a link to an audio file of a person describing the video. This would look something like the following code example in HTML.

```
<a href="../video/marslanding.mp4"><img src="../images/spaceship.jpg"
                alt="Mars landing, video-only" width="193" height="255"/></a>
                <br />
                <a href="Mars_landing_audio.mp3">Audio description of "Mars Landing"</a>
```

[12] https://www.w3.org/WAI/WCAG21/Techniques/general/G159
[13] https://www.w3.org/WAI/WCAG21/Techniques/general/G166

Testing Procedure
For a Web page that contains video-only content:
1. Check that there is link to an audio alternative which describes the contents of the video immediately before or after the video-only content.

Technique SL17: Providing Static Alternative Content for Silverlight Media Playing in a MediaElement

Example
{See Above}

Testing Procedure
{See Above}

1.2.2 Captions (Pre-Recorded) (Level A)[14] – All prerecorded audio information in synced media must have included captions unless the media is expressly marked as a text alternative and is otherwise not captioned. Some examples might have included a captioned video recording, captions that describe non-verbal occurrences within a video, or a captioned description of a complex object depicted in a media file.

Technique G93: Providing Open (always visible) Captions[15]

Example
- In order to ensure that everyone can view their online movies, even if users do not know how to turn on captions in their media player, a library association puts the captions directly into the video.
- A news organization provides open captions on all of its material.

Testing Procedure
1. Watch the synchronized media with closed captioning turned off.
2. Check that captions (of all dialogue and important sounds) are visible.

Technique G87: Providing Closed Captions[16]

Example 1
- In order to ensure that users who are deaf can use their interactive educational materials, the college provides captions and instructions for turning on captions for all of their audio interactive educational programs.

Example 2
- The online movies at a media outlet all include captions and are provided in a format that allows embedding of closed captions.

Example 3
- Special caption files including synchronization information are provided for an existing movie. Players are available that can play the captions in a separate window on screen, synchronized with the movie window.

Example 4
- A video of a local news event has captions provided that can be played over the video or in a separate window depending on the player used.

Testing Procedure

[14] https://www.w3.org/WAI/WCAG21/Understanding/captions-prerecorded.html
[15] https://www.w3.org/WAI/WCAG21/Techniques/general/G93.html
[16] https://www.w3.org/WAI/WCAG21/Techniques/general/G87.html

1. Turn on the closed caption feature of the media player
2. View the synchronized media content
3. Check that captions (of all dialogue and important sounds) are visible

Differing Methods to Meet G87 –

Technique SM11: Providing Captions Through Synchronized Text Streams in SMIL 1.0[17]

Example 1: SMIL 1.0 caption sample for Quickime player

```
<?xml version="1.0" encoding="UTF-8"?>
<smil xmlns:qt="http://www.apple.com/quicktime/resources/smilextensions"
  xmlns="https://www.w3.org/TR/REC-smil" qt:time-slider="true">
  <head>
    <layout>
      <root-layout width="320" height="300" background-color="black"/>
      <region top="0" width="320" height="240" left="0" background-color="black"
      id="videoregion"/>
      <region top="240" width="320" height="60" left="0" background-color="black"
      id="textregion"/>
    </layout>
  </head>
  <body>
    <par>
      <video dur="0:01:20.00" region="videoregion" src="salesdemo.mov"
      alt="Sales Demo"/>
      <textstream dur="0:01:20.00" region="textregion" src="salesdemo_cc.txt"
      alt="Sales Demo Captions"/>
    </par>
  </body>
</smil>
```

Example 2: SMIL 1.0 caption sample for RealMedia player

```
<?xml version="1.0" encoding="UTF-8"?>
<smil xmlns="https://www.w3.org/TR/REC-smil">
  <head>
    <layout>
      <root-layout background-color="black" height="310" width="330"/>
      <region id="video" background-color="black" top="5" left="5"
      height="240" width="320"/>
      <region id="captions" background-color="black" top="250"
      height="60" left="5" width="320"/>
    </layout>
  </head>
  <body>
    <par>
      <video src="salesdemo.mpg" region="video" title="Sales Demo"
      alt="Sales Demo"/>
      <textstream src="salesdemo_cc.rt" region="captions"
      system-captions="on" title="captions"
      alt="Sales Demo Captions"/>
    </par>
  </body>
</smil>
```

The example shows a <par> segment containing a <video> and a <code><![CDATA[<textstream> tag. The system-captions attribute indicates that the textstream should be displayed when the user's player setting for captions indicates the preference for captions to be displayed. The <layout> section defines the regions used for the video and the captions.

Example 3: SMIL 1.0 caption sample with internal text streams

```
<?xml version="1.0" encoding="UTF-8"?>
```

[17] https://www.w3.org/WAI/WCAG21/Techniques/smil/SM11.html

```
<smil xmlns="https://www.w3.org/TR/REC-smil">
  <head>
    <layout>
      <root-layout background-color="black" height="310" width="330"/>
      <region id="video" background-color="black" top="5" left="5"
      height="240" width="320"/>
      <region id="captions" background-color="black" top="250"
      height="60" left="5" width="320"/>
    </layout>
  </head>
  <body>
    <par>
      <video src="salesdemo.mpg" region="video" title="Sales Demo"
      alt="Sales Demo"/>
      <text src="data:,This%20is%20inline%20text." region="captions" begin="0s"
      dur="3" alt="Sales Demo Captions">
        <param name="charset" value="iso-8859-1"/>
        <param name="fontFace" value="System"/>
        <param name="fontColor" value="yellow"/>
        <param name="backgroundColor" value="blue"/>
      </text>
    </par>
  </body>
</smil>
```

This example shows a <text> element that includes synchronized text streams within the SMIL file.

Testing Procedure

1. Enabled caption preference in player, if present
2. Play file with captions
3. Check whether captions are displayed

Technique SM12: Providing Captions Through Synchronized Text Streams in SMIL 2.0[18]

Example 1: SMIL 2.0 caption sample for RealMedia player

```
<?xml version="1.0" encoding="UTF-8"?>
<smil xmlns="https://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <root-layout backgroundColor="black" height="310" width="330"/>
      <region id="video" backgroundColor="black" top="5" left="5"
      height="240" width="320"/>
      <region id="captions" backgroundColor="black" top="250"
      height="60" left="5" width="320"/>
    </layout>
  </head>
  <body>
    <par>
      <video src="salesdemo.mpg" region="video" title="Sales Demo"
      alt="Sales Demo"/>
      <textstream src="salesdemo_cc.rt" region="captions" systemCaptions="on"
      title="captions" alt="Sales Demo Captions"/>
    </par>
  </body>
</smil>
```

The example shows a <par> segment containing a <video> and a <textstream> tag. The systemCaptions attribute indicates that the textstream should be displayed when the user's player setting for captions indicates the preference for captions to be displayed. The <layout> section defines the regions used for the video and the captions.

Example 2: SMIL 2.0 caption sample with internal text streams for RealMedia player

```
<?xml version="1.0" encoding="UTF-8"?>
<smil xmlns="https://www.w3.org/2001/SMIL20/Language">
```

[18] https://www.w3.org/WAI/WCAG21/Techniques/smil/SM12.html

```
  <head>
    <layout>
      <root-layout backgroundColor="black" height="310" width="330"/>
      <region id="video" backgroundColor="black" top="5" left="5"
      height="240" width="320"/>
      <region id="captions" backgroundColor="black" top="250"
      height="60" left="5" width="320"/>
    </layout>
  </head>
  <body>
    <par>
      <video src="salesdemo.mpg" region="video" title="Sales Demo"
      alt="Sales Demo"/>
      <text src="data:,This%20is%20inline%20text." region="captions"
      begin="0s" dur="3">
        <param name="charset" value="iso-8859-1"/>
        <param name="fontFace" value="System"/>
        <param name="fontColor" value="yellow"/>
        <param name="backgroundColor" value="blue"/>
      </text>
      <text src="data:,This%20is%20a%20second%20text."
      region="captions" begin="3s" dur="3">
        <param name="charset" value="iso-8859-1"/>
        <param name="fontFace" value="System"/>
        <param name="fontColor" value="yellow"/>
        <param name="backgroundColor" value="blue"/>
      </text>
    </par>
  </body>
</smil>
```

This example shows a `<text>` element that includes synchronized text streams within the SMIL file.

Testing Procedure:

1. Enabled caption preference in player, if present
2. Play file with captions
3. Check whether captions are displayed

Technique H95: Using the Track Element to Provide Captions[19]

Example 1: Captions in one language

A `video` element for a video in the English language with an English caption track. The captions are provided in the WebVTT format.

```
<video poster="myvideo.png" controls>
        <source src="myvideo.mp4" srclang="en" type="video/mp4">
        <track src="myvideo_en.vtt" kind="captions" srclang="en" label="English">
                </video>
```

Example 2: Captions in multiple languages

A `video` element for a video in the English language with an English caption track. The captions are provided in the WebVTT format.

```
  <video poster="myvideo.png" controls>
       <source src="myvideo.mp4" srclang="en" type="video/mp4">
       <source src="myvideo.webm" srclang="fr" type="video/webm">
       <track src="myvideo_en.vtt" kind="captions" srclang="en" label="English">
       <track src="myvideo_fr.ttml" kind="captions" srclang="fr" label="French">
                </video>
```

Testing Procedure

For each video element used to play a video:

[19] https://www.w3.org/WAI/WCAG21/Techniques/html/H95.html

1. Check that the video contains a track element of kind captions in the language of the video.

Technique SL16: Providing Script-Embedded Text Captions for Mediaelement Content[20]

Example: MediaElement handles MarkerReached, displays marker text in existing TextBox
This example has a UI definition in XAML and interaction logic in C#. The following is the basic UI in XAML. This example is deliberately simple and does not include `AutomationProperties` for identification or user instructions. The most relevant part of this example is that the Silverlight author declares a handler for the event MarkerReached. This event fires potentially hundreds of times, once for each caption in the stream. Each time the event fires, the event handler runs and adds the text to the dedicated `TextBox` in the user interface.

```
<UserControl x:Class="MediaTimelineMarkers.MainPage"
   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
   <StackPanel x:Name="LayoutRoot" Background="White">
       <MediaElement MarkerReached="OnMarkerReached"
       HorizontalAlignment="Left"
       Source="/spacetime.wmv"
       Width="300" Height="200" />
       <ScrollViewer>
           <TextBox Name="captionText" Height="40"
           IsReadOnly="true" AcceptsReturn="true"/>
       </ScrollViewer>
   </StackPanel>
 </UserControl>

private void OnMarkerReached(object sender, TimelineMarkerRoutedEventArgs e)
{
   captionText.Focus();
   captionText.SelectedText = e.Marker.Text.ToString() + "\n";
}
```

Testing Procedure
1. Using a browser that supports Silverlight, open an HTML page that references a Silverlight application through an object tag. The application plays media that is expected to have text captioning.
2. Check that a text area in the user interface shows captions for the media.

1.2.3 Audio Description or Media Alternative (Pre-Recorded) (Level A) – This Success Criterion aims to make visual information in a synchronized media presentation accessible to those who are blind or visually impaired. There are two approaches highlighted by WCAG-EM for addressing audio descriptions or media alternatives on a web page within a target site.

Technique G69: Providing an Alternative for Time Based Media[21]

Technique G58: Placing a link to the alternative for time-based media immediately next to the non-text content[22]

Example 1: An .MOV Document in an HTML Document
Code on a page called "Olympic_Sports.htm"

```
 <a name="Olympic_Wrestling"></a>
  <p><a href="http://www.example.com/movies/olympic_wrestling.mov">Olympic Wrestling
movie</a>,
  <a href="http://www.example.com/transcripts/olympic_wrestling_transcript.htm">Olympic
  Wrestling collated Transcript</a></p>
```

Example 2: The link back to the .MOV Document in an HTML Document
Code on the page olympic_wrestling_transcript.htm

[20] https://www.w3.org/WAI/WCAG21/Techniques/silverlight/SL16.html
[21] https://www.w3.org/WAI/WCAG21/Techniques/general/G69.html
[22] https://www.w3.org/WAI/WCAG21/Techniques/general/G58.html

```
  <p>Sports announcer 1: This is a great battle tonight between England's "Will Johnson" and
  "Theodore Derringo" from Argentina</p>
  <p>Scenery: There is a mat set out in the middle of the stadium with 500 people in the
  stands...</p>
  <p> ...more dialogue...<p>
  <p> ...more scenery...</p>
  <p> ...etc...</p>
  <p>Sports announcer 2: And that is all for tonight, thank you for joining us tonight where
  Will Johnson is the new Gold Medalist.
  <a href="../movies/Olympic_Sports.htm#Olympic_Wrestling>Return to Movie page</a> </p>
```

Testing Procedure

1. Check for the presence of a link immediately before or after the non-text content.
2. Check that it is a valid link that points directly to the collated document of this particular synchronized media.
3. Check for the availability of a link or back function to get the user back to the original location of the synchronized media content.

Technique SL17: Providing Static Alternative Content for Silverlight Media Playing in a MediaElement[23]

Example

{See Above}

Testing Procedure

{See Above}

Linking to the alternative for time-based media using one of the following techniques:

Technique H53: Using the body of the object element[24]

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>
```

Example 3: The image object has content that provides a brief description of the function of the image

```
<object data="companylogo.gif" type="image/gif">
  <p>Company Name</p>
</object>
```

Example 4

This example takes advantage of the fact the object elements may be nested to provide for alternative representations of information.

```
<object classid="java:Press.class" width="500" height="500">
  <object data="Pressure.mpeg" type="video/mpeg">
    <object data="Pressure.gif" type="image/gif">
      As temperature increases, the molecules in the balloon...
    </object>
  </object>
</object>
```

Testing Procedure

1. Check that the body of each object element contains a text alternative for the object.

[23] https://www.w3.org/WAI/WCAG21/Techniques/silverlight/SL17.html
[24] https://www.w3.org/WAI/WCAG21/Techniques/html/H53.html

Technique G78: Providing A Second, User-Selectable, Audio Track That Includes Audio Descriptions[25]

Examples

- A travelogue of the northeast has additional audio description added during the gaps in the dialogue to let listeners who are blind know what the person is talking about at any point in time.
- A video shows a woodpecker carving a nest in a tree. A button within the content allows users to turn the audio description track on or off.
- A lecture has audio description added whenever the instructor says things like "and *this* is the one that is most important." The audio descriptions lets listeners who can not see the video know what "this" is.
- A movie file has two audio tracks, one of which includes audio description. Users can choose either one when listening to the movie by selecting the appropriate track in their media player.

Testing Procedure

1. Check that the ability exists to turn on the audio track that includes audio descriptions. For example, by using a control within the content itself or by selecting a control or preference in the media player or operating system.
2. Listen to the synchronized media
3. Check to see if gaps in dialogue are used to convey important information regarding visual content

Technique G78: Providing a second, user-selectable, audio track that includes audio descriptions **AND** Technique SL1: Accessing Alternate Audio Tracks in Silverlight Media[26]

Examples for G78

{See Above}

Testing Procedure for G78

{See Above}

Examples for SL1: Changing AudioStreamIndex

This example has a UI definition in XAML and interaction logic in C#. In addition to the typical Play/Pause/Stop controls, this interface includes a Play Full-Description Audio button. Activating the button invokes a function that swaps the audio channels and plays an alternative synchronized audio channel that contains a composite full-description audio track.

The following is the basic UI in XAML. This example is deliberately simple and does not include AutomationProperties. Audio streams are identified by an index in a collection.

```
<Grid x:Name="LayoutRoot" Background="White">
     <Grid.ColumnDefinitions>
         <ColumnDefinition Width="*" />
         <ColumnDefinition Width="*" />
         <ColumnDefinition Width="*"/>
     </Grid.ColumnDefinitions>
     <Grid.RowDefinitions>
         <RowDefinition Height="*" />
         <RowDefinition Height="Auto" />
         <RowDefinition Height="20" />
     </Grid.RowDefinitions>
     <MediaElement x:Name="media" Source="/combined.wmv"
        Width="300" Height="300"
        Grid.Column="0" Grid.Row="0" Grid.ColumnSpan="3"
        AutoPlay="false"
     />
     <Button Click="StopMedia"
  Grid.Column="0" Grid.Row="1" Content="Stop" />
     <Button Click="PauseMedia"
  Grid.Column="1" Grid.Row="1" Content="Pause" />
     <Button Click="PlayMedia"
```

[25] https://www.w3.org/WAI/WCAG21/Techniques/general/G78.html
[26] https://www.w3.org/WAI/WCAG21/Techniques/silverlight/SL1.html

```
     Grid.Column="2" Grid.Row="1" Content="Play" />
        <Button Name="AltAudioBtn" Grid.Row="2" HorizontalAlignment="Left" Grid.ColumnSpan="2"
        Click="AltAudioBtn_Click">Play Full-Description Audio</Button>
    </Grid>
The following is the C# logic.
        private void AltAudioBtn_Click(object sender, RoutedEventArgs e)
        {
            if (media.AudioStreamCount > 1)
            {
                if (media.AudioStreamIndex == 1)
                {
                    media.AudioStreamIndex = 0;
                    (sender as Button).Content = "Play full-description audio";
                } else {
                    media.AudioStreamIndex = 1;
                   (sender as Button).Content = "Play default audio";
                }
            }
            else
            {
                (sender as Control).IsEnabled = false;
            }
        }
        private void StopMedia(object sender, RoutedEventArgs e)
        {
            media.Stop();
        }
        private void PauseMedia(object sender, RoutedEventArgs e)
        {
            media.Pause();
        }
        private void PlayMedia(object sender, RoutedEventArgs e)
        {
            media.Play();
        }
```

Testing Procedure for SL1

1. Open the HTML page for a Silverlight application, where that application plays media and the media is expected to support an alternate audio track for the video.
2. Verify that the application user interface presents a control that enables the user to cause the media to play with an alternate audio track.
3. Activate that control. Verify that the audio portion of the media player output as played through the computer's audio system is now playing the alternate audio track

Technique G173: Providing a version of a movie with audio descriptions[27]

Technique SM6: Providing audio description in SMIL 1.0[28]

Example 1: SMIL 1.0 audio description sample for QuickTime player

```
<?xml version="1.0" encoding="UTF-8"?>
<smil xmlns:qt="http://www.apple.com/quicktime/resources/smilextensions"
  xmlns="https://www.w3.org/TR/REC-smil" qt:time-slider="true">
  <head>
    <layout>
      <root-layout background-color="black" height="266" width="320"/>
      <region id="videoregion" background-color="black" top="26" left="0"
      height="144" width="320"/>
    </layout>
  </head>
  <body>
    <par>
      <video dur="0:01:20.00" region="videoregion" src="salesdemo.mov"
      alt="Sales Demo"/>
      <audio dur="0:01:20.00" src="salesdemo_ad.mp3"
```

[27] https://www.w3.org/WAI/WCAG21/Techniques/general/G173.html
[28] https://www.w3.org/WAI/WCAG21/Techniques/smil/SM6.html

```
      alt="Sales Demo Audio Description"/>
    </par>
  </body>
</smil>
```

Example 2: SMIL 1.0 audio description sample for RealTime player

```
<?xml version="1.0" encoding="UTF-8"?>
<smil xmlns="https://www.w3.org/TR/REC-smil">
  <head>
    <layout>
      <root-layout background-color="black" height="266" width="320"/>
      <region id="videoregion" background-color="black" top="26" left="0"
      height="144" width="320"/>
    </layout>
  </head>
  <body>
    <par>
      <video src="salesdemo.mov" region="videoregion" title="Sales Demo"
      alt="Sales Demo"/>
      <audio src="salesdemo_ad.mp3" title="audio description"
      alt="Sales Demo Audio Description"/>
    </par>
  </body>
</smil>
```

Testing Procedure

1. Find method for turning on audio description from content/player (unless it is always played by default)
2. Play file with audio description
3. Check whether audio description is played

Technique SM7: Providing audio description in SMIL 2.0[29]

Example 1: SMIL 2.0 audio description sample for RealMedia player

```
<smil xmlns="https://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <root-layout backgroundColor="black" height="266" width="320"/>
      <region id="video" backgroundColor="black" top="26" left="0"
      height="144" width="320"/>
    </layout>
  </head>
  <body>
    <par>
      <video src="salesdemo.mpg" region="video" title="Sales Demo"
      alt="Sales Demo"/>
      <audio src="salesdemo_ad.mp3" begin="33.71s" title="audio description"
      alt="Sales Demo Audio Description"/>
    </par>
  </body>
</smil>
```

The example shows a `<par>` segment containing an `<audio>` and a `<video>` tag. The audio stream is not started immediately.

Testing Procedure

1. Find method for turning on audio description from content/player (unless it is always played by default)
2. Play file with audio description
3. Check whether audio description is played

Technique SL1: Accessing Alternate Audio Tracks in Silverlight Media

[29] https://www.w3.org/WAI/WCAG21/Techniques/smil/SM7.html

Examples
{See Above}

Testing Procedure
{See Above}

Technique G8: Providing a movie with extended audio descriptions[30]

Example 1
An alternate version of an online video of a family escaping from a burning building: there is a continuous dialogue between the husband and wife about where the children are. Meanwhile, in the background, a wall caves in. This is important information in the story because it will block their exit from that part of the building. The video track halts (same frame is repeated) while a narrator gives the details about the wall falling and the video continues.

Example 2
A training film has narrative that runs almost continuously throughout. An alternate version is available for people who have difficulty viewing the video portion. The alternate version freezes the video and provides audio description of key information.

Testing Procedure
1. Open the version of the movie that includes extended audio descriptions.
2. Check that the video halts for extended audio description when there is not enough space to include necessary narration between the natural dialogue.
3. Check that the necessary information is in the audio description.
4. If the alternate version(s) are on a separate page, check for the availability of link(s) to allow the user to get to the other versions.

Technique G203: Using a static text alternative to describe a talking head video[31]

Example 1: A video of a CEO speaking to shareholders
A CEO is speaking to shareholders from their office. The video has a title page at the beginning of the video giving the date. When the speaker begins, there is a strip of text at the bottom of the video saying "Jane Doe, President of XYZ Cooperation". At the end of the video are title credits that say "produced by the Honest TV Productions Ltd." As an alternative, there is a paragraph below the video which is associated with the video file using aria-describedby which says: "July 22, 2011, Jane Doe, President of XYZ cooperation, speaking from her office. Video produced by the Honest TV Productions Ltd."

Testing Procedure
1. Check that there is no important time-based information in the video track
2. Check that the programmatically associated description of the media contains any context of the content that is not contained in the audio track (e.g. speaker identification, credits, context)

1.2.4 Captions (Live) (Level AA) – This Success Criterion's goal is to make real-time presentations accessible to those who are hard of hearing or deaf. The information made available through the audio track has to be included in the captions. Along with the dialogue, captions must also list the speaker's name, sound effects, and other important audio[32]. There are three, two-part techniques that are considered acceptable by WCAG-EM for addressing live captions.

[30] https://www.w3.org/WAI/WCAG21/Techniques/general/G8.html
[31] https://www.w3.org/WAI/WCAG21/Techniques/general/G203.html
[32] https://www.w3.org/WAI/WCAG21/Understanding/captions-live.html

Technique G9: Creating captions for live synchronized media[33] **AND** Technique G93: Providing open (always visible) captions[34]

Examples for G9:
Example 1
A television studio uses a real-time captioning service to create captions for its evening news online.
Example 2
A user watches an online seminar on their mobile device, including captioning provided through the use of Communication Access Real-time Translation (CART). The captions provided also benefit in-person participants who need captioning and can view the information on their own device.

Testing Procedure for G9:
1. Check that a procedure and policy are in place to ensure that captions are delivered in real-time.

Examples for G93:
{See Above}

Testing Procedure for G93:
{See Above}

Technique G9: Creating captions for live synchronized media **AND** Technique G87: Providing closed captions[35]

Examples for G9:
{See Above}

Testing Procedure for G9:
{See Above}

Examples for G87:
{See Above}
Testing Procedure for G87:
{See Above}

Technique G9: Creating captions for live synchronized media

Technique SM11: Providing captions through synchronized text streams in SMIL 1.0[36]

Examples
{See Above}

Testing Procedure
{See Above}

Technique SM12: Providing captions through synchronized text streams in SMIL 2.0[37] **AND** Technique G87: Providing closed captions

Examples for SM12:
{See Above}
Testing Procedure for SM12:

[33] https://www.w3.org/WAI/WCAG21/Techniques/general/G9.html
[34] https://www.w3.org/WAI/WCAG21/Techniques/general/G93.html
[35] https://www.w3.org/WAI/WCAG21/Techniques/general/G87.html
[36] https://www.w3.org/WAI/WCAG21/Techniques/smil/SM11.html
[37] https://www.w3.org/WAI/WCAG21/Techniques/smil/SM12.html

{See Above} ***Examples for G87:*** {See Above} ***Testing Procedure for G87:*** {See Above}
Technique SM11: Providing captions through synchronized text streams in SMIL 1.0[38] ***Examples*** {See Above} ***Testing Procedure*** {See Above}
Technique SM12: Providing captions through synchronized text streams in SMIL 2.0 ***Examples*** {See Above} ***Testing Procedure*** {See Above}

1.2.5 Audio Description (Pre-Recorded) (Level AA)[39] – This Success Criterion aimed to make visual information in a synchronized media presentation accessible to those who are blind or visually impaired. When the video portion is unavailable, the audio description should add the necessary information to the presentation's audio section. When there are pauses in the dialogue, audio description should fill them with details regarding crucial acts, characters, scene changes, and on-screen text that are not spoken or discussed in the main soundtrack.

Technique G78: Providing A Second, User-Selectable, Audio Track That Includes Audio Descriptions ***Examples*** {See Above} ***Testing Procedure*** {See Above}
Technique G78: Providing A Second, User-Selectable, Audio Track That Includes Audio Descriptions **AND** Technique SL1: Accessing Alternate Audio Tracks in Silverlight Media ***Examples for G78:*** {See Above} ***Testing Procedure for G78:*** {See Above} ***Examples for SL1:*** {See Above}

[38] https://www.w3.org/WAI/WCAG21/Techniques/smil/SM11.html
[39] https://www.w3.org/WAI/WCAG21/Understanding/audio-description-prerecorded.html

Testing Procedure for SL1:
{See Above}

Technique G173: Providing a version of a movie with audio descriptions

Technique SM6: Providing audio description in SMIL 1.0

Examples
{See Above}

Testing Procedure
{See Above}

Technique SM7: Providing audio description in SMIL 2.0

Examples
{See Above}

Testing Procedure
{See Above}

Technique G8: Providing a movie with extended audio descriptions[40]

Technique SM1: Adding extended audio description in SMIL 1.0[41]

Example 1: SMIL 1.0 Video with audio descriptions that pause the main media in 4 locations to allow extended audio description

```
<?xml version="1.0" encoding="UTF-8"?>
<smil xmlns:qt="http://www.apple.com/quicktime/resources/smilextensions"
xmlns="https://www.w3.org/TR/REC-smil" qt:time-slider="true">
  <head>
    <layout>
      <root-layout background-color="black" height="266" width="320"/>
      <region id="videoregion" background-color="black" top="26" left="0"
      height="144" width="320"/>
    </layout>
  </head>
  <body>
  <par>
   <seq>
     <par>
       <video src="video.rm" region="videoregion" clip-begin="0s" clip-end="5.4"
       dur="8.7" fill="freeze" alt="videoalt"/>
       <audio src="no1.wav" begin="5.4" alt="audio alt"/>
     </par>
     <par>
       <video src="video.rm" region="videoregion" clip-begin="5.4" clip-end="24.1"
       dur="20.3" fill="freeze" alt="videoalt"/>
       <audio src="no2.wav" begin="18.7" alt="audio alt"/>
     </par>
     <par>
       <video src="video.rm" region="videoregion" clip-begin="24.1" clip-end="29.6"
       dur="7.7" fill="freeze" alt="videoalt"/>
       <audio src="no3.wav" begin="5.5" alt="audio alt"/>
     </par>
     <par>
       <video src="video.rm" region="videoregion" clip-begin="29.6" clip-end="34.5"
       dur="5.7" fill="freeze" alt="videoalt"/>
       <audio src="no4.wav" begin="4.9" alt="audio alt"/>
     </par>
     <par>
       <video src="video.rm" region="videoregion" clip-begin="77.4" alt="video alt"/>
```

[40] https://www.w3.org/WAI/WCAG21/Techniques/general/G8.html
[41] https://www.w3.org/WAI/WCAG21/Techniques/smil/SM1.html

```
      </par>
    </seq>
  </par>
</body>
</smil>
```

Testing Procedure

1. Play file with extended audio descriptions
2. Play file with audio description
3. Check whether video freezes in places and plays extended audio description

Technique SM2: Adding extended audio description in SMIL 2.0[42]

Example 1: Video with extended audio description.

```
<smil xmlns="https://www.w3.org/2001/SMIL20/Language">
<head>
<layout>
<root-layout backgroundColor="black" height="266" width="320"/>
<region id="video" backgroundColor="black" top="26" left="0"
height="144" width="320"/>
</layout>
</head>
<body>
<excl>
<priorityClass peers="pause">
<video src="movie.rm" region="video" title="video" alt="video" />
<audio src="desc1.rm" begin="12.85s" alt="Description 1" />
<audio src="desc2.rm" begin="33.71s" alt="Description 2" />
<audio src="desc3.rm" begin="42.65s" alt="Description 3" />
<audio src="desc4.rm" begin="59.80s" alt="Description 4" />
</priorityClass>
</excl>
</body>
     </smil>
```

Testing Procedure

1. Play file with extended audio description
2. Check whether video freezes in places and plays extended audio description

Technique G203: Using a static text alternative to describe a talking head video

Examples

{See Above}

Testing Procedure

{See Above}

1.3 Adaptable[43] **–** This section requires that the pages within the target site have the capacity to display content that may be shown in a variety of ways (for instance, a simplified layout) without losing any of its structure or information. There are six criteria points that had to be addressed in order to reach level AA conformance.

[42] https://www.w3.org/WAI/WCAG21/Techniques/smil/SM2.html

[43] https://www.w3.org/WAI/WCAG21/quickref/?showtechniques=131%2C132%2C133%2C134%2C141%2C142%2C143%2C144%2C145%2C1410%2C1411%2C1412#adaptable

1.3.1 Info and Relationships (Level A)[44] – This Success Criterion's goal is to make sure that when the presentation format changes, information and connections that are implied by visual or auditory formatting are still available. For instance, the presentation format could be altered when a screen reader reads the information or when a user style sheet takes the place of the author's style sheet. The target website might either possess the technology that would provide the semantic structure to make information and relationships conveyed through presentation programmatically determinable through one of eleven possible features or it does not and would have some type of workaround would have to be built in.

Situation A: The technology provides semantic structure to make information and relationships conveyed through presentation programmatically determinable:

Technique ARIA11: Using ARIA landmarks to identify regions of a page[45]

Example 1: Simple landmarks
The following example shows how landmarks might be added to an HTML4 or XHTML 1.0 document:

```
<div id="header" role="banner">A banner image and introductory title</div>
<div id="sitelookup" role="search">....</div>
<div id="nav" role="navigation">...a list of links here ... </div>
<div id="content" role="main"> ... Ottawa is the capital of Canada ...</div>
<div id="rightsideadvert" role="complementary">....an advertisement here...</div>
<div id="footer" role="contentinfo">(c)The Freedom Company, 123 Freedom Way, Helpville, USA</div>
```

Example 2: Multiple landmarks of the same type and aria-labelledby
The following example shows a best practice of how landmarks might be added to an HTML4 or XHTML 1.0 document in situations where there are more than two of the same type of landmark on the same page. For instance, if a navigation role is used multiple times on a Web page, each instance may have a unique label specified using aria-labelledby:

```
<div id="leftnav" role="navigation" aria-labelledby="leftnavheading">
<h2 id="leftnavheading">Institutional Links</h2>
<ul><li>...a list of links here ...</li> </ul></div>
<div id="rightnav" role="navigation" aria-labelledby="rightnavheading">
<h2 id="rightnavheading">Related topics</h2>
<ul><li>...a list of links here ...</li></ul></div>
```

Example 3: Multiple landmarks of the same type and aria-label
The following example shows a best practice of how landmarks might be added to an HTML4 or XHTML 1.0 document in situations where there are more than two of the same type of landmark on the same page, and there is no existing text on the page that can be referenced as the label.

```
<div id="leftnav" role="navigation" aria-label="Primary">
<ul><li>...a list of links here ...</li></ul> </div>
<div id="rightnav" role="navigation" aria-label="Secondary">
<ul><li>...a list of links here ...</li> </ul></div>
```

Example 4: Search form
The following example shows a search form with a "search" landmark. The search role typically goes on the form element or a div surrounding the form.

```
<form role="search">
<label for="s6">search</label><input id="s6" type="text" size="20">
...
</form>
```

Testing Procedure
1. Examine each element with a landmark role.

[44] https://www.w3.org/WAI/WCAG21/Understanding/info-and-relationships.html
[45] https://www.w3.org/WAI/WCAG21/Techniques/aria/ARIA11.html

2. Examine whether the landmark role attribute is applied to the section of the page that corresponds with that role. (i.e., the "navigation" role is applied to a navigation section, the "main" role is applied to where the main content begins.)

Technique ARIA12: Using role=heading to identify headings[46]

Example 1: Simple headings

This example demonstrates how to implement simple headings using role="heading" when retrofitting a legacy site where scripts depend on the existing element hierarchy or the level is unknown. For example, web content which is syndicated from various sources may be constructed without knowledge of what the final presentation will be.

```
<div role="heading">Global News items</div>
... a list of global news with editorial comment....

<div role="heading">Local News items</div>
... a list of local news, with editorial comment ...
```

Example 2: Additional heading levels

This example demonstrates how to implement a level 7 heading using `role="heading"` and the `aria-level` attribute. Since HTML only supports headings through level 6, there is no native element to provide these semantics.

```
...
<h5>Fruit Trees</h5>
...
<h6>Apples</h6>
<p>Apples grow on trees in areas known as orchards...</p>
...
<div role="heading" aria-level="7">Jonagold/div>
<p>Jonagold is a cross between the Golden Delicious and Jonathan varieties...</p>
```

Testing Procedure

1. Examine each element with the attribute `role="heading".`
2. Determine whether the content of the element is appropriate as a heading.
3. If the element has an `aria-level` attribute, determine whether the value is the appropriate hierarchical level.

Technique ARIA13: Using aria-labelledby to name regions and landmarks[47]

Example 1: Identify a landmark with on-page text

Below is an example of aria-labelledby used on a complementary Landmark. The region of the document to which the heading pertains could be marked with the aria-labelledby property containing the value of the id for the header.

```
<div role="complementary" aria-labelledby="hdr1">
  <h1 id="hdr1">
    Top News Stories
    ...
  </h1>
</div>
```

Example 2: Identification for Application landmarks

The following code snippet for application landmarks with static prose. If you have a regional landmark of type application and static descriptive text is available, then on the application landmark, include an aria-describedby reference to associate the application and the static text as shown here:

```
<div role="application" aria-labelledby="p123" aria-describedby="info">
  <h1 id="p123">Calendar<h1>
  <p id="info">
    This calendar shows the game schedule for the Boston Red Sox.
  </p>
```

[46] https://www.w3.org/WAI/WCAG21/Techniques/aria/ARIA12.html
[47] https://www.w3.org/WAI/WCAG21/Techniques/aria/ARIA13.html

```
<div role="grid">
  ...
</div>
```

Testing Procedure

1. Examine each element with attribute role=region or with a landmark role, where an aria-labelledby attribute is also present.
2. Check that the value of the aria-labelledby attribute is the id of an element on the page.
3. Check that the text of the element with that id accurately labels the section of the page.

Technique ARIA16: Using aria-labelledby to provide a name for user interface controls[48]

Example 1: Labelling a simple text field

The following is an example of aria-labelledby used on a simple text field to provide a label in a situation where there is no text available for a dedicated label but there is other text on the page that can be used to accurately label the control.

```
<input name="searchtxt" type="text" aria-labelledby="searchbtn">
<input name="searchbtn" id="searchbtn" type="submit" value="Search">
```

Example 2: Labelling a slider

Below is an example of `aria-labelledby` used to provide a label for a slider control. In this case the label text is selected from within a longer adjacent text string. Please note that this example is simplified to show only the labeling relationship; authors implementing custom controls also need to ensure that controls meet other success criteria.

```
<p>Please select the <span id="mysldr-lbl">number of days for your trip</span></p>
<div id="mysldr" role="slider" aria-labelledby="mysldr-lbl"></div>
```

Example 3: A label from multiple sources

The following example of `aria-labelledby` with multiple references uses the `label` element. For additional detail on concatenating multiple sources of information into a label with `aria-labelledby`, please view the technique Using ARIA labelledby to concatenate a label from several text nodes.

```
<label id="l1" for="f3">Notify me</label>
<select name="amt" id="f3" aria-labelledby="l1 f3 l2">
  <option value="1">1</option>
  <option value="2">2</option>
</select>
<span id="l2" tabindex="-1">days in advance</span>
```

Testing Procedure

For each user interface control element where an aria-labelledby attribute is present:

1. Check that the value of the aria-labelledby attribute is the id of an element or a space separated list of ids on the web page.
2. Check that the text of the referenced element or elements accurately labels the user interface control.

Technique ARIA17: Using grouping roles to identify related form controls[49]

Example 1: Social Security Number

Social security number fields which are 9 digits long and broken up into 3 segments can be grouped using `role="group"`.

```
<div role="group" aria-labelledby="ssn1">
   <span id="ssn1">Social Security#</span>
   <span style="color: #D90D0D;"> * </span>
   <input size="3" type="text" aria-required="true" title="First 3 digits" />-
   <input size="2" type="text" aria-required="true" title="Next 2 digits" />-
```

[48] https://www.w3.org/WAI/WCAG21/Techniques/aria/ARIA16.html
[49] https://www.w3.org/WAI/WCAG21/Techniques/aria/ARIA17.html

```
   <input size="4" type="text" aria-required="true" title="Last 4 digits" />
</div>
```

Example 2: Identifying radio groups
This example demonstrates use `role=radiogroup`. Note also that the radio buttons are custom controls with `role=radio`. (But the script to make the span actually work like radio buttons is not included in this example.) One may optionally employ CSS to place a border around groups of such fields to visually reinforce the group relationship. The CSS properties are available below the form.

```
<h3>Set Alerts for your Account</h3>
  <div role="radiogroup" aria-labelledby="alert1">
    <p id="alert1">Send an alert when balance exceeds $ 3,000</p>
    <div>
      <span role="radio" aria-labelledby="a1r1" name="a1radio"></span>
      <span id="a1r1">Yes</span>
    </div>
    <div>
      <span role="radio" aria-labelledby="a1r2" name="a1radio"></span>
      <span id="a1r2">No</span>
    </div>
  </div>
  <div role="radiogroup" aria-labelledby="alert2">
    <p id="alert2">Send an alert when a charge exceeds $ 250</p>
    <div>
      <span role="radio" aria-labelledby="a2r1" name="a2radio"></span>
      <span id="a2r1">Yes</span>
    </div>
    <div>
      <span role="radio" aria-labelledby="a2r2" name="a2radio"></span>
      <span id="a2r2">No</span>
    </div>
  </div>
  <p><input type="submit" value="Continue" id="continue_btn" name="continue_btn" /></p>
```

Related CSS Style Definition to place a border around the group of fields :

```
div[role=radiogroup] {
  border: black thin solid;
}
```

Testing Procedure
For groups of related controls where the individual labels for each control do not provide a sufficient description, and an additional group level description is needed:
1. Check that the group of logically related input or select elements are contained within an element with `role=group`.
2. Check that this group has an accessible name defined using `aria-label` or `aria-labelledby`.

Technique ARIA20: Using the region role to identify a region of the page[50]

Example 1: Region on a news website
A section on the home page of a news website that contains a poll that changes every week is marked up with `role="region"`. The h3 text above the form is referenced as the region's name using `aria-labelledby`.

```
<div role="region" aria-labelledby="pollhead">
<h3 id="pollhead">This week's Poll</h3>
<form method="post" action="#">
  <fieldset>
    <legend>Do you believe the tax code needs to be overhauled?</legend>
    <input type="radio" id="r1" name="poll" />
    <label for="r1">No, it's fine the way it is</label>
    <input type="radio" id="r2" name="poll" />
    <label for="r2">Yes, the wealthy need to pay more</label>
    <input type="radio" id="r3" name="poll" />
```

[50] https://www.w3.org/WAI/WCAG21/Techniques/aria/ARIA20.html

```
    <label for="r3">Yes, we need to close corporate loopholes</label>
    <input type="radio" id="r4" name="poll" />
    <label for="r4">Changes should be made across the board</label>
  </fieldset>
</form>
<a href="results.php">See Poll Results</a>
</div>
```

Example 2: Identifying a region on a banking site

A user can expand links on a bank website after logging in to see details of term deposit accounts. The details are within a span marked up with region role. The heading for the region has role=heading and is included in the `aria-labelledby` that names the region.

```
<ol>
       <li><a    id="l1"    href="#"    aria-expanded="false"    title="Show    details"    aria-
controls="block1" >John Henry's Account</a><img src="images/panel_expand.gif" alt="" />
               <div  id="block1"  class="nowHidden"  tabindex="-1"  aria-labelledby="l1  cd1"
role="region"><span id="cd1" role="heading" aria-level="3">Certificate of  Deposit:</span>
               <table>
                      <tr><th scope="row">Account:</th> <td>25163522</td></tr>
                      <tr><th scope="row">Start date:</th> <td>February 1, 2014</td></tr>
                      <tr><th scope="row">Maturity date:</th><td>February 1, 2016</td></tr>
                      <tr><th scope="row">Deposit Amount:</th> <td>$ 3,000.00</td></tr>
                      <tr><th scope="row">Maturity Amount:</th> <td>$ 3,072.43</td></tr>
               </table>
               </div>
        </li>
 </ol>
```

Example 3: Identifying a portlet with a generic region

This example shows how a generic region landmark might be added to a weather portlet. There is no existing text on the page that can be referenced as the label, so it is labelled with `aria-label`.

```
<div role="region" aria-label="weather portlet">
      ...
</div>
```

Testing Procedure

For each section marked up with `role="region"`:

1. Examine the content and ensure that it is important enough to have an independent landmark
2. Ensure that a standard landmark role is not appropriate for this content
3. Check that the region has a programmatically determined name

Technique G115: Using semantic elements to markup structure[51] **AND** Technique H49: Using semantic markup to mark emphasized or special text[52]

Examples for G115:

Example 1

A paragraph contains a hyperlink to another page. The hyperlink is marked up using the `<a>` element.

```
       <p> Do you want to try our new tool yourself? A free
               demonstration version is available in our
               <a href="download.html">download section </a></p>
```

Example 2

A page about the history of marriage uses a quotation from Jane Austen's novel, Pride and Prejudice, as an example. The reference to the book is marked up using the cite element and the quotation itself is marked up using the blockquote element.

[51] https://www.w3.org/WAI/WCAG21/Techniques/general/G115.html
[52] https://www.w3.org/WAI/WCAG21/Techniques/html/H49.html

```
        <p>Marriage is considered a logical step for a bachelor,
                as can be seen in the first chapter of the novel
                <cite>Pride and Prejudice</cite>:</p>
                <blockquote>
                <p>It is a truth universally acknowledged, that a single man in
                possession of a good fortune, must be in want of a wife.</p>
                <p>However little known the feelings or views of such a man may
                be on his first entering a neighbourhood, this truth is so well
                fixed in the minds of the surrounding families, that he is considered
                the rightful property of some one or other of their daughters.</p>
                </blockquote>
```

Example 3

A car manual explains how to start the engine. The instructions include a warning to make sure the gear is in neutral. The author feels the warning is so important that it should be emphasized so the warning is marked up using the strong element.

```
              <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>
```

Example 4

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

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

Example 5: Using highlighting and background color to visually and semantically identify important information.

```
              <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>
```

Testing Procedure for G115:

1. Check if there are parts of the content that have a semantic function.
2. For each part that has a semantic function, if corresponding semantic markup exists in the technology, check that the content has been marked up using that semantic markup.

Examples for H49:

Example 1: Using the em and strong elements to emphasize text

The `em` and `strong` elements are designed to indicate structural emphasis that may be rendered in a variety of ways (font style changes, speech inflection changes, etc.).

```
 ... What she <em>really</em> meant to say is, &quot;This is not OK,
 it is <strong>excellent</strong>&quot;! ...
```

Example 2: Using the blockquote element to mark up long quotations from another source

This example also demonstrates the use of the cite element to specify a reference.

```
<p>The following is an excerpt from the <cite>The Story Of My Life</cite>
   by Helen Keller:</p>
<blockquote>
  <p>Even in the days before my teacher came, I used to feel along the square stiff
```

```
     boxwood hedges, and, guided by the sense of smell, would find the first violets
     and lilies. There, too, after a fit of temper, I went to find comfort and to hide
     my hot face in the cool leaves and grass.</p>
</blockquote>
```

Example 3: Using the q element to mark up a shorter quotation from another source

Quotation marks aren't manually added to the quote because they are added by the user agent.

```
<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>
```

Example 4: Using the sup and sub elements to markup superscripts and subscripts
The `sup` and `sub` elements must be used only to markup typographical conventions with specific meanings, not for typographical presentation for presentation's sake.

```
<p>Beth won 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>
```

Example 5: Using the code element to mark up code
This example shows use of the code element to provide visual emphasis for a CSS rule:

```
<code>
#trial {
  background-image: url(30daytrial.jpg);
  background-position: left top;
  background-repeat: no-repeat;
  padding-top: 68px;
}
</code>
```

Testing Procedure for H49:
1. Examine the content for information that is conveyed through variations in presentation of text.
2. Check that appropriate semantic markup (such as `em`, `strong`, `cite`, `blockquote`, `sub`, and `sup`) have been used to mark the text that conveys information through variations in text.

Technique G117: Using text to convey information that is conveyed by variations in presentation of text[53]

Example 1: Indicating new content with boldface and a text indicator
The following example shows a list of accessibility standards. WCAG 2.0 is new, so is indicated in bold face. To avoid conveying information solely by presentation, the word "(new)" is included after it as well.

```
<h2>Web Accessibility Guidelines</h2>
  <ul>
  <li><strong>WCAG 2.0 (New)</strong></li>
  <li>WCAG 1.0</li>
  <li>Section 508</li>
  <li>JIS X 8341-3</li>
  ...
  </ul>
```

Example 2: Font variations and explicit statements.
An on-line document has gone through multiple drafts. Insertions are underlined and deletions are struck through. At the end of the draft a "change history" lists all changes made to each draft.

Example 3: Providing an alternate way to know which words in the text have been identified by using a different font.
An on-line test requires students to write a short summary of a longer document. The summary must contain certain words from the original document. When a sentence in the original document contains a word or phrase that must be

[53] https://www.w3.org/WAI/WCAG21/Techniques/general/G117.html

used in the summary, the word or phrase is shown in a different font than the rest of the sentence. A separate section also lists all the words and phrases that must be used in the summary.

Testing Procedures
1. Find items where variations in presentation of text are used to convey information.
2. For those items, check to determine if information conveyed visually is also stated explicitly in text.

Technique G140: Separating information and structure from presentation to enable different presentations[54]

Example 1: 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.

Example 2: Tagged PDF
A PDF document consists mostly of the content embedded with formatting information. Information about the structure is provided in a separate section of the document using XML-like tags; this is called "tagged PDF". The information in these tags can be used by assistive technologies to perform meaningful structure transformations (e.g., generating a list of sections) or to support interaction with content based on structural characteristics (e.g., jumping to the start of forms).

Testing Procedures
1. Examine the encoding of a document.
2. Check that structural information and functionality are explicitly provided and is logically separated from presentational information.

Technique ARIA24: Semantically identifying a font icon with `role="img"`[55]

Example 1: Star Icon Font used as an indicator (not interactive)
In this example a star icon is used to indicate a favorite. It is not interactive and does not disappear if the user overrides the font family via CSS.

```
Author CSS
/* default class for fonts-face with icons */
.icon { font-family: 'IconFontRoleImg' !important; }

/* specific class for icon */
.icon-star-bg:before { content: "\e982"; }

HTML
- Instead of... -
<p>
   <span class="icon icon-star-bg"></span>
</p>

- Do... -

<p>
   <span class="icon icon-star-bg" role="img" aria-label="Favorite"></span>
</p>

User CSS
*:not([role="img"]) { font-family: Verdana, sans-serif !important; }
```

Example 2: Two colored / stacked star Icon Font used as an indicator

[54] https://www.w3.org/WAI/WCAG21/Techniques/general/G140.html
[55] https://www.w3.org/WAI/WCAG21/Techniques/aria/ARIA24.html

In this example a two colored star icon is created by stacking two fonts with different colors on top of each other. This way it's possible to mimic only half the star is filled. It is not interactive and does not disappear if the user overrides the font family via CSS.

Author CSS
```
/* default class for fonts-face with icons */
.icon { font-family: 'IconFontRoleImg' !important; }

/* specific classes for icons */
.icon-star-bg:before    {content: "\e982"; }
.icon-star-half:before  {content: "\e983"; }
```

HTML
```
- Instead of... -
<span class="icon-stacked">
   <span class="icon icon-star-bg grey"></span>
   <span class="icon icon-star-half yellow"></span>
</span>

- Do... -
<span class="icon-stacked" role="img" aria-label="Favorite star half filled">
   <span class="icon icon-star-bg grey" role="img" aria-hidden="true"></span>
   <span class="icon icon-star-half yellow" role="img" aria-hidden="true"></span>
</span>
```

User CSS
```
*:not([role="img"]) { font-family: Verdana, sans-serif !important; }
```

Example 3: Email Icon Font in a link WITHOUT visible text
In this example an email icon is in a link with no visible text. It does not disappear if a user overrides font family. The icon font is identified by assistive technology as a "link image" and the name "Email" (keyboard or mouse).

Author CSS
```
/* default class for fonts-face with icons */
.icon { font-family: 'IconFontRoleImg' !important; }

/* specific class for icon */
.icon-email:before { content: "\e93e"; }
```

HTML
```
- Instead of... -
<a href="email.html">
   <span class="icon icon-email"></span>
</a>

- Do... -
<a href="email.html">
   <span class="icon icon-email" role="img" aria-label="Email"></span>
</a>
```

User CSS
```
*:not([role="img"]) { font-family: Verdana, sans-serif !important; }
```

Example 4: Multiple Icon Fonts as part of another sematic element WITH visible text
This example already has a visible text label in the link to be used as an accessible name, the mail and chevron font icons must stay visible when the font family is changed. This can be done by ensuring the icons are contained in their own element and the attribute aria-hidden="true" is used so the font icons will be ignored by assistive technologies.

Author CSS
```
/* default class for fonts-face with icons */
.icon { font-family: 'IconFontRoleImg' !important; }

/* specific class for icon */
- See style declarations in HTML examples -
```

HTML

```
- Instead of... -
<style>
.icon-double-link:before   { content: "\e93e"; }
.icon-double-link:after    { content: "\e993"; }
</style>

<a href="email.html" class="icon-double-link">
   Email
</a>

- Do... -
<style>
.icon-email:before   { content: "\e93e"; }
.icon-chevron:before {  content: "\e993";  }

.icon-double-link .icon-chevron  { float: right; margin-left: 1.5rem; }
</style>

<a href="email.html" class="icon-double-link">
   <span class="icon icon-email" role="img" aria-hidden="true"></span>
   <span class="icon icon-chevron" role="img" aria-hidden="true"></span>
   Email
</a>

User CSS
*:not([role="img"]) { font-family: Verdana, sans-serif !important; }
```

Testing Procedure
For each font icon check that:
1. The element providing the font icon has `role="img"`.

Situation B: The technology in use does NOT provide the semantic structure to make the information and relationships conveyed through presentation programmatically determinable:

Technique G117: Using text to convey information that is conveyed by variations in presentation of text

Examples
{See Above}

Testing Procedure
{See Above}

Making information and relationships conveyed through presentation programmatically determinable or available in text using the following techniques:

Technique T1: Using standard text formatting conventions for paragraphs[56]

Examples
Two paragraphs. Each starts and ends with a blank line.

```
This is the first sentence in this
paragraph. Paragraphs may be long
or short.

    In this paragraph the first line is
indented. Indented and non-indented
sentences are allowed. White space within
the paragraph lines is ignored in
defining paragraphs. Only completely blank
lines are significant.
```

Testing Procedures

[56] https://www.w3.org/WAI/WCAG21/Techniques/text/T1.html

For each paragraph:

1. Check that the paragraph is preceded by exactly one blank line, or that the paragraph is the first content in the Web page
2. Check that the paragraph is followed by at least one blank line, or that the paragraph is the last content in the Web page.
3. Check that no paragraph contains any blank lines

Technique T2: Using standard text formatting conventions for lists[57]

Example 1: Unordered list

```
- unordered list item
- unordered list item
- unordered list item
```

Example 2: Numeric ordered list

```
1. Ordered list item
2. Ordered list item
3. Ordered list item
```

Example 3: Roman numeral ordered list

```
i.   Ordered list item
ii.  Ordered list item
iii. Ordered list item
iv.  Ordered list item
```

Example 4: Alphabetic ordered list

```
A) Ordered list item
B) Ordered list item
C) Ordered list item
```

Testing Procedure

For each list in the text content

1. Check that each list item is a paragraph that starts with a label
2. Check that the list contains no lines that are not list items
3. Check that all list items in a list use the same style label
4. Check that the labels in ordered lists are in sequential order
5. Check that the labels in each unordered list are the same

Technique T3: Using standard text formatting conventions for headings[58]

Example

A paragraph is followed by two blank lines, then a heading, then one blank line, then another paragraph:

```
...this is the end of paragraph 1.

The Text of the Heading

This is the beginning of paragraph 2.
```

Testing Procedure

For each heading in the content:

1. Check that each heading is preceded by two blank lines
2. Check that each heading is followed by a blank line
3. Check that no heading contains any blank lines

[57] https://www.w3.org/WAI/WCAG21/Techniques/text/T2.html
[58] https://www.w3.org/WAI/WCAG21/Techniques/text/T3.html

1.3.2 Meaningful Sequence (Level A)[59] – This Success Criterion's goal is to make sure that when the presentation format changes, information and connections that are implied by visual or auditory formatting are still available. For instance, the presentation format can be altered when a screen reader reads the information or when a user style sheet takes the place of the author's style sheet. It's important to note that some platforms might not offer a way to programmatically determine certain sorts of data and relationships. In that situation, a text description of the data and connections is required. Asterisks (*) are used to indicate that a field is necessary. When the page is linearized, the text description should be close to the information it is describing, such as in the parent element or in the adjacent element.

Technique G57: Ordering the content in a meaningful sequence[60]

Examples

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.

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;
}
```

Testing Procedure

1. Linearize content using a standard approach for the technology (e.g., removing layout styles or running a linearization tool)
2. Check to see if the order of content yields the same meaning as the original

Marking sequences in the content as meaningful **AND** Technique G57: Ordering the content in a meaningful sequence:

[59] https://www.w3.org/WAI/WCAG21/Understanding/info-and-relationships.html
[60] https://www.w3.org/WAI/WCAG21/Techniques/general/G57.html

Technique H34: Using a Unicode right-to-left mark (RLM) or left-to-right mark (LRM) to mix text direction inline[61]

Example
This example shows an Arabic phrase in the middle of an English sentence. The exclamation point is part of the Arabic phrase and should appear on its left. Because it is between an Arabic and Latin character and the overall paragraph direction is LTR, the bidirectional algorithm positions the exclamation mark to the right of the Arabic phrase.

The title is "مفتاح معايير الويب!" in Arabic.

Visually-ordered ASCII version (RTL text in uppercase, LTR in lower):
the title is "HCTIWS SDRADNATS BEW!" in arabic.
Inserting a Unicode right-to-left mark in the code immediately after the exclamation mark positions it correctly when you view the displayed text (see below). You can use a character escape or the (invisible) control character to insert the right-to-left mark.

The title is "مفتاح معايير الويب!" in Arabic.

Visually-ordered ASCII version:
the title is "!HCTIWS SDRADNATS BEW" in arabic.

Testing Procedure
1. Examine the source for places where text changes direction.
2. When text changes direction, check whether neutral characters such as spaces or punctuation occur adjacent to text that is rendered in the non-default direction.
3. When check 2 is true and the HTML bidirectional algorithm would produce the wrong placement of the neutral characters, check whether the neutral characters are followed by Unicode right-to-left or left-to-right marks that cause neutral characters to be placed as part of the preceding characters.

Technique H56: Using the `dir` attribute on an inline element to resolve problems with nested directional runs[62]

Example: Defining the text direction of a nested, mixed-direction phrase, in Hebrew and English, to be right-to-left
Because the whole quote is in Hebrew, and therefore runs right to left, the text "W3C" and the comma should appear to the left of (i.e., after) the Hebrew text, like this:

The title is "פעילות הבינאום, W3C" in Hebrew.

Visually-ordered ASCII version (RTL text in uppercase, LTR in lower):
the title is "w3c ,YTIVITCA NOITAZILANOITANRETNI" in Hebrew.
The Unicode bidirection algorithm alone is insufficient to achieve the right result, and leaves the text "W3C" on the right side of the quote:

The title is "פעילות הבינאום, W3C" in Hebrew.

Visually-ordered ASCII version:
the title is "YTIVITCA NOITAZILANOITANRETNI, w3c" in hebrew.

The following markup will produce the expected result:
```
<p>The title says "<span lang="he" dir="rtl">פעילות הבינאום, W3C</span>" in Hebrew.</p>
```

Testing Procedure
1. Examine the text direction of text in the document

[61] https://www.w3.org/WAI/WCAG21/Techniques/html/H34.html
[62] https://www.w3.org/WAI/WCAG21/Techniques/html/H56.html

2. If the text direction is right-to-left, check that for the ancestor element that has a dir attribute, the attribute has the value "`rtl`"
3. If the text direction is left-to-right, check that there is no ancestor element with a dir attribute, or that for the ancestor element that has a dir attribute, the attribute has the value "`ltr`"

Technique C6: Positioning content based on structural markup[63]

Example

In this example structural markup (definition lists) have been applied to the content. CSS has been used to style the content into columnar form. Each class absolutely positions the content into columns and the margins have been set to 0 to override the default behavior of user agents to display HTML definition lists with the DD element indented.

Here is the content to be displayed:

```
<div class="box">
 <dl>
   <dt class="menu1">Products</dt>
   <dd class="item1">Telephones</dd>
   <dd class="item2">Computers</dd>
   <dd class="item3">Portable MP3 Players</dd>
   <dt class="menu2">Locations</dt>
   <dd class="item4">Idaho</dd>
   <dd class="item5">Wisconsin</dd>
   </dt>
 </dl>
</div>
```

Here is the CSS which positions and styles the above elements:

```
.item1 {
  left: 0;
  margin: 0;
  position: absolute;
  top: 7em;
}
.item2 {
  left: 0;
  margin: 0;
  position: absolute;
  top: 8em;
}
.item3 {
  left: 0;
  margin: 0;
  position: absolute;
  top: 9em;
}
.item4 {
  left: 14em;
  margin: 0;
  position: absolute;
  top: 7em;
}
.item5 {
  left: 14em;
  margin: 0;
  position: absolute;
  top: 8em;
}
.menu1 {
  background-color: #FFFFFF;
  color: #FF0000;
  font-family: sans-serif;
  font-size: 120%;
  left: 0;
```

[63] https://www.w3.org/WAI/WCAG21/Techniques/css/C6.html

```
  margin: 0;
  position: absolute;
  top: 3em;
}
.menu2 {
  background-color: #FFFFFF;
  color: #FF0000;
  font-family: sans-serif;
  font-size: 120%;
  left: 10em;
  margin: 0;
  position: absolute;
  top: 3em;
}
#box {
  left: 5em;
  position: absolute;
  top: 5em;
}
```

When style sheets are applied, the data are displayed in two columns of "Products" and "Locations." When the style sheets are not applied, the text appears in a definition list which maintains the structure and reading order.

Testing Procedure

For content which uses CSS for positioning

1. Remove the style information from the document or turn off use of style sheets in the user agent.
2. Check that the structural relations and the meaning of the content are preserved.

Technique C8: Using CSS letter-spacing to control spacing within a word[64]

Example: Separating characters in a word

The following CSS would add the equivalent of a space between each character in a level-2 heading:

```
h2 {   letter-spacing: 1em; }
So for the markup:
<h2>Museum</h2>
the rendered result might look something like:
M u s e u m
```

Testing Procedure

For each word that appears to have non-standard spacing between characters:

1. Check whether the CSS letter-spacing property is used to control spacing.

Technique C27: Making the DOM order match the visual order[65]

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.

Testing Procedure

1. Visually examine the order of the content in the Web page as it is presented to the end user.
2. Examine the elements in the DOM using a tool that allows you to see the DOM.
3. Ensure that the order of the content in the source code sections match the visual presentation of the content in the Web page. (e.g., for an English language page, the order is from top to bottom and from left to right.) "

[64] https://www.w3.org/WAI/WCAG21/Techniques/css/C8.html
[65] https://www.w3.org/WAI/WCAG21/Techniques/css/C27.html

1.3.3 Sensory Characteristics (Level A)[66] – The purpose of this criterion is to make it possible for a user agent to deliver content in a different way while still maintaining the necessary reading sequence for comprehension. According to the guidelines, it is crucial that at least one logical sequence of the content could be determined automatically. The goal is to identify that when assistive technology reads the material out of sequence, when other style sheets are used, or when other formatting modifications are made, content that does not match this Success Criterion could cause users to become confused or disoriented.

Note: Other techniques may also be sufficient if they meet the success criterion.

Technique G96: Providing textual identification of items that otherwise rely only on sensory information to be understood[67]

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
Instructions for a Web page providing on-line training state, "Use the list of links to the right with the heading, 'Class Listing' to navigate to the desired on-line course." This description provides location as well as textual clues to help find the correct list of links.

Example 3
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.

```
<table>
  <tbody>
  <tr>
  <td colspan="2">Push the lower right [Preview] button.</td>
  <td>
  <span style="background: ButtonFace; color: ButtonText; border:
  medium outset ButtonShadow;
  width: 5em; display: block; font-weight: bold; text-align: center;">
  Print</span>
  </td>
  </tr>
  <tr>
  <td>
  <span style="background: ButtonFace; color: ButtonText; border:
  medium outset ButtonShadow;
  width: 5em; display: block; font-weight: bold; text-align: center;">
  Cancel</span>
  </td>
  <td>
  <span style="background: ButtonFace; color: ButtonText; border:
  medium outset ButtonShadow;
  width: 5em; display: block; font-weight: bold; text-align: center;">
  OK</span>
  </td>
  <td>
  <span style="background: ButtonFace; color: ButtonText; border:
  medium outset ButtonShadow;
  width: 5em; display: block; font-weight: bold; text-align: center;">
  Preview</span>
  </td>
  </tr>
  </tbody>
  </table>
```

[66] https://www.w3.org/WAI/WCAG21/Understanding/meaningful-sequence.html
[67] https://www.w3.org/WAI/WCAG21/Techniques/general/G96.html

Testing Procedure Find all references in the Web page that mention the shape, size, or position of an object. For each such item: 1. Check that the reference contains additional information that allows the item to be located and identified without any knowledge of its shape, size, or relative position.

1.3.4 Orientation (Level AA)[68] – This Success Criterion's objectives are to ensure that anyone who wants to use the content may do so, even if they are unable to grasp concepts like size, shape, or orientation. Some material depended on information about the position or shape of objects that are not provided by the content's structure (for example, "round button" or "button to the right"). Because of how their assistive technologies are designed, some disabled users might be unable to discern shape or position. Additional information must be provided to clarify instructions that rely on this kind of information in order to meet this success criterion.

Note: Other techniques may also be sufficient if they meet the success criterion.

Using CSS to set the orientation to allow both landscape and portrait.
Use of show/hide controls to allow access to content in different orientations.

1.3.5 Identify Input Purpose (Level AA)[69] – The purpose of this criteria point is to make sure that content displays in the user's desired orientation (landscape or portrait). Some websites and applications restrict the screen's display orientation automatically and expect users to respond by turning their devices to conform. However, this can lead to issues. Some users have their devices fixedly mounted (e.g. on the arm of a power wheelchair). Websites and software should therefore support both orientations by not limiting the orientation. This criterion, which is concentrated on limits of orientation, does not encompass changes in content or functionality caused by changes in display size.

The input field serves a purpose identified in the Input Purposes for User Interface Components section; and the content is implemented using technologies with support for identifying the expected meaning for form input data.

1.3.6 Identify Purpose (Level AA)[70] – In order for user agents to extract and convey this purpose to users using various modalities, it must be possible to programmatically establish the purpose of a form input gathering user information. Filling out forms is made simpler, especially for those with cognitive limitations, by the ability to programmatically indicate the specific type of data required in a certain field. Users may benefit from having fields that collect specific types of information be rendered in an unambiguous, consistent, and possibly customized way for different modalities - either through defaults in their user agent, or with the help of assistive technologies. Appropriate visible labels and instruction can help users understand the purpose of form input fields.

Technique ARIA11: Using ARIA landmarks to identify regions of a page ***Examples*** {See Above} ***Testing Procedures*** {See Above}
Using microdata to markup user interface components (future link)

[68] https://www.w3.org/WAI/WCAG21/Understanding/sensory-characteristics.html
[69] https://www.w3.org/WAI/WCAG21/Understanding/orientation.html
[70] https://www.w3.org/WAI/WCAG21/Understanding/identify-input-purpose.html

1.4 Distinguishable[71] – Under WCAG, this section requires that the target site create content that is simpler for users to see and hear, distinguishing foreground from background. There are nine criteria points. The first two grant level A conformance while the remaining seven are for level AA.

1.4.1 Use of Color (Level A)[72] – This Success Criterion's goal is to make sure that all visually impaired users can access information that is communicated through color disparities – through the use of color when each color has a specific meaning. Users with color blindness might not be able to see the color if the information is presented by color differences in an image (or other non-text format). In this instance, giving the color conveys information through another visual medium guarantees that users who are color-blind can still understand the content.

Situation A: If the color of particular words, backgrounds, or other content is used to indicate information:

Technique G14: Ensuring that information conveyed by color differences is also available in text[73]

Example 1: A color-coded schedule
The schedule for sessions at a technology conference is organized into three tracks. Sessions for Track 1 are displayed over a blue background. Sessions in Track 2 are displayed over a yellow background. Sessions in Track 3 are displayed on a green background. After the name of each session is a code identifying the track in text: T1 for Track 1, T2 for Track 2, and T3 for Track 3.

Example 2: A color-coded schedule with icons
The schedule for sessions at a technology conference is organized into three tracks. Next to the title of each session is an icon consisting of a colored circle with a number in the middle showing what track it belongs to: blue circles with the number 1 represent track 1, yellow circles with the number 2 represent Track 2, and green circles with the number 3 represent Track 3. Each icon is associated with a text alternative reading "Track 1," "Track 2," or "Track 3," as appropriate.

Example 3: A form with required fields
A form contains several required fields. The labels for the required fields are displayed in red. In addition, at the end of each label is an asterisk character, *. The instructions for completing the form indicate that "all required fields are displayed in red and marked with an asterisk *", followed by an example.
Note: Asterisks may not be read by all screen readers (in all reading modes) and may be difficult for users with low vision because they are rendered in a smaller size than default text. It is important for authors to include the text indicating that asterisk is used and to consider increasing the size of the asterisk that is presented.

Example 4: A form with a green submit button
An on-line loan application explains that green buttons advance in the process and red buttons cancel the process. A form contains a green button containing the text *Go*. The instructions say, "Press the button labeled *Go* to submit your results and proceed to the next step."

Testing Procedure
For each item where a color difference is used to convey information:
1. Check that the information conveyed is also available in text and that the text is not conditional content.

Technique G205: Including a text cue for colored form control labels[74]

Example: Required fields in an HTML form
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.

[71] https://www.w3.org/WAI/WCAG21/quickref/?showtechniques=131%2C132%2C133%2C134%2C141%2C142%2C143%2C144%2C145%2C1410%2C1411%2C1412#distinguishable
[72] https://www.w3.org/WAI/WCAG21/Understanding/use-of-color.html
[73] https://www.w3.org/WAI/WCAG21/Techniques/general/G14.html
[74] https://www.w3.org/WAI/WCAG21/Techniques/general/G205.html

```
<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>
```

Testing Procedure

For any content where color differences are used to convey information:

1. Check that the same information is available through text or character cues.

Technique G182: Ensuring that additional visual cues are available when text color differences are used to convey information[75]

Examples

- The default formatting for links on a page includes presenting them both in a different color than the other text on the page underlining them to make the links identifiable even without color vision.
- An article comparing the use of similar elements in different markup languages uses colored text to identify the elements from each language. Elements from the first markup language are identified using BLUE, bolded text. Elements from the second are presented as RED, italicized text.
- A news site lists links to the articles appearing on its site. Additional information such as the section the article appears in, the time the article is posted, a related location or an indication that it is accompanied by live video appears in some cases. The links to the articles are in a different color than the additional information but the links are not underlined, and each link is presented in a larger font than the rest of the information so that users who have problems distinguishing between colors can identify the links more easily.
- Short news items sometimes have sentences that are also links to more information. Those sentences are printed in color and use a sans-serif font face while the rest of the paragraph is in black Times-Roman.

Testing Procedure

1. Locate all instances where the color of text is used to convey information.
2. Check that any text where color is used to convey information is also styled or uses a font that makes it visually distinct from other text around it.

Technique G183: Using a contrast ratio of 3:1 with surrounding text and providing additional visual cues on hover for links or controls where color alone is used to identify them[76]

Example 1: Colors that would provide 3:1 contrast with black words and 4.5:1 contrast with a white background

Refer to Links with a 3:1 contrast ratio with surrounding text

Example 2

The hypertext links in a document are medium-light blue (#3366CC) and the regular text is black (#000000). Because the blue text is light enough, it has a contrast of 3.9:1 with the surrounding text and can be identified as being different than the surrounding text by people with all types of color blindness, including those individuals who cannot see color at all.

Testing Procedure

For each instance where color is used to convey information about text:

1. Check that the relative luminance of the color of the text differs from the relative luminance of the surrounding text by a contrast ratio of at least 3:1.
2. Check that hovering over the link causes a visual enhancement (such as an underline, font change, etc.)

[75] https://www.w3.org/WAI/WCAG21/Techniques/general/G182.html
[76] https://www.w3.org/WAI/WCAG21/Techniques/general/G183.html

Situation B: If color is used within an image to convey information:

Technique G111: Using color and pattern[77]

Example 1
A real estate site provides a bar chart of average housing prices in several regions of the United States. The bar for each region is displayed with a different solid color and a different pattern. The legend uses the same colors and patterns to identify each bar.

Example 2
An on-line map of a transportation system displays each route in a different color. The stops on each route are marked with a distinctive icon such as a diamond, square, or circle to help differentiate each route.

Example 3
A flow chart describes a set of iterative steps to complete a process. It uses dashed, arrowed lines with a green background to point to the next step in the process when the specified condition passes. It uses dotted arrowed lines with a red background to point to the next step in the process when the specified condition fails.

Example 4
The content includes an interactive game. The game pieces for the 4 players are distinguished from one another using both color and pattern.

Testing Procedure
For each image within the Web page that use color differences to convey information:
1. Check that all information that is conveyed using color is also conveyed using patterns that do not rely on color.

Technique G14: Ensuring that information conveyed by color differences is also available in text

Examples
{See Above}
Testing Procedure
{See Above}

1.4.2 Audio Control (Level A)[78] – On the target site, if other audio is playing at the same time as the voice output, people who use screen reading software might have trouble hearing it. When the screen reader's voice output is software-based (as most are) and is controlled by the same volume control as the sound, this problem is made worse. The guidelines make it clear that it is crucial that the user be able to turn off the background noise because of this. Reducing the volume to zero is part of having control over the volume.

Technique G60: Playing a sound that turns off automatically within three seconds[79]

Examples
- Example 1: A Web page opens with a trumpet fanfare and then goes silent
- Example 2: A homepage opens with the chairman saying "Binfor, where quality is our business." then going silent.
- Example 3: A Web page opens with instructions on how to get started: "To begin, press the enter key."
- Example 4: A Web page opens with a warning and then goes silent.

Testing Procedures
1. Load the Web page
2. Check that all sound that plays automatically stops in 3 seconds or less

[77] https://www.w3.org/WAI/WCAG21/Techniques/general/G111.html
[78] https://www.w3.org/WAI/WCAG21/Understanding/audio-control.html
[79] https://www.w3.org/WAI/WCAG21/Techniques/general/G60.html

Technique G170: Providing a control near the beginning of the Web page that turns off sounds that play automatically[80]

Example 1
A Web page contains a time-based media presentation that includes an audio track as well as an animated video describing how to repair a lawnmower engine. The page contains 2 buttons that say "Pause" and "Stop", which give the user control over when and if the time-based media plays.

Example 2
A Web page contains an embedded short film. The page contains a button that says, "Pause the movie", which allows the user to pause the film.

Example 3
A Web page contains a presentation that includes video and audio. The page contains a button that says "Turn off multimedia", which allows the user to stop any video and audio from playing.

Testing Procedure
1. Load a Web page.
2. Check for music or sounds that start automatically.
3. Check that a control that allows the user to turn off the sounds is provided near the beginning of the page.

Technique G171: Playing sounds only on user request[81]

Example 1
A Web page from a grey whale conservation society has a looping background sound of grey whales singing. There are also sounds of water splashing. The sounds do not start automatically. Instead, the Web content provides a link at the top of the page to allow the user to start the sounds manually. The button says, "Turn sounds on." After pressing the "turn sounds on" button, the sounds are heard. The user is then presented with an option to "turn sounds off."

Example 2
A link is provided to a sound file that includes the sounds of the grey whales. The link text says, "Hear the song of the grey whale (mp3)."

Testing Procedure
1. Load a Web page that is known to contain sounds that play for 3 seconds or longer.
2. Check that no sounds play automatically.
3. Check that there is a way for a user to start sounds manually.

1.4.3 Contrast (Minimum) (Level AA)[82] – Hue and saturation have little to no impact on readability, as measured by reading performance, in those without color impairments (Knoblauch et al., 1991). Luminance contrast can be slightly impacted by color deficits. So that those with color vision impairments will also have sufficient contrast between the text and the backdrop, the recommendation's contrast calculation does not take color into account. Excluded is ornamental text that provides no information. For instance, it won't need to meet this requirement if random words are used to form a background and the words can be swapped out or rearranged without changing their meaning.

Situation A: text is less than 18-point if not bold and less than 14-point if bold

Technique G18: Ensuring that a contrast ratio of at least 4.5:1 exists between text (and images of text) and background behind the text[83]

Examples

[80] https://www.w3.org/WAI/WCAG21/Techniques/general/G170.html
[81] https://www.w3.org/WAI/WCAG21/Techniques/general/G171.html
[82] https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html
[83] https://www.w3.org/WAI/WCAG21/Techniques/general/G18.html

- A black background is chosen so that light colored letters that match the company logo can be used.
- Text is placed over a picture of the college campus. Since a wide variety of colors and shades appear in the picture, the area behind the text is fogged white so that the picture is very faint, and the maximum darkness is still light enough to maintain a 4.5:1 contrast ratio with the black text written over the picture.

  See also the contrast samples in related resources.

Testing Procedure

(1) Measure the relative luminance of each letter (unless they are all uniform) using the formula:
- L = 0.2126 * **R** + 0.7152 * **G** + 0.0722 * **B** where **R**, **G** and **B** are defined as:
  - if R sRGB <= 0.04045 then **R** = R sRGB /12.92 else **R** = ((R sRGB +0.055)/1.055) ^ 2.4
  - if G sRGB <= 0.04045 then **G** = G sRGB /12.92 else **G** = ((G sRGB +0.055)/1.055) ^ 2.4
  - if B sRGB <= 0.04045 then **B** = B sRGB /12.92 else **B** = ((B sRGB +0.055)/1.055) ^ 2.4

and R sRGB, G sRGB, and B sRGB are defined as:
- R sRGB = R 8bit /255
- G sRGB = G 8bit /255
- B sRGB = B 8bit /255

(2) Measure the relative luminance of the background pixels immediately next to the letter using same formula.

(3) Calculate the contrast ratio using the following formula.
- (L1 + 0.05) / (L2 + 0.05), where
  - L1 is the relative luminance of the lighter of the foreground or background colors, and
  - L2 is the relative luminance of the darker of the foreground or background colors.

(4) Check that the contrast ratio is equal to or greater than 4.5:1

Technique G148: Not specifying background color, not specifying text color, and not using technology features that change those defaults[84]

Examples

The author specifies neither text color nor background and does not use CSS. As a result, the user can set their browser defaults to provide the colors and contrasts that work well for them.

Testing Procedure

1. Look in all places that text color can be specified
2. Check that text color is not specified
3. Look in all areas that background color or image used as a background can be specified
4. Check that no background color or image used as a background is specified

---

Technique G174: Providing a control with a sufficient contrast ratio that allows users to switch to a presentation that uses sufficient contrast[85]

Examples

- A page with some headlines that do not meet the 3:1 contrast requirements has a high contrast (5:1) link at the top of the page that takes the user to a new version of the page with minimum 4.5:1 contrast on all text and images of text.
- A page uses shaded backgrounds for effect but results in text to background contrast of 4:1. A control at the top of the page says, "high contrast". Clicking on it causes different styles to be used and dropping the background colors to achieve 7:1 contrast.

Testing Procedure

1. Check that a link or control exists on the original page that provides access to the alternate version.

---

[84] https://www.w3.org/WAI/WCAG21/Techniques/general/G148.html
[85] https://www.w3.org/WAI/WCAG21/Techniques/general/G174.html

2. Check that the link or control on the original page conforms to all success criteria for the conformance level being tested.
3. Check that the alternate version meets the contrast and all other success criteria for the conformance level being tested.

Situation B: text is at least 18-point if not bold and at least 14-point if bold

Technique G145: Ensuring that a contrast ratio of at least 3:1 exists between text (and images of text) and background behind the text[86]

Examples

- A black background is chosen so that light colored letters that match the company's logo can be used.

  Larger-scale text is placed over a picture of the college campus. Since a wide variety of colors and darknesses appear in the picture, the area behind the text is fogged white so that the picture is very faint, and the maximum darkness is still light enough to maintain a 3:1 contrast ratio with the black text written over the picture.

Testing Procedure

(1) Measure the relative luminance of each letter (unless they are all uniform) using the formula:
   - L = 0.2126 * **R** + 0.7152 * **G** + 0.0722 * **B** where **R**, **G** and **B** are defined as:
     - if R sRGB <= 0.04045 then **R** = R sRGB /12.92 else **R** = ((R sRGB +0.055)/1.055) ^ 2.4
     - if G sRGB <= 0.04045 then **G** = G sRGB /12.92 else **G** = ((G sRGB +0.055)/1.055) ^ 2.4
     - if B sRGB <= 0.04045 then **B** = B sRGB /12.92 else **B** = ((B sRGB +0.055)/1.055) ^ 2.4

   and R sRGB, G sRGB, and B sRGB are defined as:
     - R sRGB = R 8bit /255
     - G sRGB = G 8bit /255
     - B sRGB = B 8bit /255

(2) Measure the relative luminance of the background pixels immediately next to the letter using same formula.
(3) Calculate the contrast ratio using the following formula.
   - (L1 + 0.05) / (L2 + 0.05), where
     - L1 is the relative luminance of the lighter of the foreground or background colors, and
     - L2 is the relative luminance of the darker of the foreground or background colors.
(4) Check that the contrast ratio is equal to or greater than 3:1

Technique G148: Not specifying background color, not specifying text color, and not using technology features that change those defaults

Examples
{See Above}
Testing Procedure
{See Above}

Technique G174: Providing a control with a sufficient contrast ratio that allows users to switch to a presentation that uses sufficient contrast

Examples
{See Above}
Testing Procedure
{See Above}

[86] https://www.w3.org/WAI/WCAG21/Techniques/general/G145.html

1.4.4 Resize Text (Level AA)[87] – The goal of this Success Criterion is to make sure that visually rendered text, including text-based controls, can be successfully scaled so that people with mild visual impairments can read it without the aid of assistive technology like a screen magnifier. Text-based controls are text characters that have been displayed so they can be seen, as opposed to text characters that are still in data form like ASCII. All of the web page's content may benefit from scaling for users, but text is the most important.

Technique G142: Using a technology that has commonly available user agents that support zoom[88]

Examples

- Internet Explorer 7 and Opera 9 provide a zoom function that scales HTML/CSS page content uniformly.
- To allow users to resize text, Adobe Reader provides a magnification tool that scales PDF pages uniformly.

Testing Procedures

1. Display content in a user agent
2. Zoom content to 200%
3. Check whether all content and functionality is available

Ensuring that text containers resize when the text resizes AND using measurements that are relative to other measurements in the content by using one or more of the following techniques:

Technique C28: Specifying the size of text containers using `em` units[89]

Example 1: Em units for sizes for layout container containing text

In this example, `a div` element, with `id` value of "nav_menu", is used to position the navigation menu along the left-hand side of the main content area of the Web page. The navigation menu consists of a list of text links, with id value of "nav_list." The text size for the navigation links and the width of the container are specified in em units.

```
#nav_menu { width: 20em; height: 100em }

#nav_list { font-size: 100%; }
```

Example 2: Em units for text-based form controls

In this example, `input` elements that contain text or accept text input by the user have been given the class name "form1." CSS rules are used to define the font size in percent units and width for these elements in `em` units. This will allow the text within the form control to resize in response to changes in text size settings without being cropped (because the width of the form control itself also resizes according to the font size).

```
input.form1 { font-size: 100%; width: 15em; }
```

Example 3: Em units in dropdown boxes

In this example, select elements have been given the class name "pick." CSS rules are used to define the font size in percent units and width in em units. This will allow the text within the form control to resize in response to changes in text size settings without being cropped.

```
input.pick { font-size: 100%; width: 10em; }
```

Example 4: Em units for non-text-based form controls

In this example, `input` elements that define checkboxes or radio buttons have been given the class name "choose." CSS rules are used to define the width and height for these elements in `em` units. This will allow the form control to resize in response to changes in text size settings.

```
input.choose { width: 1.2em; height: 1.2em; }
```

Testing Procedures

[87] https://www.w3.org/WAI/WCAG21/Understanding/resize-text.html
[88] https://www.w3.org/WAI/WCAG21/Techniques/general/G142.html
[89] https://www.w3.org/WAI/WCAG21/Techniques/css/C28.html

- Identify containers that contain text or allow text input.
- Check the container's width and/or height are specified in `em` units.

Techniques for relative measurements:

Technique C12: Using percent for font sizes[90]

Example: Percent font sizes in CSS

This example defines the font size for the strong element so that its text will always be larger than the surrounding text, in whatever context it is used. Assuming that headings and paragraphs use different font sizes, the emphasized words in this example will each be larger than their surrounding text.

```
strong {font-size: 120%}

...

<h1>Letting the <strong>user</strong> control text size</h1>
<p>Since only the user can know what size text works for him,
it is <strong>very</strong> important to let him configure the text size.
…
```

Testing Procedures

1. Check that the value of the CSS property that defines the font size is a percentage.

Technique C13: Using named font sizes[91]

Example 1: Named font sizes in CSS

This example selects a larger font size for strong elements so that their text will always be larger than the surrounding text, in whatever context they are used. Assuming that headings and paragraphs use different font sizes, the emphasized words in this example will each be larger than their surrounding text.

```
strong {font-size: larger}

...

<h1>Letting the <strong>user</strong> control text size</h1>
<p>Since only the user can know what size text works for him,
it is <strong>very</strong> important to let him configure the text size.
…
```

Testing Procedures

1. Check that the value of the CSS property that defines the font size is one of xx-small, xx-small, x-small, small, medium, large, x-large, xx-large, xsmaller, or larger.

Technique C14: Using `em` units for font sizes[92]

Example 1: Em font sizes in CSS

This example defines the font size for strong element so that its text will always be larger than the surrounding text, in whatever context it is used. Assuming that headings and paragraphs use different font sizes, the strong words in this example will each be larger than their surrounding text.

```
strong {font-size: 1.6em}

...
<h1>Letting the <strong>user</strong> control text size</h1>
<p>Since only the user can know what size text works for him,
it is <strong>very</strong> important to let him configure the text size.  </p>
…
```

[90] https://www.w3.org/WAI/WCAG21/Techniques/css/C12.html
[91] https://www.w3.org/WAI/WCAG21/Techniques/css/C13.html
[92] https://www.w3.org/WAI/WCAG21/Techniques/css/C14.html

Testing Procedures

1. Check that the value of the CSS property that defines the font size is expressed in em units.

Techniques for text container resizing:

Technique SCR34: Calculating size and position in a way that scales with text size[93]

Examples

The Javascript function:

```
function calculatePosition(objElement, strOffset)
{
    var iOffset = 0;

    if (objElement.offsetParent)
    {
        do
        {
            iOffset += objElement[strOffset];
            objElement = objElement.offsetParent;
        } while (objElement);
    }

    return iOffset;
}
```

The following example illustrates using the function above by aligning an object beneath a reference object, the same distance from the left:

```
// Get a reference object
var objReference = document.getElementById('refobject');
// Get the object to be aligned
var objAlign = document.getElementById('lineup');

objAlign.style.position = 'absolute';
objAlign.style.left = calculatePosition(objReference, 'offsetLeft') + 'px';
objAlign.style.top      =      calculatePosition(objReference,      'offsetTop')      +
objReference.offsetHeight + 'px';
```

Testing Procedures

1. Open a page that is designed to adjust container sizes as text size changes.
2. Increase the text size up to 200% using the browser's text size adjustment (not the zoom feature).
3. Examine the text to ensure the text container size is adjusted to accommodate the size of the text.
4. Ensure that no text is "clipped" or has disappeared as a result of the increase in text size.

Technique G146: Using liquid layout

Example: Simple liquid layout in HTML and CSS

The following fairly simple example uses HTML and CSS to create a liquid layout. The three columns adjust their size as text size is adjusted. When the total horizontal width exceeds the available width of the columns, the last column wraps to be positioned below, rather than beside, the previous column. The font size can be increased without either clipping or introducing horizontal scrolling until the longest word no longer fits in a column. This particular example uses percent sizes for the columns and defines them as floating regions using the "float" property.

```
             <!DOCTYPE    html    PUBLIC    "-//W3C//DTD    XHTML    1.0    Strict//EN"
"https://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
               <html xmlns="http://www.w3.org/1999/xhtml">
               <head>
               <meta http-equiv="content-type" content="text/html; charset=utf-8" />
               <title>Example of Basic Liquid Layout</title>
               <style type="text/css">
               .column
               {
               border-left: 1px solid green;
```

[93] https://www.w3.org/WAI/WCAG21/Techniques/client-side-script/SCR34.html

```
padding-left:1%;
float: left;
width: 32%;
}
#footer
{
border-top: 1px solid green;
clear: both;
}
</style>
</head>
<body>
<h1>WCAG Example</h1>
<h2>Text in Three Columns</h2>
<div title="column one" class="column">
<h3>Block 1</h3>
<p> The objective of this technique is to be able to present content
without introducing horizontal scroll bars by using layout
techniques that adapt to the available horizontal space.
</p>
</div>
<div title="column two" class="column">
<h3>Block 2</h3>
<p> This is a very simple example of a page layout that adapts as the
text size changes.
</p>
</div>
<div title="column three" class="column">
<h3>Block 3</h3>
<p> For techniques that support more complex page layouts, see the
Resources listed below.
</p>
</div>
<p id="footer">Footer text</p>
</body>
</html>
```

Testing Procedure

1. Display content in a user agent.
2. Increase text size to 200%.
3. Check whether all content and functionality is available with no horizontal scrolling.

Technique G178: Providing controls on the Web page that allow users to incrementally change the size of all text on the page up to 200 percent[94]

Examples

- A newspaper article has two buttons near the top of the page. The "increase text size" button has a big letter "T" with an upward arrow and the "decrease text size" button has a small letter "T" with a down arrow. There is alt text on each button.
- A site has a number of style sheets with different text size. The user can choose any of the style sheets if their browser provides this functionality. Each page also includes the links "Increase text size" and "Decrease text size" that will change the style sheet currently applied to the appropriate alternate style sheet.
- A site includes the text "Change text size:" followed by text links "Up" and "Down" on every Web page. The links trigger a Javascript that alters the value of the text-size property accordingly.
- A site includes a link on every page that reads "Change text size." The resulting page includes a series of links that includes options representing the available sizes. The links read, "Smallest font size," "Small font size," "Default font size," "Large font size," etc. Instructions preceding the list direct users to choose a link to change to the desired font size.

Testing Procedure

1. Set the viewport size to 1024px by 768px or larger.

[94] https://www.w3.org/WAI/WCAG21/Techniques/general/G178.html

2. Increase the text size and check to see if the text size increased.
3. Check that the text size can be increased to 200% of the original size.
4. Check that after increasing the text size to 200% of the original size, there is no loss of content or functionality (e.g. no parts of the text are clipped, boxes do not overlap, controls are not obscured or separated from their labels, etc.).
5. Decrease the text size to its default value and check to see if it in fact returned to the default size.

Technique G179: Ensuring that there is no loss of content or functionality when the text resizes, and text containers do not change their width[95]

Example 1: A multi-column page layout
HTML and CSS are used to create a two-column layout for a page of text. Using the default value of the white-space property, normal, causes text to wrap. So as the size of the text is increased to 200%, the text reflows and the column of text grows longer. If the column is too long for the viewport, the user agent provides scrollbars so the user can scroll text into view because the author has specified the CSS rule overflow:scroll or overflow:auto.

Example 2
A newspaper layout with blocks of text in columns. The blocks have a fixed width, but no height set. When the text is resized in the browser, the text wraps and makes the blocks taller.

Testing Procedure
1. Increase text size to 200%.
2. Check whether all content and functionality is available.

1.4.5 Images of Text (Level AA)[96] – The information should be presented as text rather than a picture if the author can do so without sacrificing the material's visual impact. If the author is unable to format the text to achieve the desired result for any reason, the effect won't be reliably displayed on user agents that are widely used or if using a technology to meet this criterion would conflict with meeting other criterion such as 1.4.4, then an image of the text may be used instead. This includes situations, such as type samples, logotypes, branding, etc., where a certain text presentation is necessary to the information being communicated. Images of text may also be used to ensure that the text is anti-aliased on all user agents or to utilize a specific typeface that is either not widely used or that the author does not have permission to share.

Technique C22: Using CSS to control visual presentation of text[97]

Example 1: Using CSS font-family to control the font family for text
The XHTML component:
```
<p>The Javascript method to convert a string to uppercase is <code>toUpperCase()</code>.</p>
```

The CSS component:
```
code { font-family:"Courier New", Courier, monospace }
```

Example 2: Using CSS text-align to control the placement (alignment) of text
The XHTML component:
```
<p class="right">This text should be to the right of the viewport.</p>
```

The CSS component:
```
.right { text-align: right; }
```

Example 3: Using CSS font-size to control the size of text
The XHTML component:
```
<p>09 <strong class="largersize">March</strong> 2008</p>
```

[95] https://www.w3.org/WAI/WCAG21/Techniques/general/G179.html
[96] https://www.w3.org/WAI/WCAG21/Understanding/images-of-text.html
[97] https://www.w3.org/WAI/WCAG21/Techniques/css/C22.html

The CSS component:

```
strong.largersize { font-size: 1.5em; }
```

Example 4: Using CSS color to control the color of text

The style used in this example is not used to convey information, structure or relationships.

The XHTML component:

```
<p>09 <em class="highlight">March</em> 2008</p>
```

The CSS component:

```
.highlight{ color: red; }
```

Example 5: Using CSS font-style to italicize text

The style used in this example is not used to convey information, structure or relationships.

The XHTML component:

```
<p>The article is available in the <a href="http://www.example.com"
class="featuredsite">Endocrinology
Blog</a>.</p>
```

The CSS component:

```
.featuredsite{ font-style:italic; }
```

Example 6: Using CSS font-weight to control the font weight of the text

The style used in this example is not used to convey information, structure or relationships.

The XHTML component:

```
<p>This deal is available <span class="highlight">now!</span></p>
```

The CSS component:

```
.highlight { font-weight:bold; color:#990000; }
```

Example 7: Using CSS text-transform to control the case of text

The style used in this example is not used to convey information, structure or relationships.

The XHTML component:

```
<p>09 <span class="caps">March</span> 2008</p>
```

The CSS component:

```
.caps { text-transform:uppercase; }
```

Example 8: Using CSS line-height to control spacing between lines of text

The CSS line-height property is used to display the line height for the paragraph at twice the height of the font.

The XHTML component:

```
<p>Concern for man and his fate must always form the<br />
chief interest of all technical endeavors. <br />
Never forget this in the  midst of your diagrams and equations. </p>
```

The CSS component:

```
p { line-height:2em; }
The CSS line-height property is used to display the line height for the text at less than the
height of the font. The second line of text is positioned after the first line of text and
visually appears as though the text is part of the first line but dropped a little.
```

The XHTML component:

```
<h1 class="overlap"><span class="upper">News</span><br />
<span class="byline">today</span></h1>
```

The CSS component:

```
.overlap { line-height:0.2em;  }
```

```
.upper { text-transform:uppercase; }
.byline { color:red; font-style:italic; font-weight:bold; padding-left:3em; }
```

Example 9: Using CSS letter-spacing to space text
The CSS letter-spacing property is used to display the letters farther apart in the heading.

The XHTML component:
```
<h1 class="overlap"><span class="upper">News</span><br />
<span class="byline">today</span></h1>
```

The CSS component:
```
.overlap { line-height:0.2em;  }
.upper { text-transform:uppercase; }
.byline { color:red; font-style:italic; font-weight:bold; padding-left:3em; letter-spacing:-
0.1em; }
```

The CSS letter-spacing property is used to display the letters closer together in the second line of text.

The XHTML component:
```
<h1 class="upper2">News</h1>
The CSS component:
.upper2 { text-transform:uppercase; letter-spacing:1em; }
```

Example 10: Using CSS background-image to layer text and images
The CSS font-style property is used to display the textual component of a banner and background-image property is used to display a picture behind the text.

The XHTML component:
```
<div id="banner"><span id="bannerstyle1">Welcome</span>
<span id="bannerstyle2">to your local city council</span></div>
```

The CSS component:
```
#banner {
  color:white;
  background-image:url(banner-bg.gif);
  background-repeat:no-repeat;
  background-color:#003399;
  width:29em;
}

#bannerstyle1 {
  text-transform:uppercase;
  font-weight:bold;
  font-size:2.5em;
}

#bannerstyle2 {
  font-style:italic;
  font-weight:bold;
  letter-spacing:-0.1em;
  font-size:1.5em;
}
```

Example 11: Using CSS first-line to control the presentation of the first line of text
The CSS ::first-line pseudo-element is used to display the first line of text in a larger, red font.

The XHTML component:
```
<p class="startline">Once upon a time...<br />
...in a land far, far away...  </p>
```

The CSS component:
```
.startline::first-line { font-size:2em; color:#990000; }
```

Example 12: Using CSS first-letter to control the presentation of the first letter of text
The CSS ::first-letter pseudo-element is used to display the first letter in a larger font size, red and vertically aligned in the middle.

The XHTML component:
```
<p class="startletter">Once upon a time...</p>
```

The CSS component:
```
.startletter::first-letter { font-size:2em; color:#990000; vertical-align:middle; }
```

Testing Procedure
1. Check whether CSS properties are used to control the visual presentation of text

Technique C30: Using CSS to replace text with images of text and providing user interface controls to switch[98]

Examples
A design studio site uses a style switcher to allow users to view two presentations of their home page. For the default version, the heading text is replaced with images of text. A control on the page allows users to switch to a version that presents the headings as text.

The CSS component:
```
...
<div id="Header">
  <h1><span>Pufferfish Design Studio</span></h1>
  <h2><span>Surprising Identity and Design Solutions</span></h2>
  </div>
...
```
The CSS for the presentation that includes images of text follows. Note that the CSS uses positioning to place the contents of the heading elements offscreen so that the text remains available to screen reader users.
```
...
#Header h1 {
        background-image: url(pufferfish-logo.png);
        height: 195px;
        width: 290px;
        background-repeat: no-repeat;
        margin-top: 0;
        position: absolute;
        }
#Header h1 span {
        position: absolute;
         left: -999em;
        }
#Header h2 {
        background-image: url(beauty.png);
        background-repeat: no-repeat;
        height: 234px;
        width: 33px;
        margin-left: 8px;
        position: absolute;
        margin-top: 250px;
        }
#Header h2 span {
        position: absolute;
         left: -999em;
        }
```

The CSS for the presentation that does not include images of text.
```
...
#Header h1 {
        font: normal 200%/100% Garamond, "Times New Roman", serif;
        margin-bottom: 0;
        color: #000099;
```

[98] https://www.w3.org/WAI/WCAG21/Techniques/css/C30.html

```
        background: #ffffff;
        }

#Header h2 {
        font: normal 160%/100% Garamond, "Times New Roman", serif;
        margin-bottom: 0;
        color: #336600;
        background: #ffffff;
        }
```

Testing Procedure

1. Check that the Web page includes a control that allows users to select an alternate presentation.
2. Check that when the control is activated the resulting page includes text (programmatically determined text) wherever images of text had been used.

Technique G140: Separating information and structure from presentation to enable different presentations

Examples

{See Above}

Testing Procedure

{See Above}

1.4.10 Reflow (Level AA)[99] – This Success Criterion is meant to assist those who have impaired eyesight and need to magnify text so they can read it in a single column. Content reflows, or is presented in one column, so that scrolling in more than one direction is not required, when the browser zoom is used to scale content to 400%. Enlarged text with reflow makes reading possible for those with low vision. It is essential. Character perception is made possible through enlargement. Reflow makes tracking possible. Tracking involves moving between the ends of one line and the beginning of the next line of text. The guidelines state that it is crucial to avoid having to scroll in the direction of reading to disclose lines that are obscured by the viewport because doing so considerably increases the amount of effort needed to read. Additionally, it's crucial that content is not obscured off-screen. A vertically scrolling website shouldn't, for instance, have material that is hidden to the side when zoomed into.

Technique C32: Using media queries and grid CSS to reflow columns[100]

Example: Grid layout in HTML and CSS - Medium complexity

The following medium complexity example uses HTML and CSS to create a grid layout. The layout regions adjust their size as the viewport is adjusted. When the total viewport width matches the width defined via media queries, columns wrap to be positioned below, rather than beside each other or vice versa.

The zoom level can be increased to 400% without requiring scrolling in more than one direction. This particular example uses fr units as a fraction of the free space of the grid container for the grid items by using the "grid-template-columns" property and are laid out in source order.

```
<!DOCTYPE html>
  <html lang="en">
    <head>
      <meta charset="UTF-8">
      <title>CSS: Using media queries and grid CSS to reflow columns</title>
      <style>

      /* Reflow Styling */
      header[role="banner"]       { grid-area: header; }
      main[role="main"]           { grid-area: main; }
      aside[role="complementary"] { grid-area: aside; }
```

[99] https://www.w3.org/WAI/WCAG21/Understanding/reflow.html

[100] https://www.w3.org/WAI/WCAG21/Techniques/css/C32.html

```
        footer[role="contentinfo"]  { grid-area: footer; }

        .grid,
        .subgrid {
          display: grid;
          grid-template-columns: minmax(0, 1fr);
        }

        .grid {
          grid-template-areas:
          'header'
          'main'
          'aside'
          'footer';
          width: 100%;
        }

        .subgrid {
          width: calc(100% + 2rem);
          margin: 0 -1rem;
        }

        .grid-item,
        .subgrid-item {
          padding: 1rem;
        }

        @media all and (min-width: 576px) {
          .subgrid {
            grid-template-columns: minmax(0, 1fr) minmax(0, 1fr);
            margin-bottom: 1rem;
          }
          .subgrid-item {
            padding-bottom: 0.25rem;
          }
        }

        @media all and (min-width: 992px) {
          .grid {
            grid-template-areas:
              'header header header'
              'main main aside'
              'footer footer footer';
            grid-template-columns: minmax(0, 1fr) minmax(0, 1fr) minmax(0, 1fr);
          }
        }

        </style>

      </head>

      <body class="grid">

        <header role="banner" class="grid-item">
          ...
        </header>

        <main role="main" class="grid-item">
          ...
          ...
          <div class="subgrid">
            <div class="subgrid-item">
              ...
            </div>
            <div class="subgrid-item">
              ...
            </div>
          </div>
        </main>
```

```
      <aside role="complementary" class="grid-item">
        ...
      </aside>

      <footer role="contentinfo" class="grid-item">
        ...
      </footer>

    </body>
  </html>
```

Testing Procedure

1. Display the web page in a user agent capable of 400% zoom and set the viewport dimensions (in CSS pixels) to 1280 wide and 1024 high.
2. Zoom in by 400%.
3. For content read horizontally, check that all content and functionality is available without horizontal scrolling.
4. For content read vertically, check that all content and functionality is available without vertical scrolling.

NOTE: If the browser is not capable of zooming to 400%, you can reduce the width or height of the browser proportionally. For example, at 300% zoom the viewport should be sized to 960px wide.

Technique C31: Using CSS Flexbox to reflow content[101]

Examples: Medium complex flexbox layout in HTML and CSS

The following medium complex example uses HTML and CSS to create a flexbox layout. The layout regions adjust their size as the viewport is adjusted. When the total viewport width matches the width defined via media queries, columns wrap to be positioned below, rather than beside each other or vice versa.

The zoom level can be increased to 400% without requiring scrolling in more than one direction. This particular example uses percent sizes for the flex items by using the "flex-basis" property and are laid out in source order.

```
<!DOCTYPE html>
<html lang="en">
  <head>
  <meta charset="UTF-8">
  <title>Using CSS Flexbox for Reflow</title>
  <style>

  /* Reflow Styling */

  .row {
    width: 100%;
    display: flex;
    flex-flow: row wrap;
  }

  .row-nested {
    width: calc(100% + 2rem);
    margin: 0 -1rem 1rem -1rem;
  }

  .col {
    padding: 1rem;
    flex: 0 1 100%;
  }

  @media all and (min-width: 576px) {
    .col-panel {
      flex: 0 1 50%;
      padding-bottom: 0.25rem;
    }
  }
```

[101] https://www.w3.org/WAI/WCAG21/Techniques/css/C31.html

```
    @media all and (min-width: 992px) {
      main[role="main"] {
        flex: 0 1 66.333333%;
      }
      aside[role="complementary"] {
        flex: 0 1 33.333333%;
        margin-top: 0;
      }
    }

    </style>

    </head>

    <body class="row">

      <header role="banner" class="col">
        ...
      </header>

      <main role="main" class="col">
        ...
        ...
        <div class="row row-nested">
          <div class="col col-panel">
            ...
          </div>
          <div class="col col-panel">
            ...
          </div>
        </div>
      </main>

      <aside role="complementary" class="col">
        ...
      </aside>

      <footer role="contentinfo" class="col">
        ...
      </footer>

    </body>
  </html>
```

Testing Procedure

1. Display the web page in a user agent capable of 400% zoom and set the viewport dimensions (in CSS pixels) to 1280 wide and 1024 high.
2. Zoom in by 400%.
3. For content read horizontally, check that all content and functionality is available without horizontal scrolling.
4. For content read vertically, check that all content and functionality is available without vertical scrolling.

NOTE: If the browser is not capable of zooming to 400%, you can reduce the width of the browser proportionally. For example, at 300% zoom the viewport should be sized to 960px wide.

Technique C33: Allowing for Reflow with Long URLs and Strings of Text[102]

Examples: Breaking long URLs

Using the following CSS will cause long URLs to break at appropriate places (hyphens, spaces, etc.) and within words without causing reflow.

List of CSS declarations used and why they are used:

- **overflow-wrap: break-word**: Allows words to be broken and wrapped within words.
- **word-wrap: break-word**: Allows words to be broken and wrapped within. (Microsoft only)

```
a {overflow-wrap: break-word;}
```

[102] https://www.w3.org/WAI/WCAG21/Techniques/css/C33.html

IE and Edge only support this declaration when used with the * (wildcard) selector

```
* { word-wrap: break-word;}
```

Testing Procedure

1. Display the web page in a user agent capable of 400% zoom and set the viewport dimensions (in CSS pixels) to 1280 wide and 1024 high.
2. Zoom in by 400%.
3. For content read horizontally, check that all content and functionality is available without horizontal scrolling.
4. For content read vertically, check that all content and functionality is available without vertical scrolling.

NOTE: If the browser is not capable of zooming to 400%, you can reduce the width of the browser proportionally. For example, at 300% zoom the viewport should be sized to 960px wide.

Technique C38: Using CSS `width, max-width and flexbox to fit labels` and `inputs`[103]

Examples: Fitting labels, inputs and flexbox layout with HTML and CSS.

The following example uses HTML and CSS to fit labels and inputs within various width containers, including the viewport. The layout regions adjust their size as the viewport is adjusted. The labels and inputs subsequently adjust their size to fit within the layout region containers.

The zoom level can be increased to 400% without requiring horizontal scrolling. This particular example uses a percent size for the width and max-width for the labels and inputs. The max-width is applied in order to fix elements spilling out of the grid in a cross-browser way, as replaced elements such as the select have intrinsic sizing.

```
<style>

/* Fitting Inputs Styling */

.form-group {
  display: flex;
  flex-flow: row wrap;
  margin: 0 -1rem 1rem -1rem;
}

[class*="form-col"] {
  flex: 0 1 100%;
  padding: 0 1rem;
}

@media (min-width: 576px) {
  .form-col-4 {
    flex: 0 0 33.33333%;
    max-width: 33.33333%;
  }

  .form-col-8 {
    flex: 0 0 66.66667%;
    max-width: 66.66667%;
  }

  .offset-form-col-4 {
    margin-left: 33.33333%;
  }

}

input {
  display: block;
  width: 100%;
}

label,
select {
```

[103] https://www.w3.org/WAI/WCAG21/Techniques/css/C38.html

```
  display: block;
  width: 100%;
  max-width: 100%;
}

</style>

<div class="form-group">
  <div class="form-col-4">
    <label for="fname">First Name</label>
  </div>
  <div class="form-col-8">
    <input type="text" id="fname" autocomplete="given-name">
  </div>
</div>

<div class="form-group">
  <div class="form-col-4">
    <label for="lname">Last Name</label>
  </div>
  <div class="form-col-8">
    <input type="text" id="lname" autocomplete="family-name">
  </div>
</div>

<div class="form-group">
  <div class="form-col-4">
   <label for="favorite-fruit">Favorite fruit</label>
  </div>
  <div id="favorite-fruit" class="form-col-8">
    <select>
      <option>Banana</option>
      <option>Pineapple</option>
      <option>Strawberry</option>
    </select>
  </div>
</div>

<div class="form-group">
  <div class="offset-form-col-4 form-col-8">
    <button>Submit</button>
  </div>
</div>
```

Testing Procedure

1. Display the web page in a user agent capable of 400% zoom and set the viewport dimensions (in CSS pixels) to 1280 wide and 1024 high.
2. Zoom in by 400%.
3. For vertically scrolling content, all labels and inputs fit in their available space without horizontal scrolling.

NOTE: If the browser is not capable of zooming to 400%, you can reduce the width of the browser proportionally. For example, at 300% zoom the viewport should be sized to 960px wide.

Technique SCR34: Calculating size and position in a way that scales with text size

Examples

{See Above}

Testing Procedure

{See Above}

Technique G206: Providing options within the content to switch to a layout that does not require the user to scroll horizontally to read a line of text[104]

[104] https://www.w3.org/WAI/WCAG21/Techniques/general/G206.html

Example 1
A real estate company has an online annual report that has an identical layout to that of their print version, and as such, requires horizontal scrolling to read a line of text. A control is on the page that switches the stylesheet and provides a layout that does not require horizontal scrolling.

Example 2
A financial spreadsheet is online. It includes text explaining changes in the housing market in January. Off-screen to the right, there is a column with an explanation of changes to the market in September. The user can horizontally scroll to the September area and read each line of text without any further scrolling when the window size is maximized.

Testing Procedure
1. Open the content that requires horizontal scrolling on a full screen window.
2. Check that there is an option within the content to switch to a layout that does not require the user to scroll horizontally to read a line of text.
3. Activate the option.
4. Check to make sure that horizontal scrolling is not required to read any line of text.

1.4.11 Non-Text Contrast (Level AA)[105] – According to WGAC-EM, low contrast controls are more challenging to see and may go entirely unnoticed by those who have vision impairments. Like this, if a graphic is required to comprehend the information or functioning of a webpage, it should be readable without the use of contrast-enhancing assistive technology for individuals with low vision or other disabilities. Although it is not a requirement for this success criterion that controls have a visual boundary showing the hit region, if the control's visual indicator is the sole method to recognize it, it must have enough contrast. If there is no obvious sign of the hit region and text (or an icon) within a button or placeholder text inside a text entry is visible, the success criterion is met. There is no additional contrast requirement beyond the text contrast specified in 1.4.3 Contrast (Minimum) if a button with text additionally has a colored border because the border does not serve as the lone indicator. Notably, it is advised to mark the limits of controls for those with cognitive disorders to facilitate control detection and, as a result, activity completion.

User Interface Component contrast:

Technique G195: Using an author-supplied, visible focus indicator[106]

Example 1: Links
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 2: Form Elements
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. The red is equivalent to a hex color of #FF3838, providing a 3.6:1 contrast ratio with the white background.

Example 3: Menus
A Web page includes an interactive menu with sub-menus. A user can move focus in the menu using the arrow keys. As focus moves, the currently focused menu item changes its background to a different color, which has a 3:1 contrast ratio with the surrounding items and a 4.5:1 contrast ratio with its own text.

Testing Procedures
For each user interface component on the page that should receive keyboard focus:
1. Navigate to the component and check that it has a visible focus indicator.
2. Check that the focus indicator area is at least the size of a 1 CSS px border around the component.

[105] https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast.html
[106] https://www.w3.org/WAI/WCAG21/Techniques/general/G195.html

3. If the focus indicator area is not at least equal to the area of a 1 CSS pixel border, check that it has an area of at least 4 CSS pixels along the shortest side of the component.
4. Check that the change of contrast of the indicator between focused and unfocused states has a ratio of 3:1 or more for the minimum focus indicator area.
5. If the focus indicator does not have 3:1 contrast ratio with its adjacent colors, check that it is at least 2px thick.

Graphics with sufficient contrast:

Technique G207: Ensuring that a contrast ratio of 3:1 is provided for icons[107]

Example 1: Solid icon color against the background
A solid icon such as a telephone symbol uses orange on a white background. The color orange (#E3660E) is tested against the white background (#FFFFFF) and it has a contrast ratio of 3.4:1.

Example 2: Solid icon color against a custom background
A solid icon such as a telephone symbol used within an orange background. The orange and white colors are the same as in example 1, in this case the contrast against the white background is not relevant, the white icon within the orange background is what provides the information in the icon and as a result needs to meet the contrast requirement.

Example 3: Solid icon with a gradient background
A solid icon such as a telephone symbol using a dark blue icon on a white-to-blue gradient background. The first test of the icon should be against the darkest (least contrasting) background that is adjacent to the icon color. If that is at least 3:1, it passes the success criterion.

Example 4: Solid icon with gradient background overlapping in contrast
A solid icon on a gradient background can overlap in contrast if the graphic is still understandable where it does not have contrast against all of the background. If you find the part of the gradient where it does not meet a 3:1 ratio with the graphic and treat that part as if it is removed, does the icon still convey the appropriate meaning?
A method of visualizing this is to remove the non-contrasting area and check that you can still understand the icon. If so, it is sufficient. The images below shows an icon on a gradient background, and a second version where it removes the area of the icon that does not meet the 3:1 contrast ratio. It is still recognizable as a phone icon, so passes the success criterion.

Testing Procedures
For each graphical object required for understanding use a color contrast tool to:
1. Determine the foreground color of the graphical object.
2. Determine the adjacent background color. If the background color is a gradient or pattern, identify the color with the least contrast to the foreground color.
3. Check that the contrast ratio is equal to or greater than 3:1.
4. If part of the background area does not meet 3:1 with the foreground, assume that parts of the icon adjacent to the area or areas are not visible.
5. Check that the icon is still recognizable without any area of insufficient contrast.

Technique G209: Provide sufficient contrast at the boundaries between adjoining colors[108]

Testing Procedures
For each graphical object required for understanding use a color contrast tool to:
1. Measure the contrast ratio of each color compared to the adjacent color(s) or border (if present).
2. Check that the contrast ratio is at least 3:1 for each adjacent color or border (if present).

Text in or over graphics:

[107] https://www.w3.org/WAI/WCAG21/Techniques/general/G207.html
[108] https://www.w3.org/WAI/WCAG21/Techniques/general/G209.html

Technique G18: Ensuring that a contrast ratio of at least 4.5:1 exists between text (and images of text) and background behind the text

Examples
{See Above}
Testing Procedure
{See Above}

Technique G145: Ensuring that a contrast ratio of at least 3:1 exists between text (and images of text) and background behind the text

Examples
{See Above}
Testing Procedure
{See Above}

Technique G174: Providing a control with a sufficient contrast ratio that allows users to switch to a presentation that uses sufficient contrast

Examples
{See Above}
Testing Procedure
{See Above}

1.4.12 Text Spacing (Level AA)[109] – This Success Criteria is concerned with how well information can adjust to wider spaces between lines, words, letters, and paragraphs. Any of these in combination may help a user read text efficiently. The possibility that users can overrule the author's spacing preferences is also considerably increased by making sure that the material adapts effectively when users do so. For instance, to read content clearly, a user might need to switch to a broader font family than the author has chosen.

Technique C36: Allowing for text spacing override[110]

Examples: A paragraph expands vertically within container

```
/* CSS: No height property is set.*/

<!-- HTML -->
 <div class="card">
  <img src="image.png" alt="proper alt text">
  <h3>Heading</h3>
  <p class="lede">Long lede paragraph…</p>
 </div>
```

None of the paragraphs on this page have a height specified, so all are effectively using this technique.

Testing Procedure
For elements which contain text that is intended to wrap:
1. Set zoom level to 100%.
2. Use a tool or another mechanism to apply the text spacing metrics (line height, and paragraph, letter, and word spacing), such as the Text Spacing Bookmarklet or a user-style browser plugin.
3. Check that all content and functionality is available e.g., text in containers is not truncated and does not overlap other content.

[109] https://www.w3.org/WAI/WCAG21/Understanding/text-spacing.html
[110] https://www.w3.org/WAI/WCAG21/Techniques/css/C36.html

Technique C35: Allowing for text spacing without wrapping[111]

Example 1: A box sized with space to allow for expansion
The containers are sized to a value greater than the default width of the text.

```
/* Links are less than 8ex wide, so 10ex width of each li allows for expanded letter and word width*/
 nav li { width: 10em; }

<!-- HTML -->
 <nav>
  <ul>
   <li><a href="/">Home</a></li>
   <li><a href="/contact/">Contact</a></li>
  <ul>
 </nav>
```

If the navigation element used fix-width containers of the same size, the width would need to allow for text 20% larger than the longest word.

Example 2: A box which expands with the text size

```
/* CSS containers are given a display of inline-block. No negative margins set. */
 nav li { display: inline-block; }

<!-- HTML -->
 <nav>
  <ul>
   <li><a href="/">Home</a></li>
   <li><a href="/contact/">Contact</a></li>
  <ul>
 </nav>
```

In the case of variable-width text containers for each item, the parent item may need to allow for wrapping of the items.

Testing Procedure
For elements which contain text that is not intended to wrap:

1. Set zoom level to 100%.
2. Use a tool or another mechanism to apply the text spacing metrics (line height, and paragraph, letter, and word spacing), such as the Text Spacing Bookmarklet or a user-style browser plugin.
3. Check that all content and functionality is available e.g., text in containers is not truncated and does not overlap other content.

1.4.13 Content on Hover or Focus (Level AA)[112] – This requirement is there to make sure that the extra material won't make it difficult to see or use the page's original content. The amount of the page that can be seen in the viewport when it is magnified can be considerably diminished. Frequently, mouse users will pan the magnified viewport to show another area of the screen. It is challenging for a user to pan without re-triggering the new content because practically the entire area of the page displayed in this constrained viewport may trigger it. There is a fix by using the keyboard to remove the extra text.

- @@ ARIA: Using role="tooltip"
- @@ CSS: Using hover and focus pseudo classes

Section 2 – Operable

[111] https://www.w3.org/WAI/WCAG21/Techniques/css/C35.html
[112] https://www.w3.org/WAI/WCAG21/Understanding/content-on-hover-or-focus.html

2.1 Keyboard Accessible[113] – This required that all functionality within the target site is available from a keyboard. There are three criteria points in this section to reach level A conformance. For this section, there is not an option for level AA conformance – only A or AAA.

2.1.1 Keyboard (Level A)[114] – The purpose of this Success Criterion is to ensure that information could be accessed and used with a keyboard or keyboard interface whenever and wherever it is feasible to do so (so an alternate keyboard can be used). When content can be controlled by a keyboard or an alternative keyboard, it can be utilized by people who require the use of alternative keyboards or input devices that emulate keyboards, as well as by people who are blind and, as a result, are unable to control devices such as mice that require eye-hand coordination. Examples of keyboard emulators include speech input software, sip-and-puff, on-screen keyboards, scanning software, and a variety of assistive devices and alternative keyboards. Other types of keyboards include alternative keyboards. People who have trouble seeing clearly may have a difficult time following the movement of a pointer, and these individuals may be unable to use software unless it can be controlled via the keyboard. Certain situations qualify as "specific timings for individual keystrokes," such as those in which a user is required to quickly repeat or carry out a series of keystrokes, or in which a user must hold down a key for an extended period of time before the keystroke is recognized.

Technique G202: Ensuring keyboard control for all functionality[115]

Examples

- 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.
- A page that allows users to click and drag items in a list to reorder them also includes a series of controls that allows keyboard users to move items up, down or to the beginning and end of the list.
- The mobile version of a web site includes a menu button that is tapped to open a site menu, which is implemented as a floating overlay. To provide access to people using external keyboards or ability switches with their mobile device, the menu button and the site menu are both implemented such that they can be operated via the mobile device's keyboard interface.

Testing Procedure

1. Identify all functionality on the content.
2. Check that all functionality can be accessed using only the keyboard or keyboard interface.

Ensuring keyboard control by using one of the following techniques.

Technique H91: Using HTML form controls and links[116]

Example 1: Links

User agents provide mechanisms to navigate to and select links. In each of the following examples, the role is "link" from the <a href>. Note that <a name> does not provide a role of "link". The value is the URI in the 'href' attribute.

Example 1a

In example 1a, the name is the text inside the link, in this case "Example Site".

```
<a href="www.example.com">Example Site</a>
```

Example 1b

In example 1b of an image inside a link, the 'alt' attribute for the image provides the name. Some tools for viewing APIs, such as Microsoft Inspect Objects, will not surface this, but AT does.

```
<a href="www.example.com"><img src="example_logo.gif" alt="Example"></a>
```

[113] https://www.w3.org/WAI/WCAG21/quickref/?showtechniques=131%2C132%2C133%2C134%2C141%2C142%2C143%2C144%2C145%2C1410%2C1411%2C1412#keyboard-accessible

[114] https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html

[115] https://www.w3.org/WAI/WCAG21/Techniques/general/G202.html

[116] https://www.w3.org/WAI/WCAG21/Techniques/html/H91.html

Example 1c
In example 1c, some assistive technology will not automatically insert a space character when concatenating the image's alt text and the text of the link. If the text should not be concatenated without a space, it is safest to insert a space between the image and the adjacent word so that words will not run together.
```
<a href="www.example.com"><img src="example_logo.gif" alt="Example"> Text</a>
```

Example 2: Buttons
There are several ways to create a button in HTML, and they all map to the "push button" role.

Example 2a
In example 2a, the text is contained in the button element, in this case "save", as the name. There is no value.
```
<button>Save</button>
```

Example 2b
Example 2b uses the 'value' attribute, in this case "Save", "Submit", or "Reset" as the name.
```
<input type="button" value="Save" />
<input type="submit" value="Submit" />
<input type="reset" value="Reset" />
```

Example 2c
Example 2c uses the 'alt' attribute, in this case "save", as the name.
```
<input type="image" src="save.gif" alt="save" />
```

Example 2d
In example 2d, there is no 'alt' attribute so the 'title' attribute, in this case "save", is used as the name.
```
<input type="image" src="save.gif" title="save" />
```

Example 2e
Example 2e clarifies how the user agent determines the name if the author specifies both the 'alt' and 'title' attributes of the input element. In this case, the user agent uses the 'alt' attribute ("Save") and ignores the 'title' attribute.
```
<input type="image" src="save.gif" alt="save" title="save the file" />
```

Example 3

Example 3a
In example 3a, the input field has a role of "editable text". The label element is associated to the input element via the 'for' attribute which references the 'id' attribute of the input> element. The name comes from the label element, in this case, "Type of fruit". Its value comes from its value attribute, in this case "bananas".
```
<label for="text_1">Type of fruit</label>
<input id="text_1" type="text" value="bananas">
```

Example 3b
In example 3b, the input field has the same role as example 3a, but the value is the empty string and the field gets its name from the 'title' attribute.
```
<input id="text_1" type="text" title="Type of fruit">
```

Example 4: Checkbox
Example 4 has a role of "checkbox", from the 'type' attribute of the input element. The label element is associated with the input element via the 'for' attribute which refers to the 'id' attribute of the input element. The name comes from the label element, in this case "cheese". Its state can be "checked" or "unchecked" and comes from the 'checked' attribute. The state can be changed by the user's interaction with the control.
```
<label for="cb_1">Cheese</label>
<input id="cb_1" type="checkbox" checked="checked">
```

Example 5: Radio Buttons
Example 5 has a role of "radio button" from the 'type' attribute on the input element. Its name comes from the label element. The state can be "checked" or "unchecked" and comes from the 'checked' attribute. The state can be changed by the user.

```
<input type="radio" name="color" id="r1" checked="checked"/><label for="r1">Red</label>
<input type="radio" name="color" id="r2" /><label for="r2">Blue</label>
<input type="radio" name="color" id="r3" /><label for="r3">Green</label>
```

Example 6

Example 6a
Example 6a has a role of "list box" from the select element. Its name is "Numbers" from the label element. Forgetting to give a name to the select is a common error. The value is the option element that has the 'selected' attribute present (with a value of "selected" in XHTML). In this case, the default value is "Two".

```
<label for="s1">Numbers</label>
<select id="s1" size="1">
 <option>One</option>
 <option selected="selected">Two</option>
 <option>Three</option>
</select>
```

Example 6b
Example 6b has the same name, role, and value as the above, but sets the name with the 'title' attribute on the select element. This technique can be used when a visible label is not desirable.

```
<select id="s1" title="Numbers" size="1">
 <option>One</option>
 <option selected="selected">Two</option>
 <option>Three</option>
</select>
```

Example 7: Textarea

Example 7a
Example 7a has a role of "editable text" from the textarea element. The name is "Type your speech here" from the label element. The value is the content inside the textarea element, in this case "Four score and seven years ago".

```
<label for="ta_1">Type your speech here</label>
<textarea id="ta_1" >Four score and seven years ago</textarea>
```

Example 7b
Example 7b has the same role, the name is set using the 'title' attribute, and the value is the empty string.

```
<textarea id="ta_1" title="Type your speech here" >Four score and seven years ago</textarea>
```

Example 8
Radio Fieldset
The radio fieldset in example 8 has a role of "grouping". The name comes from the legend element.

```
<fieldset>
  <legend>Choose a Color:</legend>
     <input id="red" type="radio" name="color" value="red" /><label for="red">Red</label><br
/>
     <input      id="blue"      type="radio"      name="color"      value="blue"      /><label
for="blue">Blue</label><br />
     <input     id="green"     type="radio"     name="color"     value="green"     /><label
for="green">Green</label>
</fieldset>
```

Technique G90: Providing keyboard-triggered event handlers[117]

Examples

- **Example 1: A drag and drop feature** A photo application includes a "drag" and "drop" feature to allow users to re-order photographs in an on-line album for presentation as a slide show. It also includes a feature that allows users to select a photo and 'cut' and 'paste' the items into the list at the appropriate point using only the keyboard.

[117] https://www.w3.org/WAI/WCAG21/Techniques/general/G90.html

- **Example 2: A reorder feature** A Web application that allows users to create surveys by dragging questions into position includes a list of the questions followed by a text field that allows users to re-order questions as needed by entering the desired question number.

Testing Procedure

1. check that all functionality can be accessed using only the keyboard

Technique SCR20: Using both keyboard and other device-specific functions[118]

Example 1

In this example of an image link, the image is changed when the user positions the pointer over the image. To provide keyboard users with a similar experience, the image is also changed when the user tabs to it.

```
<a href="menu.php" onmouseover="swapImageOn('menu')" onfocus="swapImageOn('menu')"
onmouseout="swapImageOff('menu')" onblur="swapImageOff('menu')">
  <img id="menu" src="menu_off.gif" alt="Menu" />
</a>
```

Example 2

This example shows a custom link control where both the mouse and the keyboard can be used to activate the function. The mouse onclick event is duplicated by an appropriate keyboard onkeypress event. The tabindex attribute ensures that the keyboard will have a tab stop on the span element. Note that in this example, the nextPage() function should check that the key pressed is Enter, otherwise it will respond to all keyboard actions while the span has focus, which is not the desired behavior.

```
<span onclick="nextPage();" onkeypress="nextPage();" role="link" tabindex="0">
  <img alt="Go to next page" src="arrow.gif">
</span>
```

Testing Procedure

1. Find all interactive functionality
2. Check that all interactive functionality can be accessed using the keyboard alone

Technique SCR35: Making actions keyboard accessible by using the onclick event of anchors and buttons[119]

Example 1

Link that runs a script and has no fallback for non-scripted browsers. This approach should only be used when script is relied upon as an Accessibility Supported Technology.
Even though we do not want to navigate from this link, we must use the href attribute on the a element in order to make this a true link and get the proper eventing. In this case, we're using "#" as the link target, but you could use anything. This link will never be navigated.
The "return false;" at the end of the doStuff() event handling function tells the browser not to navigate to the URI. Without it, the page would refresh after the script ran.

```
<script>
function doStuff()
 {
  //do stuff
    return false;
  }
</script>
<a href="#" onclick="return doStuff();">do stuff</a>
```

Example 2

Link that runs script, but navigates to another page when script is not available. This approach can be used to create sites that don't rely on script, if and only if the navigation target provides the same functionality as the script. This example is identical to the example 1, except that its href is now set to a real page, dostuff.htm. Dostuff.htm

[118] https://www.w3.org/WAI/WCAG21/Techniques/client-side-script/SCR20.html
[119] https://www.w3.org/WAI/WCAG21/Techniques/client-side-script/SCR35.html

must provide the same functionality as the script.The "return false;" at the end of the doStuff() event handling function tells the browser not to navigate to the URI. Without it, the browser would navigate to dostuff.htm after the script ran.

```
<script>
function doStuff()
 {
  //do stuff
  return false;
 }
</script>
<a href="dostuff.htm" onclick="return doStuff();">do stuff</a>
```

A working example of this code is available. Refer to Creating Action Links using JavaScript.

Example 3
Button that runs a script and falls back to a form post for users without script. This approach can be used by sites that do not rely on script, if and only if the form post provides the same functionality as the script. The onsubmit="return false;" prevents the form from submitting.

```
<script>
  function doStuff()
 {
     //do stuff
 }
</script>
<form action="doStuff.aspx" onsubmit="return false;">
 <input type="submit" value="Do Stuff" onclick="doStuff();" />
</form>
```

A working example of this code is available. Refer to Creating Action Buttons using JavaScript.

Example 4
Button that runs a script, implemented with input `type="image".` Note that an alt attribute must be added to the `input` to provide a text equivalent for the image. This approach should only be used when script is relied upon.

```
<script>
  function doStuff()
  {
     //do stuff
   return false;
  }
</script>
<input  type="image"  src="stuff.gif"  alt="Do stuff"  onclick="return doStuff();" />
```

Example 5
Button that runs a script, implemented with input `type="submit"`, `input type="reset"` or `input type="button"`. This approach should only be used when script is relied upon.

```
<input type="submit" onclick="return doStuff();" value="Do Stuff" />
```

Example 6
Button that runs a script, implemented with button…/button. This is valuable when you want more control over the look of your button. In this particular example, the button contains both an icon and some text. This approach should only be used when script is relied upon.

```
<button onclick="return doStuff();">
 <img src="stuff.gif" alt="stuff icon">
 Do Stuff
</button>
```

Testing Procedure

1. In a user agent that supports Scripting
   - Click on the control with the mouse.
   - Check that the scripting action executes properly.
   - If the control is an anchor element, check that the URI in the `href` attribute of the anchor element is not invoked.
   - Check that it is possible to navigate to and give focus to the control via the keyboard.
   - Set keyboard focus to the control.
   - Check that pressing ENTER invokes the scripting action.
   - If the control is an anchor element, check that the URI in the href attribute of the anchor element is not invoked.
2. In a user agent that does not support Scripting
   - Click on the control with the mouse.
   - If the control is an anchor element, check that the URI in the href attribute of the anchor element is invoked.
   - Check that it is possible to navigate to and give focus to the control via the keyboard.
   - Set keyboard focus to the control.
   - If the control is an anchor element, check that pressing ENTER invokes the URI of the anchor element's `href` attribute.

Technique SCR2: Using redundant keyboard and mouse event handlers[120]

Examples

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Changing Image Source in a Device Independent Manner</title>
  <script>
    /* This function will change the image src of an image element.
     * param imgId - the id of the image object to change
     * param isOver - true when mouse is over or object has focus,
      false when mouse move out or focus is lost
    */
    function updateImage(imgId, isOver) {
    var theImage = document.getElementById(imgId);
    if (theImage != null) {
      if (isOver) {
        theImage.setAttribute("src","yellowplus.gif");
      }
      else {
        theImage.setAttribute("src","greyplus.gif");
      }
    }
  }
  </script>
 </head>
 <body>
 <p>Mouse over or tab to the links below and see the image change.</p>
 <a href="https://www.w3.org/WAI/"
    onmouseover="updateImage('wai', true);" onfocus="updateImage('wai', true);"
    onmouseout="updateImage('wai',false);" onblur="updateImage('wai',false);">
    <img alt="" id="wai" src="greyplus.gif">
    W3C Web Accessibility Initiative</a> &amp;
 <a href="https://www.w3.org/International/" onmouseover="updateImage('i18n', true);"
    onfocus="updateImage('i18n',true);" onmouseout="updateImage('i18n',false);"
    onblur="updateImage('i18n',false);">
    <img alt="" id="i18n" src="greyplus.gif">
    W3C Internationalization</a>
 </body>
 </html>
```

[120] https://www.w3.org/WAI/WCAG21/Techniques/client-side-script/SCR2.html

Testing Procedure

Load the Web page and test the events using a mouse and via the keyboard.

1. Check that the "standard" image is displayed as expected when the Web page is loaded.
2. Using the Mouse
    1. Move the mouse over the element containing the event handlers (in this example it is an anchor element). Check that the image changes to the expected image.
    2. Move the mouse off of the element. Check that the image changes back to the "standard" image.
3. Using the Keyboard
    1. Use the keyboard to set focus to the element containing the event handlers. Check that the image changes to the expected image.
    2. Use the keyboard to remove focus from the element (generally by moving focus to another element). Check that the image changes to the "standard" image.
4. Verify that the layout of other elements on the page is not affected when the image is changed.

2.1.2 No Keyboard Trap (Level A)[121] – The purpose of this Success Criteria is to eliminate the possibility of keyboard focus becoming "stuck" within specific sections of content on a web page. This is a common problem that arises if multiple file formats are combined on a single page in a manner that is produced by plug-ins or other embedded programs. The functionality of the website may, on occasion, restrict the user's focus to a certain section of the material; however, as long as the user is aware of how to "untrap" the focus, this limitation should not affect the user's experience.

Technique G21: Ensuring that users are not trapped in content[122]

Examples

- Once a user tabs into an applet, further tabs are handled by the applet preventing the person from tabbing out. However, the applet is designed so that it returns keyboard focus back to the parent window when the person finishes tabbing through the tab sequence in the applet.
- A page that includes content that is not accessibility-supported contains instructions about how to move focus back to the accessibility-supported content via the keyboard. The instructions precede the non accessibility-supported content.
- The help information available from the content that is not accessibility supported documents how to move focus back to the accessibility-supported content via the keyboard, and the help information can be accessed via the keyboard.
- The help information available for the Web page documents how to move focus from the content that is not accessibility supported to the accessibility-supported content via the keyboard, and the help information can be accessed via the keyboard.

Testing Procedure

1. Tab through content from start to finish.
2. Check to see that keyboard focus is not trapped in any of the content.
3. If keyboard focus appears to be trapped in any of the content, check that help information is available explaining how to exit the content and can be accessed via the keyboard.

2.1.4 Character Key Shortcuts (Level A)[123] – The purpose of this Success Criterion is to reduce the chances of inadvertently using a keyboard shortcut by making the shortcuts less obvious. Users of speech input, who enter words as strings of letters, and users of keyboards who frequently touch keys by accident find character key shortcuts inefficient and bothersome. Character key shortcuts, on the other hand, are helpful for users of keyboards who touch keys frequently by accident. In order to solve this issue, authors need to provide users the opportunity to turn off or customize shortcuts that solely use character keys.

[121] https://www.w3.org/WAI/WCAG21/Understanding/no-keyboard-trap.html
[122] https://www.w3.org/WAI/WCAG21/Techniques/general/G21.html
[123] https://www.w3.org/WAI/WCAG21/Understanding/character-key-shortcuts.html

Sufficient Techniques:

- Users have a way to turn off single-key shortcuts.
- A mechanism is provided to allow users to change character-key shortcuts. The remapping mechanism includes non-printing characters. The alternative shortcuts could be longer strings of up to 25 characters that would directly serve as native speech commands for any speech engine.

2.2 Enough Time[124] – The purpose of this section is to ensure that users interacting with the target site had enough time to read and use the content. There are two criteria points for this section in order to obtain level A conformance. For this section, there is not an option for level AA conformance – only A or AAA.

2.2.1 Timing Adjustable (Level A)[125] – The purpose of this Success Criterion is to make certain that users with disabilities are provided with sufficient time to interact with material on the website whenever it is feasible to do so. Reading text or performing duties such as filling out online forms may take longer for people with disabilities such as blindness, low vision, dexterity difficulties, and cognitive limitations. These people may require additional time. If certain Web functions are time-sensitive, it will be difficult for some users to complete the task that is necessary before the timer reaches its maximum. Because of this, it's possible that they won't be able to use the service. People with impairments will have a better chance of completing functions successfully if these functions are designed so that they are not time dependent. It is helpful for users who require more time than expected to effectively accomplish activities to have the ability to disable time limitations, alter the length of time restrictions, or request more time before a time limit occurs. These choices are presented in the order in which the user will find them to be of the greatest assistance to them. Turning off time restrictions is preferable to adjusting the length of time limits, which in turn is preferable to asking additional lead time before a time limit kicks in.

Situation A: If there are session time limits:

Technique 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[126]

Examples
Testing Procedure

Technique G198: Providing a way for the user to turn the time limit off[127]

Examples
Testing Procedure

Situation B: If a time limit is controlled by a script on the page:

Technique G198: Providing a way for the user to turn the time limit off

Examples
{See Above}
Testing Procedure
{See Above}

[124] https://www.w3.org/WAI/WCAG21/quickref/?showtechniques=131%2C132%2C133%2C134%2C141%2C142%2C143%2C144%2C145%2C1410%2C1411%2C1412#enough-time
[125] https://www.w3.org/WAI/WCAG21/Understanding/timing-adjustable.html
[126] https://www.w3.org/WAI/WCAG21/Techniques/general/G133.html
[127] https://www.w3.org/WAI/WCAG21/Techniques/general/G198.html

Technique G180: Providing the user with a means to set the time limit to 10 times the default time limit[128] ***Examples*** ***Testing Procedure***
Technique SCR16: Providing a script that warns the user a time limit is about to expire[129] AND Technique SCR1: Allowing the user to extend the default time limit[130] ***Examples for SCR16:*** ***Testing Procedure SCR16:*** ***Examples SCR1:*** ***Testing Procedure SCR1:***

Situation C: If there are time limits on reading:

Technique G4: Allowing the content to be paused and restarted from where it was paused[131] ***Examples*** ***Testing Procedure***
Technique G198: Providing a way for the user to turn the time limit off ***Examples*** {See Above} ***Testing Procedure*** {See Above}
Technique SCR33: Using script to scroll content, and providing a mechanism to pause it[132] ***Examples*** ***Testing Procedure***
Technique SCR36: Providing a mechanism to allow users to display moving, scrolling, or auto-updating text in a static window or area[133] ***Examples*** ***Testing Procedure***

2.2.2 Pause, Stop, Hide (Level A)[134] – When visitors are interacting with the target site, this Success Criterion's main objective is to ensure that they did not become sidetracked in any way. Content is said to be "moving, blinking, and scrolling" when it gives the impression of being in motion just by its appearance. A few common examples include moving visuals, synchronized media presentations, animations, real-time games, and scrolling financial tickers. The term "auto-updating" refers to content that alters or disappears at

[128] https://www.w3.org/WAI/WCAG21/Techniques/general/G180.html
[129] https://www.w3.org/WAI/WCAG21/Techniques/client-side-script/SCR16.html
[130] https://www.w3.org/WAI/WCAG21/Techniques/client-side-script/SCR1.html
[131] https://www.w3.org/WAI/WCAG21/Techniques/general/G4.html
[132] https://www.w3.org/WAI/WCAG21/Techniques/client-side-script/SCR33.html
[133] https://www.w3.org/WAI/WCAG21/Techniques/client-side-script/SCR36.html
[134] https://www.w3.org/WAI/WCAG21/Understanding/pause-stop-hide.html

regular intervals according to a schedule that the user specifies. Examples of time-based content include audio, automatically updated weather reports, news, automatically progressing presentations and messaging, and stock price updates. The requirements for content that is moving, blinking, or scrolling and the requirements for content that is automatically updating are equivalent, with the exception that authors may give users a way to regulate the frequency of updates when content is auto-updating, and there is no five-second exception for auto-updating because it is ineffective to auto-update for a short period of time before ceasing to do so.

2.3 Seizures and Physical Reactions[135] – This section worked to ensure that the target site did not design its content in a manner that would be known to cause seizures or any other physical reactions. There is one criterion point for this section to be level A. For this section, there is not an option for level AA conformance – only A or AAA.

2.3.1 Three Flashes or Below Threshold (Level A)[136] – Users should be able to access the entirety of a website without having their photosensitive seizures triggered in order for this Success Criterion to be considered met. People who have photosensitive seizure disorders are more likely to have seizures if they are exposed to information that flashes at certain frequencies for a longer period of time than a few times. Because people are significantly more sensitive to intense red flashing than they are to other colors, a separate test has been developed specifically for it. Rules from the broadcasting sector served as the foundation for these guidelines, which are later adapted for desktop displays, on which content is viewed more closely (using a larger angle of vision). Flashing might be caused by the display itself, the computer that is rendering the image, or the material that is being created. The author does not have any influence over the first two points. It is possible to address issues by using the layout of the computer and display as well as the processing capability of the computer. This criterion is developed to ensure that flickering that is more severe than the flash threshold is not caused by the actual material being displayed on the screen. A video clip, an animated graphic, or even close-up shots of a series of strobe lights or explosions could be used as part of the material, for instance.

[135] https://www.w3.org/WAI/WCAG21/quickref/?showtechniques=131%2C132%2C133%2C134%2C141%2C142%2C143%2C144%2C145%2C1410%2C1411%2C1412#seizures-and-physical-reactions

[136] https://www.w3.org/WAI/WCAG21/Understanding/three-flashes-or-below-threshold.html