ARIA Abstract Roles

command	A form of widget that performs an action but does not receive input data.
composite	A widget that may contain navigable descendants or owned children.
input	A generic type of widget that allows user input.
landmark	A region of the page intended as a navigational landmark.
range	An input representing a range of values that can be set by the user.
roletype	The base role from which all other roles in this taxonomy inherit.
section	A renderable structural containment unit in a document or application.
sectio-nhead	A structure that labels or summarizes the topic of its related section.
select	A form widget that allows the user to make selections from a set of choices.
structure	A document structural element.
widget	An interactive component of a graphical user interface (GUI).
window	A browser or application window.

The roles above are used to support the WAI-ARIA role taxonomy for the purpose of defining general role concepts.

Widget Roles

alert	A message with important, and usually time-sensitive, inform-ation. See related a-lertdialog and status.	alert()
alertd-ialog	A type of dialog that contains an alert message, where initial focus goes to an element within the dialog. See related alert and dialog.	
button	An input that allows for user-triggered actions when clicked or pressed. See related link.	`<button>` `<input type="submit">` `<input type="reset">` `<input type="image">`
checkbox	A checkable input that has three possible values: true, false, or mixed.	`<input type="checkbox">`

Widget Roles (cont)

combobox	A presentation of a select; usually similar to a textbox where users can type ahead to select an option, or type to enter arbitrary text as a new item in the list. See related listbox.	`<select>` HTML5 `<input>`
dialog	A dialog is an application window that is designed to interrupt the current processing of an application in order to prompt the user to enter information or require a response. See related alertdialog.	
grid	A grid is an interactive control which contains cells of tabular data arranged in rows and columns, like a table.	`<table>`
gridcell	A cell in a grid or treegrid.	`<td>`
link	An interactive reference to an internal or external resource that, when activated, causes the user agent to navigate to that resource. See related button.	`<a href...>` `<area>`
listbox	A widget that allows the user to select one or more items from a list of choices. See related combobox and list.	
log	A type of live region where new information is added in meaningful order and old information may disappear. See related marquee.	
marquee	A type of live region where non-essential information changes frequently. See related log.	
menu	A type of widget that offers a list of choices to the user.	

Widget Roles (cont)

menubar	A presentation of menu that usually remains visible and is usually presented horizontally.
menuitem	An option in a set of choices contained by a menu or menubar.
menuitemcheckbox	A values are true, false, or mixed.
menuitemradio	A checkable role menuitemradio, only one of which can be checked at a time.
option	A selectable item in a select list. `<option>`
progressbar	An element that displays the progress status for tasks that take a long time. `<progress>`
radio	A checkable input in a group of radio roles, only one of which can be checked at a time. `<input type="radio">`
radiogroup	A group of radio buttons.
scrollbar	A graphical object that controls the scrolling of content within a viewing area, regardless of whether the content is fully displayed within the viewing area.
slider	A user input where the user selects a value from within a given range. `<input type="range">`
spinbutton	A form of range that expects the user to select from among discrete choices.
status	A container whose content is advisory information for the user but is not important enough to justify an alert, often but not necessarily presented as a status bar. See related alert. `<output>`

Widget Roles (cont)

tab	A grouping label providing a mechanism for selecting the tab content that is to be rendered to the user.
tablist	A list of elements, which are references to tabpanel elements.
tabpanel	A container for the resources associated with a tab, where each tab is contained in a tablist.
textbox	Input that allows free-form text as its value. `<input type="text|password|email">` `<textarea>`
timer	A type of live region containing a numerical counter which indicates an amount of elapsed time from a start point, or the time remaining until an end point.
tooltip	A contextual popup that displays a description for an element.
tree	A type of list that may contain sub-level nested groups that can be collapsed and expanded.
treegrid	A grid whose rows can be expanded and collapsed in the same manner as for a tree.
treeitem	An option item of a element within a tree that may be expanded or collapsed if it contains a sub-level group of treeitem elements.

The roles above act as standalone user interface widgets or as part of larger, composite widgets.

Document Structure Roles

article	A section of a page that consists of a composition that forms an independent part of a document, page, or site. `<article>`

Document Structure Roles (cont)

column-header	A cell containing header inform-ation for a column.	`<th scope="col">`
definition	A definition of a term or concept.	
directory	A list of references to members of a group, such as a static table of contents.	
document	A region containing related inform-ation that is declared as document content, as opposed to a web appli-cation.	`<body>`
group	A set of user interface objects which are not intended to be included in a page summary or table of contents by assistive techno-logies.	`<fieldset>` `<optgroup>`
heading	A heading for a section of the page.	`<h1>`, `<h2>` `(aria-level must be set appropriately)`

Document Structure Roles (cont)

region	A large perceivable section of a web page or document, that is important enough to be included in a page summary or table of contents, for example, an area of the page containing live sporting event statistics.	`<div>` `<frame>` `<section>`
row	A row of cells in a grid.	`<tr>`
rowgroup	A group containing one or more row elements in a grid.	`<thead>` `<tbody>` `<tfoot>`
rowheader	A cell containing header information for a row in a grid.	`<th scope="row">`
separator	A divider that separates and distinguishes sections of content or groups of menuitems.	` `
toolbar	A collection of commonly used function buttons or controls represented in compact	`<menu type="toolbar">`

The roles above describe structures that organize content in a page. Document structures are not usually interactive.

Landmark Roles

applic-ation	A region declared as a web application, as opposed to a web **document**.	
banner	A region that contains mostly site-oriented content, rather than page-specific content.	`<header>` `<div id="header">`
comple-mentary	A supporting section of the document, designed to be complementary to the main content at a similar level in the DOM hierarchy, but remains meaningful when separated from the main content.	`<as ide>`
conten-tinfo	A large perceivable region that contains information about the parent document.	`<fo ote r>`

img	A container for a collection of elements that form an image.	`<img>`
list	A group of non-inter-active list items. See related l-istbox.	`<ul>, <ol>`
listitem	A single item in a list or directory.	`<li>`
math	Content that represents a mathematical expression.	
note	A section whose content is parenthetic or ancillary to the main content of the resource.	
presen-tation	An element whose implicit native role semantics will not be mapped to the accessibi-lity API.	`alt=""`

Landmark Roles (cont)

form	A **landmark** region that contains a collection of items and objects that, as a whole, combine to create a form. See related **search**.	`<fo rm>`
main	The main content of a document.	`<div id=" con ten - t">` `<ma in>`
navigation	A collection of navigational elements (usually links) for navigating the document or related documents.	`<na v>`
search	A **landmark** region that contains a collection of items and objects that, as a whole, combine to create a search facility. See related **form**.	

The roles above are regions of the page intended as navigational landmarks. All of these roles inherit from the **landmark** base type and, with the exception of **application**, all are imported from the *Role Attribute [ROLE]*. The roles are included here in order to make them clearly part of the WAI-ARIA Role taxonomy.

ARIA Global States and Properties

aria-a- tomic	Indicates whether assistive technologies will present all, or only parts of, the changed region based on the change notifications defined by the **aria-relevant** attribute. See related **aria-relevant**.
aria- busy	Indicates whether an element, and its subtree, are currently being updated.
aria-c- ontrols	Identifies the element (or elements) whose contents or presence are controlled by the current element. See related **aria-owns**.
aria-d- esc- ribedby	Identifies the element (or elements) that describes the object. See related **aria-labelledby**.
aria-d- isabled	Indicates that the element is perceivable but disabled, so it is not editable or otherwise operable. See related **aria-h- idden** and **aria-readonly**.

ARIA Global States and Properties (cont)

aria-d- rop- effect	Indicates what functions can be performed when the dragged object is released on the drop target. This allows assistive technologies to convey the possible drag options available to users, including whether a pop-up menu of choices is provided by the application. Typically, drop effect functions can only be provided once an object has been grabbed for a drag operation as the drop effect functions available are dependent on the object being dragged.
aria-f- lowto	Identifies the next element (or elements) in an alternate reading order of content which, at the user's discretion, allows assistive technology to override the general default of reading in document source order.
aria-g- rabbed	Indicates an element's "grabbed" state in a drag-and- drop operation.
aria-h- aspopup	Indicates that the element has a popup context menu or sub-level menu.
aria-h- idden	Indicates that the element and all of its descendants are not visible or perceivable to any user as implemented by the author. See related **aria-disabled**.
aria-i- nvalid	Indicates the entered value does not conform to the format expected by the application.
aria- label	Defines a string value that labels the current element. See related **aria-labelledby**.
aria-l- abe- lledby	Identifies the element (or elements) that labels the current element. See related **aria-label** and **aria-describedby**.
aria-live	Indicates that an element will be updated, and describes the types of updates the user agents, assistive techno- logies, and user can expect from the live region.

ARIA Global States and Properties (cont)

aria-owns	Identifies an element (or elements) in order to define a visual, functional, or contextual parent/child relationship between DOM elements where the DOM hierarchy cannot be used to represent the relationship. See related **aria-controls**.
aria-relevant	Indicates what user agent change notifications (additions, removals, etc.) assistive technologies will receive within a live region. See related **aria-atomic**.

Widget Attributes

aria-autocomplete	Indicates whether user input completion suggestions are provided.
aria-checked	Indicates the current "checked" state of checkboxes, radio buttons, and other widgets. See related **aria-pressed** and **aria-selected**.
aria-disabled	Indicates that the element is perceivable but disabled, so it is not editable or otherwise operable. See related **aria-hidden** and **aria-readonly**.
aria-expanded	Indicates whether the element, or another grouping element it controls, is currently expanded or collapsed.
aria-haspopup	Indicates that the element has a popup context menu or sub-level menu.
aria-hidden	Indicates that the element and all of its descendants are not visible or perceivable to any user as implemented by the author. See related **aria-disabled**.
aria-invalid	Indicates the entered value does not conform to the format expected by the application.
aria-label	Defines a string value that labels the current element. See related **aria-labelledby**.
aria-level	Defines the hierarchical level of an element within a structure.

Widget Attributes (cont)

aria-multiline	Indicates whether a text box accepts multiple lines of input or only a single line.
aria-multiselectable	Indicates that the user may select more than one item from the current selectable descendants.
aria-orientation	Indicates whether the element and orientation is horizontal or vertical.
aria-pressed	Indicates the current "pressed" state of toggle buttons. See related **aria-checked** and **aria-selected**.
aria-readonly	Indicates that the element is not editable, but is otherwise operable. See related **aria-disabled**.
aria-required	Indicates that user input is required on the element before a form may be submitted.
aria-selected	Indicates the current "selected" state of various widgets. See related **aria-checked** and **aria-pressed**.
aria-sort	Indicates if items in a table or grid are sorted in ascending or descending order.
aria-valuemax	Defines the maximum allowed value for a range widget.
aria-valuemin	Defines the minimum allowed value for a range widget.
aria-valuenow	Defines the current value for a range widget. See related **aria-valuetext**.
aria-valuetext	Defines the human readable text alternative of widget.

This section contains attributes specific to common user interface elements found on GUI systems or in rich internet applications which receive user input and process user actions. These attributes are used to support the widget roles.

Live Region Attributes

aria-atomic	Indicates whether assistive technologies will present all, or only parts of, the changed region based on the change notifications defined by the **aria-relevant** attribute. See related **aria-relevant**.

Live Region Attributes (cont)

aria-busy	Indicates whether an element, and its subtree, are currently being updated.
aria-live	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.
aria-relevant	Indicates what user agent change notifications (additions, removals, etc.) assistive technologies will receive within a live region. See related **aria-atomic**.

This section contains attributes specific to live regions in RIAs. These attributes may be applied to any element. The purpose of these attributes is to indicate that content changes may occur without the element having focus, and to provide assistive technologies with information on how to process those content updates. Some roles specify a default value for the **aria-live** attribute specific to that role. An example of a live region is a ticker section that lists updating stock quotes.

Drag-and-Drop Attributes

aria-drop-effect	Indicates what functions can be performed when the dragged object is released on the drop target. This allows assistive technologies to convey the possible drag options available to users, including whether a pop-up menu of choices is provided by the application. Typically, drop effect functions can only be provided once an object has been grabbed for a drag operation as the drop effect functions available are dependent on the object being dragged.
aria-grabbed	Indicates an element's "grabbed" state in a drag-and-drop operation.

This section lists attributes which indicate information about drag-and-drop interface elements, such as draggable elements and their drop targets. Drop target information will be rendered visually by the author and provided to assistive technologies through an alternate modality.

Relationship Attributes

aria-activedescendant	Identifies the currently active descendant of a widget.

Relationship Attributes (cont)

aria-controls	Identifies the element (or elements) whose contents or presence are controlled by the current element. See related **aria-owns**.
aria-describedby	Identifies the element (or elements) that describes the object. See related **aria-labelledby**.
aria-flowto	Identifies the next element (or elements) in an alternate reading order of content which, at the user's discretion, allows assistive technology to override the general default of reading in document source order.
aria-labelledby	Identifies the element (or elements) that labels the current element. See related **aria-label** and **aria-describedby**.
aria-owns	Identifies an element (or elements) in order to define a visual, functional, or contextual parent/child relationship between DOM elements where the DOM hierarchy cannot be used to represent the relationship. See related **aria-controls**.
aria-posinset	Defines an element's number or position in the current set of listitems or treeitems. Not required if all elements in the set are present in the DOM. See related **aria-setsize**.
aria-setsize	Defines the number of items in the current set of listitems or treeitems. Not required if all elements in the set are present in the DOM. See related **aria-posinset**.

This section lists attributes that indicate relationships or associations between elements which cannot be readily determined from the document structure.