Created by Charles Brandt

www.sizzlepopboom.com
www.animateinexile.blogspot.com

Accessible Class Documentation

Attributes:

reader

-- A Boolean; is true if a screen reader has been successfully detected and false if it has not.
--testScreenReader or detectScreenReader must be called to check for a screen reader; if a screen reader is detected, reader will be set to true
--without calling testScreenReader or detectScreenReader, reader will remain false even if a screen reader is present

Events:

READER_ACTIVE

-- dispatched one time when a screen reader is detected by the **detectScreenReader** method

Ex: stage.addEventListener("READER_ACTIVE", readerActiveHandler);

Methods:

setTab(item:InteractiveObject, tabIndex:int(optional))

-- Sets the items tabIndex.
-- If you are changing the tabIndex, it checks if the old tab is in its dictionary, if it is, it removes it. If something else occupies the desired tab, the occupant is moved to the next highest slot.

Ex: Accessible.setTab(myItem);
Accessible.setTab(myItem, 20); Both are valid.

getTab(item:InteractiveObject):int

-- Returns the item's current tabIndex as an int.
-- A tabIndex of -1 indicates that object's tabIndex has not yet been set.

Ex: trace(Accessible.getTab(myItem));

stopTab(item:InteractiveObject)

-- Stops tabbing on any object enabled with this class or through the Accessibility panel.

Ex: Accessible.stopTab(myItem);

startTab(item:InteractiveObject)

-Enables tabbing on an object.

Ex: Accessible.startTab(myItem);

showTabList()

-- Traces a list of all objects and their tabIndexes that have been assigned through the Accessible class.

Ex: Accessible.showTabList();

detectScreenReader(flashLocation)

-- Sees if a screen reader is running and communicating with Flash.

-- Controls this class's reader attribute and switches it to true if a reader is detected

-- Dispatches and event "READER_ACTIVE" when a screen reader is detected.

-- Once a reader is detected, this method turns itself off, otherwise it continues detecting on each keypress.

Note: it may take the screen reader a short time to show as active, so it's better to test for it on a keyPress to give it time to register.

-- You must pass a reference to the stage or an object on stage for this function to work (static classes can't access the stage, need stage access for the event handler).

Ex: Accessible.detectScreenReader(this);
Accessible.detectScreenReader(stage);
Accessible.detectScreenReader(myItem); All are valid.

testScreenReader():Boolean

-- Sees if a screen reader is running and communicating with Flash at a given moment.

-- Returns a boolean true if a screen reader is detected.

Note: This is a one-time check

Ex: trace(Accessible.testScreenReader());

setName(item:InteractiveObject,altName:String)
-- Set the alternative text name of an item.

Ex: Accessible.setName(myItem, “Name of my item”);

getName(item:InteractiveObject):String
-- Shows what a screen reader will read as this item's name.
-- Returns a string or undefined.

Ex: trace(Accessible.getName(myItem));

setDescrip(item:InteractiveObject,altDescrip:String)
-- Sets the alternative text description of an item

Ex: Accessible.setDescrip(myItem, “A description of myItem goes here.”);

getDescrip(item:InteractiveObject):String
-- Shows what a screen reader will read as this item's description.
-- Returns a string or undefined.

Ex: trace(Accessible.getDescrip(myItem));

setShortcut(item:InteractiveObject,shortCut:String)
-- Sets the shortcut keys for an object.
-- The syntax for the shortcut string uses long names for modifier keys, and the plus (+) character to indicate key combination. Examples of valid strings are "Ctrl+F", "Ctrl+Shift+Z", and so on.

Ex: Accessible.setShortcut(myItem, “Ctrl+F”);

getShortcut(item:InteractiveObject):String
-- Gets the shortcut keys for an object.
-- Returns a string or undefined.

Ex: trace(Accessible.getShortcut(myItem));

keyTrace(flashLocation)
-- Enables listening for keyPresses and writes trace statements that list the keycodes of buttons as they are pressed.

-- This was meant to be used to look up keycodes when and test if keypresses are registering when developing (thus, it does not turn off).
-- Argument (flashLocation) is reference to the stage or any object on the stage (this is necessary because it is a static method, and cannot access the stage without a reference).

Ex: Accessible.keyTrace(this);
Accessible.keyTrace(stage);
Accessible.keyTrace(myItem); All are valid.

makeReaderFriendly(htmlText:String):String

-- Removes any html tags from text so the tags will not be read by the screen reader, tags are defined as anything between < and >.
-- Text using < or > in addition to tags may have yield unpredictable results
-- Changes any lowercase " a " or " i " to uppercase so screen readers will pronounce it correctly.
-- Returns the parsed text as a String.
--**Caution** if the text contains a ”<” or “>” not part of a tag, the text may be altered in an unexpected manner.

Ex: trace(Accessible.makeReaderFriendly(“text to make reader-friendly goes here”));

stripTags(htmlText:String):String

-- Removes any html tags from text so the tags will not be read by the screen reader, tags are defined as anything between < and >.
--**Caution** if the text contains a ”<” or “>” not part of a tag, the text may be altered in an unexpected manner.

Ex: trace(Accessible.stripTags(“text to remove tags from goes here”));

Note:
Demos of JAWS (http://www.freedomscientific.com/jaws-hq.asp) and WindowEyes (http://www.gwmicro.com/Window-Eyes/) are available online. The NVDA screenreader is also available for free full version download at: http://www.nvda-project.org/
Currently Internet Explorer offers the best accessibility features of the popular web browsers.