

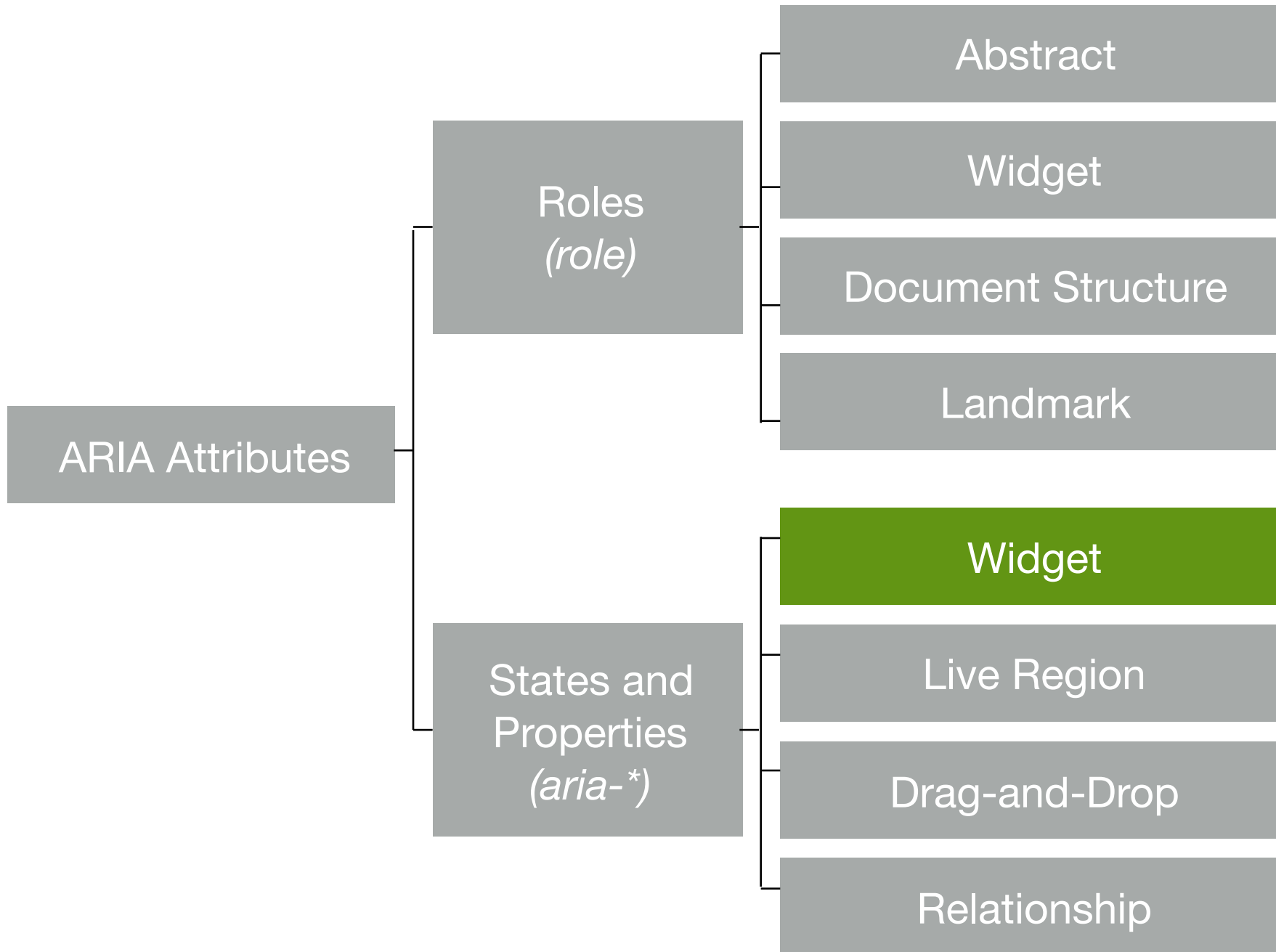
ARIA

STATES & PROPERTIES

We are now going to look in more detail at the **different state and property attributes.**

As with **role** attributes, it's not possible to cover all of these attributes today, so I'm going to **highlight some attributes that are important to know about.**

Widget attributes



Widget attributes are specific to common user interface elements **which receive user input and process user actions.**

- aria-autocomplete
- aria-checked (state)
- aria-disabled (state)
- aria-errormessage (1.1)
- aria-expanded (state)
- aria-haspopup
- aria-hidden (state)
- aria-invalid (state)
- aria-label
- aria-level
- aria-modal (1.1)
- aria-multiline
- aria-multiselectable
- aria-orientation
- aria-placeholder (1.1)
- aria-pressed (state)
- aria-readonly
- aria-required
- aria-selected (state)
- aria-sort
- aria-valuemax
- aria-valuemin
- aria-valuenow
- aria-valuetext

aria-errormessage

The `aria-errormessage` attribute identifies an element that **provides error messages** for form controls.

For example, an `<input>` may display **an inline error message** when users enter data incorrectly.

The inline error message can be **programmatically associated with the input** using a matching **aria-errormessage** and **ID** values.

```
<label for="email">Email</label>
```

```
<input id="email" aria-errormessage="msg01"  
type="text">
```

```
<span id="msg01">Error message</span>
```

Authors **must use** the `aria-invalid` attribute in conjunction with the `aria-errormessage` attribute.

Initially, **the object is in a valid state** and either has `aria-invalid` set to `false` or no `aria-invalid` attribute.

```
<!-- Initial valid state -->
```

```
<label for="email">Email</label>
```

```
<input id="email" aria-errormessage="msg01"  
type="text" aria-invalid="false">
```

```
<span id="msg01"></span>
```

If the user enters an invalid value for the form control, `aria-invalid` must be set to `true` to indicate that `aria-errormessage` is **now relevant**.


```
<!-- User has input an invalid value -->
```

```
<label for="email">Email</label>
```

```
<input id="email" aria-errormessage="msg01"  
type="text" aria-invalid="true">
```

```
<span id="msg01">
```

```
    Error message
```

```
</span>
```

Authors can also use **live regions** for the error message element such as `aria-live` or `alert`.

```
<!-- Initial valid state -->
```

```
<label for="email">Email</label>
```

```
<input id="email" aria-errormessage="msg01"  
type="text" aria-invalid="false">
```

```
<span id="msg01" aria-live="off"></span>
```

```
<!-- User has input an invalid value -->
```

```
<label for="email">Email</label>
```

```
<input id="email" aria-errormessage="msg01"  
type="text" aria-invalid="true">
```

```
<span id="msg01" aria-live="assertive">
```

```
    Error message
```

```
</span>
```

Finally, the error message content must not be available to users **unless there is an invalid value.**

So, the error message content should be initially hidden or not present, **then made available** only when an invalid value is used.

```
<!-- Initial valid state -->
```

```
<label for="email">Email</label>
```

```
<input id="email" aria-errormessage="msg01"  
type="text" aria-invalid="false">
```

```
<span id="msg01" aria-live="off"></span>
```

```
<!-- User has input an invalid value -->
```

```
<label for="email">Email</label>
```

```
<input id="email" aria-errormessage="msg01"  
type="text" aria-invalid="true">
```

```
<span id="msg01" aria-live="assertive">  
    Error message  
</span>
```


Exercise 5:

Disabled, Invalid, Hidden

Open **exercise05/start.html** in a browser and also in a text editor.

To make a disabled form control be **announced as “disabled” or “dimmed”** we can add the **aria-disabled** attribute.

```
<input  
  class="input input__disabled"  
  id="email"  
  type="text"  
  disabled  
  aria-disabled="true"  
>
```

Keep in mind that the boolean `disabled` attribute **defines the input as disabled**, the `aria-disabled` attribute is only used to **tell screen reader users of it's state**.

To make an invalid form control be **announced as “invalid”** we can add the `aria-invalid` attribute.

```
<input  
  class="input input__invalid"  
  id="address"  
  type="text"  
  invalid  
  aria-invalid="true"  
>
```

Keep in mind that the boolean `invalid` attribute **defines the input as invalid**, the `aria-invalid` attribute is only used to **tell screen reader users of it's state**.

To **hide an element and all of its child elements** from screen readers, you can apply the `aria-hidden` attribute.

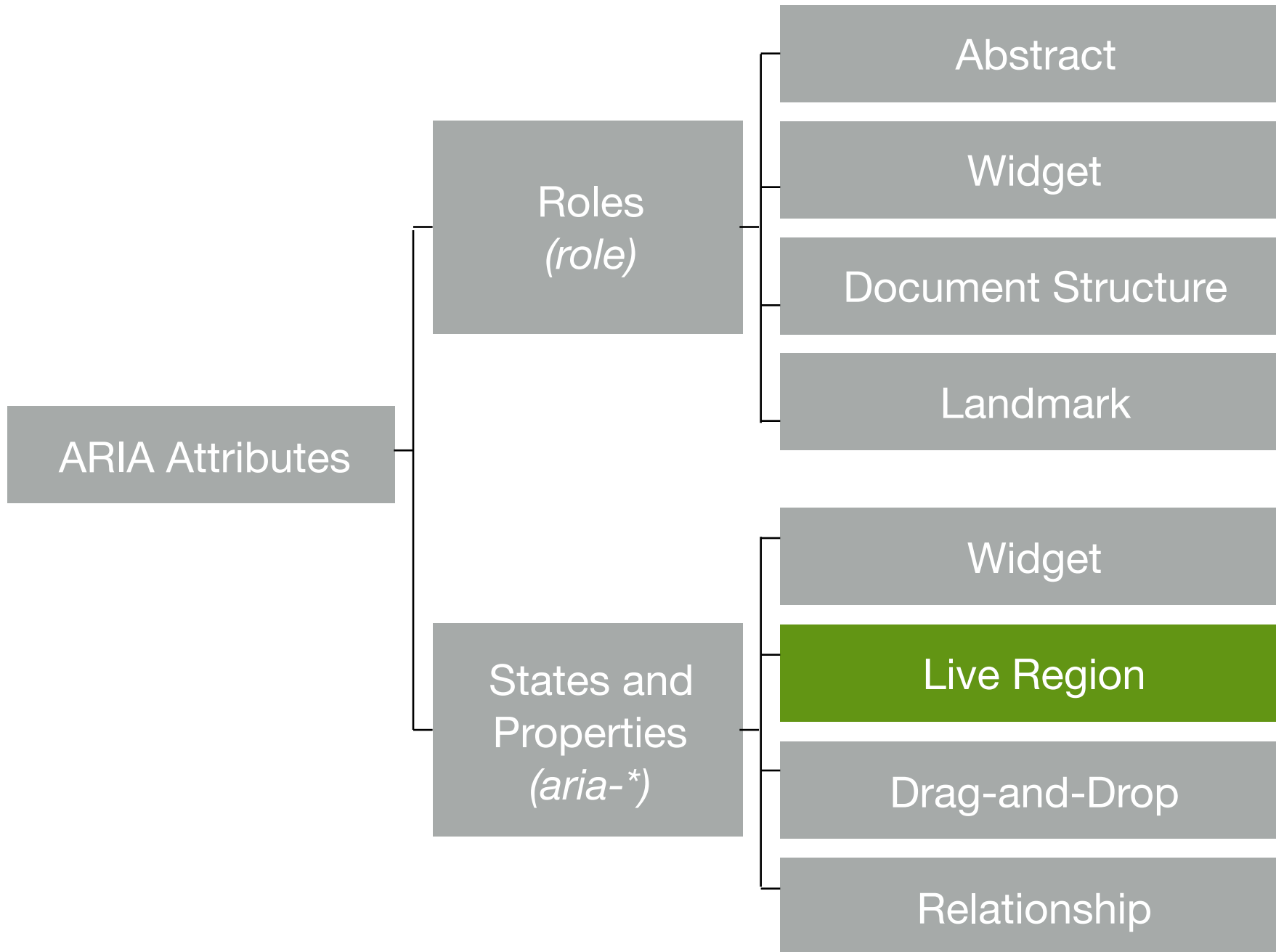
```
<p aria-hidden="true">This content is hidden from  
Screen Readers.</p>
```

```
<p hidden>This content is hidden.</p>
```

```
<p aria-hidden="false">This content is not hidden  
from Screen Readers.</p>
```

Keep in mind that the boolean `hidden` attribute **means that the element is not displayed to anyone**, the `aria-hidden` attribute is only used to **hide the element from screen readers**.

Live region attributes



These attributes indicate to Assistive Technologies **that content changes may occur** without the element having focus, and to provide assistive technologies with information on how to process those content updates.

- `aria-atomic`
- `aria-busy (state)`
- `aria-live`
- `aria-relevant`

aria-live

We can apply the `aria-live` attribute
to **any HTML element**.

```
<div aria-live="polite">
```

```
...
```

```
</div>
```

If anything dynamically changes inside that element after the initial page load, screen readers will be **made aware of these changes**.

The aria-live attribute can be used for any page regions that are **likely to get updates** after the initial page is loaded.

Info alerts! Some info to be aware of


















Success alerts! Your changes are saved

Warning alerts! Something has changed

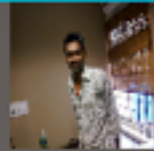
Error alerts! Fix the error and try again

Alert messages

Dynamic stock info

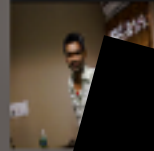
World Market Watch			
Exchange	Last	Prev close	Change
 DJIA®	16958.61	16906.62	41.78
 SP500 ETF	196.44		0.20
 TSX			43.09
			0.61
	25.5	25.49	0.01
 CAC 40	4359.84	4342.53	6.31
 DAX	9808.2	9772.67	19.15
 AEX	410.49	409.76	-0.05
 WIG20	2343.39	2360.91	-21.53
 IPC	43534.07	43403.43	139.66
 Bovespa	53634.69	53801.83	-179.17
 Nikkei	15302.65	15314.41	108.33
 Hang Seng	23176.07	23541.38	-231.98
 Shanghai	2038.61	2064.02	-23.02
 NZX 50	5122.74	5166.08	-43.34
 TSEC	9510.05	9526.23	-15.88
 Sensex	25444.81	25582.11	-170.47
 Nifty	7585	7623.2	-52.95
 Russel ETF	116.16	116.34	-0.18
Powered by: HowTheMarketWorks			
		<input type="button" value="GET QUOTE.."/>	<input type="button" value="Go"/>

mR aNdRoID MaD TuToRiAL

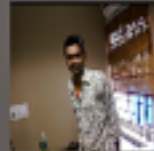


Android RSS Feed Reader Example

Fri, 08 Jun 2012 13:55:00 +0000

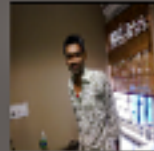


Android Turn ON wifi
programmatically



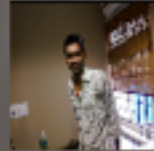
Android Widge

Tue, 29 May 2012 06:13:00 +0000



Pull to Refresh ListView in
Android.

Mon, 21 May 2012 07:25:00 +0000



Always show zoom controls on a
MapView in Android

Wed, 14 Mar 2012 06:38:00 +0000

Change Sorting Mode

Dynamic RSS feeds





































Date:

January 2011

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Date pickers

The Person List

		Name	City	Birth date	Status	Record date		
		Andy More	Trabzon	1971-03-10	Passive	2013-01-16		
		Andy Puggy	Volos	1973-06-25	Active	2013-01-16		
		Andy Yates	Adana	1948-10-05	Active	2013-01-16		
				1957-08-10	Active	2013-01-16		
		Arda Maalouf		1972-02-13	Active	2013-01-16		
		Arda More	London	1972-02-15	Active	2013-01-16		
		Audrey Brown	Berlin	1975-05-21	Active	2013-01-16		
		Audrey Brown	Istanbul	1960-11-21	Active	2013-01-16		
		Audrey Kalkan	London	1994-06-14	Active	2013-01-16		
		Austin Anar	Paris	1969-11-27	Active	2013-01-16		

Sortable tables

|< < 1 2 3 ... 11 12 13 > >| Showing 11 to 20 of 128 records + Add new record

There are **three possible values** for
aria-live:

```
aria-live="off"
```

Assistive technologies **should not announce updates** unless the assistive technology is currently focused on that region.

The **off** value can be used for information that is **not critical** for users to know about immediately.

```
<div aria-live="off">
```

```
...
```

```
</div>
```

`aria-live="polite"`

Assistive technologies should announce updates at the **next graceful opportunity** (eg end of current sentence).

The `polite` value can be used for **warning notifications** that users may need to know.

```
<div aria-live="polite">
```

```
...
```

```
</div>
```


`aria-live="assertive"`

Assistive technologies should
announce updates immediately.

The **assertive** value should only be used if the interruption is **imperative for users to know immediately** such as error alerts.

```
<div aria-live="assertive">
```

```
...
```

```
</div>
```

aria-atomic

The `aria-atomic` attribute 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.

```
<!-- The true attribute will present the region as a  
whole when changes are detected. -->
```

```
<div aria-live="off" aria-atomic="true">  
</div>
```

```
<!-- The false attribute will present only the  
changed regions. This is the default. -->
```

```
<div aria-live="off" aria-atomic="false">  
</div>
```

aria-relevant

The **aria-relevant** attribute describes **semantically meaningful changes**, as opposed to merely presentational ones.

additions are **insertion of nodes**
within the live region. These should be
considered relevant and announced to
screen readers.

```
<!-- additions attribute -->
```

```
<div aria-live="off" aria-relevant="additions">  
</div>
```

removals are **deletion of nodes** within the live region. These should be considered relevant and announced to screen readers.

```
<!-- removals attribute -->
```

```
<div aria-live="off" aria-relevant="removals">  
</div>
```

text are **changes to the textual content of existing nodes** within the live region. These should be considered relevant and announced to screen readers.

```
<!-- text attribute -->
```

```
<div aria-live="off" aria-relevant="text">  
</div>
```

all are **insertion of deletion of nodes** within the live region. These should be considered relevant and announced to screen readers.


```
<!-- all attribute -->
```

```
<div aria-live="off" aria-relevant="all">  
</div>
```

Multiple, **comma-separated values**
can be used with the **aria-relevant**
attribute.

```
<!-- multiple attributes -->
```

```
<div aria-live="off" aria-relevant="text, removals">  
</div>
```

The **removals** and **all** values **should be used sparingly**. Assistive technologies only need to be informed of important change.

aria-busy

The `aria-busy` state indicates whether an element, and its subtree, are **currently being updated**.

If multiple parts of the same element need to be loaded or modified, the live region can be set to **true while the loading is in progress.**

```
<!-- true attribute -->
```

```
<div aria-live="off" aria-busy="true">  
</div>
```


When **the last part has been loaded**,
the live region can be set to **false**.

```
<!-- false attribute -->
```

```
<div aria-live="off" aria-busy="false">  
</div>
```

role=alert

The `role="alert"` attribute is used to **define a message with important,** and usually time-sensitive, information.

```
<!-- role=alert attribute -->
```

```
<div role="alert">
```

```
</div>
```

Elements with the `role="alert"` have an implicit `aria-live` value of **“assertive”**, and an implicit `aria-atomic` value of **“true”**.

For this reason there is **no need to use** the `aria-live` attribute as well as `role="alert"` on the same element.

```
<!-- Not needed -->
```

```
<div aria-live="assertive" role="alert">  
</div>
```


role=alertdialog

The `role="alertdialog"` attribute defines a type of **dialog that contains an alert message**, where initial focus goes to an element within the dialog.

The `role="alertdialog"` attribute goes on the **node containing both the alert message and the rest of the dialog.**

Unlike `role="alert"`, the `role="alertdialog"` attribute **can receive a response** from the user - such as a “Confirm” or “Save” button.

```
<!-- role=alertdialog attribute -->
```

```
<div role="alertdialog">
```

```
</div>
```

Exercise 6:

Live regions

Open **exercise06/start/** files in a browser and also in a text editor.

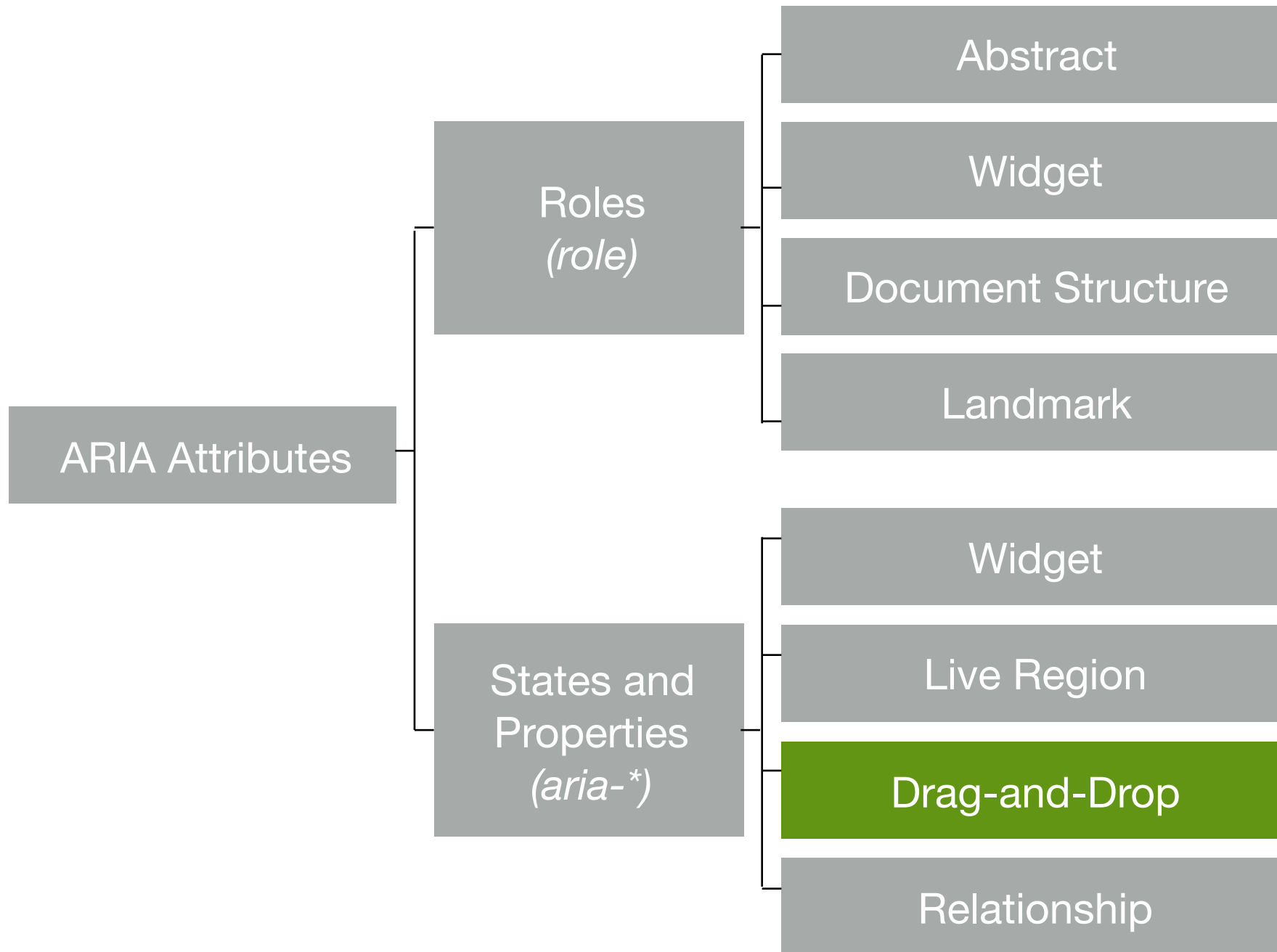
```
<div id="message" class="error" aria-  
live="assertive"></div>
```



```
<div id="message" class="error" aria-live="off"></div>
```

```
<div id="message" class="error" aria-live="polite"></div>
```

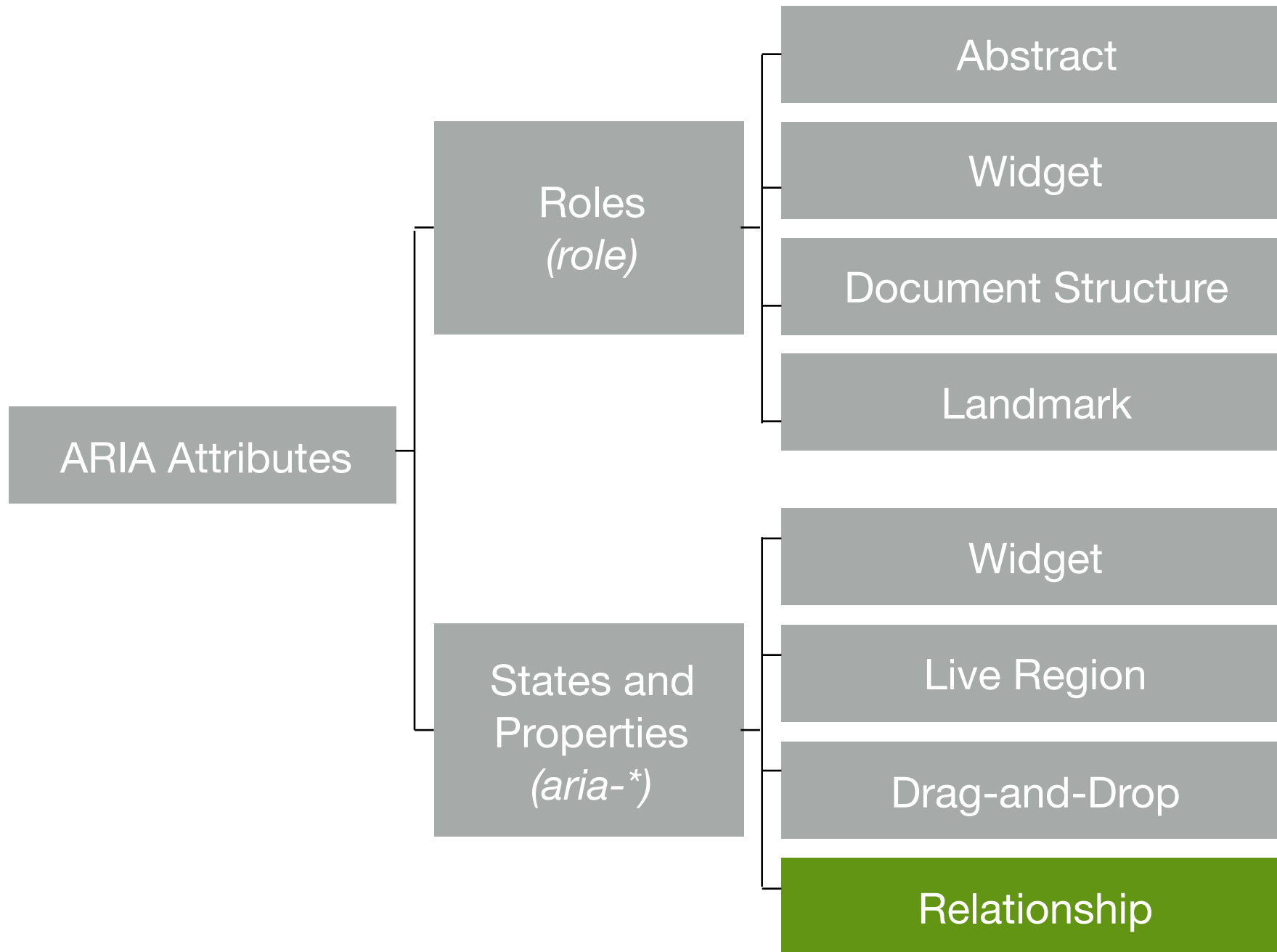
Drag-and-drop
attributes



These attributes indicate **information about drag-and-drop interface elements**, such as draggable elements and their drop targets.

- `aria-dropeffect`
- `aria-grabbed (state)`

Relationship
attributes



Relationship attributes **indicate relationships or associations between elements** which cannot be readily determined from the document structure.

- aria-activedescendant
- aria-colcount (1.1)
- aria-colindex (1.1)
- aria-colspan (1.1)
- aria-controls
- aria-describedby
- aria-details (1.1)
- aria-errormessage (1.1)
- aria-flowto
- aria-labelledby
- aria-owns
- aria-posinset
- aria-rowcount (1.1)
- aria-rowindex (1.1)
- aria-rowspan (1.1)
- aria-setsize

Difference between label,
labelledby and
describedby

The `aria-labelledby` attribute is used to **establish a relationship between the widget and its label.**

The `aria-labelledby` provides a label that describes **the essence of an object**.

```
<!-- Widget -->
```

```
<input aria-labelledby="instructions">
```

```
<!-- Label -->
```

```
<p id="instructions">
```

```
  Label content
```

```
</p>
```

The `aria-describedby` attribute is used to establish **a relationship between the widget and its description.**

The `aria-describedby`: provides a description with **more information that the user might need.**


```
<!-- Widget -->
```

```
<input aria-describedby="instructions">
```

```
<!-- Description -->
```

```
<div id="instructions">
```

```
  Description
```

```
</div>
```

The `aria-label` attribute is used to provide a **label for the current element**.

The `aria-label` attribute is **only announced to screen readers**, it is not visible on screen.

The `aria-label` attribute **should not be used on an element which is non-interactive** unless it has an `aria-`
`role`.

```
<button type="button" aria-label="Close and return to  
account details">
```

aria-owns

The `aria-owns` attribute allows us **to define a parent/child contextual relationship** for assistive technologies that is otherwise impossible to infer from the DOM.

In other words, **we can define the**
<input> element as the parent, and
the **** element as the child element.


```
<input aria-owns="results">
```

```
<ul id="results">
```

```
...
```

```
</ul>
```

Exercise 7:

Menubar

Open **exercise07/start.html** in a browser and also in a text editor.

```
<nav role="navigation" aria-label="Main menu">
```

```
<ul id="menubar1" role="menubar">
```

```
<a role="menuitem" aria-haspopup="true" aria-expanded="false"  
href="#">About</a>  
<a role="menuitem" aria-haspopup="true" aria-expanded="false"  
href="#">Admissions</a>  
<a role="menuitem" aria-haspopup="true" aria-expanded="false"  
href="#">Academics</a>
```

```
<ul role="menu" aria-label="About">  
<ul role="menu" aria-label="Admissions">  
<ul role="menu" aria-label="Academics">
```

```
<li role="none">
```

```
<li role="none">
```

```
<li role="none">
```

```
<li role="none">
```

```
<li role="none">
```

```
<li role="none">
```

```
<li role="none">
```

```
<li role="none">
```

```
<li role="none">
```



```
<a role="menuitem" href="#">Overview</a>
```

Exercise 8:

Tab panel

Open **exercise08/start.html** in a browser and also in a text editor.

```
<ul role="tablist">  
  <li role="presentation"> ... </li>  
  <li role="presentation"> ... </li>  
  <li role="presentation"> ... </li>  
</ul>
```

```
<ul role="tablist">
  <li role="presentation">
    <a href="#panel1" id="tab1" role="tab">Apple</a>
  </li>
  <li role="presentation">
    <a href="#panel2" id="tab2" role="tab">Pears</a>
  </li>
  <li role="presentation">
    <a href="#panel3" id="tab3" role="tab">Oranges</a>
  </li>
</ul>
```

```
<ul role="tablist">
  <li role="presentation">
    <a href="#panel1" id="tab1" role="tab"
      aria-controls="panel1">Apple</a>
  </li>
  <li role="presentation">
    <a href="#panel2" id="tab2" role="tab"
      aria-controls="panel2">Pears</a>
  </li>
  <li role="presentation">
    <a href="#panel3" id="tab3" role="tab"
      aria-controls="panel3">Oranges</a>
  </li>
</ul>
```

```
<ul role="tablist">
  <li role="presentation">
    <a href="#panel1" id="tab1" role="tab"
      aria-controls="panel1" aria-selected="true">Apple</a>
  </li>
  <li role="presentation">
    <a href="#panel2" id="tab2" role="tab"
      aria-controls="panel2" aria-selected="false">Pears</a>
  </li>
  <li role="presentation">
    <a href="#panel3" id="tab3" role="tab"
      aria-controls="panel3" aria-selected="false">Oranges</a>
  </li>
</ul>
```

```
<div id="panel1" role="tabpanel">  
  Panel 1  
</div>  
<div id="panel1" role="tabpanel">  
  Panel 2  
</div>  
<div id="panel1" role="tabpanel">  
  Panel 3  
</div>
```



```
<div id="panel1" role="tabpanel" aria-labelledby="tab1">  
  Panel 1  
</div>  
<div id="panel1" role="tabpanel" aria-labelledby="tab2">  
  Panel 2  
</div>  
<div id="panel1" role="tabpanel" aria-labelledby="tab3">  
  Panel 3  
</div>
```