Accessible Accordion

Date Developed: 1/31/2014

Last Modified: 8/1/16

# Example URLs:

<http://www.csun.edu/afvp>

<http://www.csun.edu/academic-affairs>

# Summary:

This document describes the technical and analytical aspects of creating accessible Accordion tabs. It introduces a coding methodology consists of two major techniques that are used to overcome challenges in developing an accessible Accordion.

# Proper Navigational Instruction

## Purpose:

Users should be properly acknowledged when they land on an accordion region while navigating through the web page. They should be informed of the general behavior of the navigational structure that they are about to experience before they start moving through the accordion tabs. In addition, a descriptive message should be provided to describe the navigational hints specific to the accordion region.

## Technique:

This technique provides an instruction to the Assistive Technology users which is announced when the user lands on the Accordion region. The instruction briefly explains how to interact with the accordion tabs. No ARIA attribute is used to embed the instruction provided to the Assistive Technology; instead the instruction is controlled by jQuery and is appended to the HTML when the user lands on the accordion region. Therefore it works consistently on all common browsers (Firefox, Internet Explorer and Chrome). In Chrome users may hear an additional message conveying the same meaning that is generated by Assistive Technology itself.

## Source Code

### jQuery:

<script src="<http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js>"></script>

<script>

$(document).ready(function(){

  $("# accordion-instruction").focusin(function(){

     $("#instruction").append("You have reached an accordion control. The following tabs will be activated by spacebar");

// making invisible content as instruction that is available and readable by A.T.

     $("#instruction").css({"position": "absolute", "clip": "rect(1px, 1px, 1px, 1px)"});

 });

   $("# accordion-instruction ").focusout(function(){

  $("#instruction").empty();

  });

$(".guide").append("To activate tab page press spacebar.");

// making invisible content as instruction that is available and readable by A.T.

$(".guide").css({"position": "absolute", "clip": "rect(1px, 1px, 1px, 1px)"});

});

</script>

---------------------------------------------------------------------------------------------------

### HTML:

<body>

This is a link before starting the Accordion region:

<a href="  "> a link </a>

<h1>

IT FAQs for Students

</h1>

</div>

<div class="layout-csun">

<div class="layout-csun--columns layout-csun--columns-left">

<div class="panel-separator"></div><div class="panel-pane pane-node-content" >

<article class="node-1441 node node-page view-mode-full clearfix" about="/it/it-faqs-students" typeof="foaf:Document">

<div id="accordion-instruction" tabindex="0">

<span id="instruction"> </span>

   <div id="accordion" class="ui-accordion ui-widget ui-helper-reset ui-accordion-icons" role="tablist">

 <h2 class="field field-name-field-title-text field-type-text field-label-hidden ui-accordion-header ui-helper-reset ui-state-default ui-corner-all" role="tab" aria-expanded="false" tabindex="0">

    <span class="ui-icon ui-icon-triangle-1-e"></span>

     College Transcripts

     <span class="guide"></span>

   </h2>

</div>

</div>

</article

# Proper keyboard Tab Status Announcement

## Purpose:

Any Accordion tab is expanded when it is received focus and activated. Users should be properly acknowledged when the current Accordion tab is activated. The status of an Accordion tab is defined as expanded/open or collapsed/close. Therefore, while the user is navigating through the Accordion tabs, she should be properly acknowledged if any one tab is open or close; therefore she can navigate to the content panel of the tab.

## Technique:

This technique utilizes **ARIA** attributes and properties to make each Accordion tab’s status announced properly to the user agent. By using proper ARIA markup the current status of the focused Accordion item is reflected to the user agent. This technique is used to change the visually hidden status of the accordion when the user activates (expand/collapse) any menu item of the accordion. The jQuery handles the changes of the accordion status by using the technique of clipping absolute position of the element. This involves controlling the CSS properties of the accordion items. By using aria-atomic attribute, it will signal the user agent to examine the changed element and return the changes (for example, the screen reader will announce “Universal Design Center Expanded” when user expands the accordion menu item).

### Definitions:

aria-live property: Indicates that an element will be updated, and describes the types of updates the user agents, assistive technologies, and user can expect from the live region. With the politeness value of this property, the changed content of the region will be read without any interruption of the current speaking.

aria-atomic property: When the live region changes, user agents/assistive technologies examine the changed element and apply the appropriate behavior. If aria-atomic is explicitly set to true, assistive technologies will present the entire changed contents of the element.

aria-expanded attribute: Indicates whether the element, or another grouping element it controls, is currently expanded or collapsed.

aria-hidden attribute:Indicates that the element and all of its descendants are not visible or perceivable to any user.

## Source Code

### jQuery:

<script>

$(document).ready (function () {

$(".trigger").keypress (function (event) //using keyboard

{

if ( event.which == 13 || event.which == 9 )

{

var $indicator = $(this).find(".indicator:first");

var $content = $(this).next (".content");

if ( $indicator.attr("**aria-expanded**") == "false")

{

$indicator.attr ("**aria-expanded**", "true");

$indicator.text ("Expanded");

$indicator.css({"position": "absolute", "clip": "rect(1px, 1px, 1px, 1px)"});

$content.show () .attr ("**aria-hidden**", "false");

}

else if ( $indicator.attr("**aria-expanded**") == "true")

{

$indicator.attr ("**aria-expanded**", "false");

$indicator.text ("Collapsed");

$indicator.css({"position": "absolute", "clip": "rect(1px, 1px, 1px, 1px)"});

$content.hide()

.attr ("**aria-hidden**", "true");

} // if

}// if

}) // function

$(".trigger").click (function () { //using mouse

var $indicator = $(this).find(".indicator:first");

var $content = $(this).next (".content");

if ( $indicator.attr("**aria-expanded**") == "false") {

$indicator.attr ("**aria-expanded**", "true");

$indicator.text ("Expanded");

$indicator.css({"position": "absolute", "clip": "rect(1px, 1px, 1px, 1px)"});

$content.show ().attr ("**aria-hidden**", "false");

} else if ( $indicator.attr("**aria-expanded**") == "true") {

$indicator.attr ("**aria-expanded**", "false");

$indicator.text ("Collapsed");

$indicator.css({"position": "absolute", "clip": "rect(1px, 1px, 1px, 1px)"});

$content.hide()

.attr ("**aria-hidden**", "true");

} // if

//$indicator.focus();

}); // click .trigger

}); // ready

//alert("loaded");

</script>

--------------------------------------------------------------------------------------

### HTML:

<div id="expandingSectionsContainer">

<h2 class="trigger" **aria-atomic="true"** **aria-live="polite"** tabindex="0">

Universal Design Center <span **aria-expanded="false"** class="indicator"></span>

</h2>

<div class="content" **aria-hidden="true"**>

<p>Here is more expanded content.</p>

<p>Here is more expanded content.</p>

<p>Here is more expanded content.</p>

<ul>

<li><a href="[http://www.w3.org](view-source:http://www.w3.org/)">The World Wide Web Consortium</a></li>

<li><a href="[http://www.w3.org/TR/wai-aria/](view-source:http://www.w3.org/TR/wai-aria/)">ARIA 1.0</a></li>

</ul>

</div>

<h2 class="trigger" **aria-atomic="true"** **aria-live="polite"** tabindex="0">

Information Technology <span **aria-expanded="false"** class="indicator"></span>

</h2>

<div class="content" **aria-hidden="true"**>

<p>Here is more expanded content.</p>

<p>Here is more expanded content.</p>

<p>Here is more expanded content.</p>

<p>Here is more expanded content.</p>

</div>

<h2 class="trigger" **aria-atomic="true"** **aria-live="polite"** tabindex="0">

Accessibility <span **aria-expanded="false"** class="indicator"></span>

</h2>

<div class="content" **aria-hidden="true"**>

<p>Here is more expanded content.</p>

<p>Here is more expanded content.</p>

<p>Here is more expanded content.</p>

<p>Here is more expanded content.</p>

</div>

</div><!-- expandingSectionsContainer -->

# References:

* <http://www.w3.org/TR/wai-aria/states_and_properties#aria-atomic>
* <http://www.w3.org/TR/wai-aria/states_and_properties#aria-live>
* <http://webaim.org/techniques/css/invisiblecontent/>
* [http://ncamftp.wgbh.org/sp/tests/expandingSections/expandingSections.html#](http://ncamftp.wgbh.org/sp/tests/expandingSections/expandingSections.html) (National Center for Accessible Media)