

ARIA Implementation: Moving Beyond the Basics



WebAIM.org

ARIA



ARIA expands the
vocabulary of HTML to
support what screen
readers already understand

```
<slider>???
```


ARIA Paves the Cow Paths



ARIA Paves the Cow Paths



ARIA Paves the Cow Paths



ARIA Paves the Cow Paths



ARIA Paves the Cow Paths



ARIA Paves the Cow Paths



You can only make things
more accessible by
implementing ARIA now

... if you do it correctly

ARIA



ALL THE THINGS

Rule #1 of ARIA

Don't use ARIA

... unless you must

Always use native, accessible
HTML first

... then use ARIA to enhance or
fill in gaps.

Rule #2 of ARIA

Do not change native semantics, unless you really have to.

```
<div role=navigation>
```

```
  <ul>...
```

instead of

```
<ul role=navigation>...
```

Rule #3 of ARIA

All interactive ARIA controls must be usable with the keyboard.

Rule #4 of ARIA

Do not use `role="presentation"` or `aria-hidden="true"` on visible, focusable elements.

Rule #5 of ARIA

All interactive elements must have
an accessible name.

ARIA does not change
functionality, it only changes the
presented roles/properties to
screen reader users

ARIA Design Patterns

ARIA slider design pattern:

Description:	A slider is user input where the user selects a value from within a given range. Sliders typically have a button such that when moved will change the current value within the current range of the slider. The button must be keyboard accessible. It is typically possible to add or subtract to the current value by using directional keys such as arrow keys.
Keyboard Interaction:	<ul style="list-style-type: none">• Right Arrow and Up Arrow increase the value of the slider.• Left Arrow and Down Arrow decrease the value of the slider.• Home and End move to the minimum and maximum values of the slider.• Tab into and out of the slider.• Page Up and Page Down optionally increment or decrement the slider by a given amount. <p>Focus is placed on the slider. (The visual object that the mouse user would move, also known as the thumb.)</p> <p>Localization for right to left languages may wish to reverse the left and right arrows.</p>
WAI-ARIA Roles, States, and Properties:	<ul style="list-style-type: none">• The slider control has the role slider.• Sliders support the aria-valuemin, aria-valuemax, and aria-valuenow properties representing the minimum possible value of the slider, the maximum possible value, and the current value. All of these are decimal numbers. The minimum and maximum are typically fixed and do not change.• Sometimes the value is not user readable, such as a number for the day of the week, e.g., "1". In those cases, use the aria-valuetext property to provide a human readable string for the slider's value, e.g. "Monday".• It is recommended that authors provide a visible label for the slider, referencing it using aria-labelledby.
Example:	<ul style="list-style-type: none">• Open Ajax Alliance Sliders• MINDTROVE Rating Widget• Paciello Group slider <p>Please note that not all examples work in all browser and version combinations. For example, note the compatibility statement.</p>

Device Independence

Use device independent event handlers

or

combine mouse (e.g, `onmouseover`) and
keyboard (e.g, `onkeypress`) dependent event
handlers

Use Links/Buttons Appropriately

- Links open a new page or jump to another location within the existing page
- Buttons submit form data or perform an in-page function

Ensure Interactive Elements are Links or Form Controls

or...

make non-focusable elements focusable with
tabindex

Avoid Tabindex

... unless you're sure you know what you're doing.

If the default tab order is not logical,
fix your source code order.

Avoid this
`tabindex="1+"` defines an explicit tab order

`tabindex="0"` allows things besides links and form elements to receive keyboard focus.

`tabindex="-1"` allows things besides links and form elements to receive programmatic focus (by scripting, links, etc.)


```
<div onclick="submitForm()">  
  Submit Search</div>
```

```
<div tabindex="0"  
onclick="submitForm()">  
Submit Search</div>
```

WARNING!

Click events do not always trigger via keyboard
for things other than links or form controls...

... even with `tabindex="0"`

Check for Enter (13) and Space (32) key presses

```
if (event.keyCode==13 || event.keyCode==32)
{
    doStuff();
}
```

```
function a1lyClick(event) {
    if(event.type === 'click') {
        return true;
    }
    else if(event.type === 'keypress') {
        var code = event.charCode || event.keyCode;
        if(code === 32) { // don't scroll the page
            event.preventDefault();
        }
        if((code === 32) || (code === 13)) {
            return true;
        }
    }
    else {
        return false;
    }
}
}
*****
$('#fake-button').on('click keypress', function(event) {
    if(a1lyClick(event) === true) {
        // do magic javascript stuff
    }
});
```

Just Use *A* Button

```
<div tabindex="0"  
onclick="submitForm()">  
Submit Search</div>
```



```
<a href="#"  
onclick="submitForm()">  
Submit Search</a>
```

```
<a href="#" role="button"  
  onclick="submitForm()">  
  Submit Search</a>
```

Just Use A Button!!!

```
<button onclick="submitForm()">  
  Submit Search</button>
```

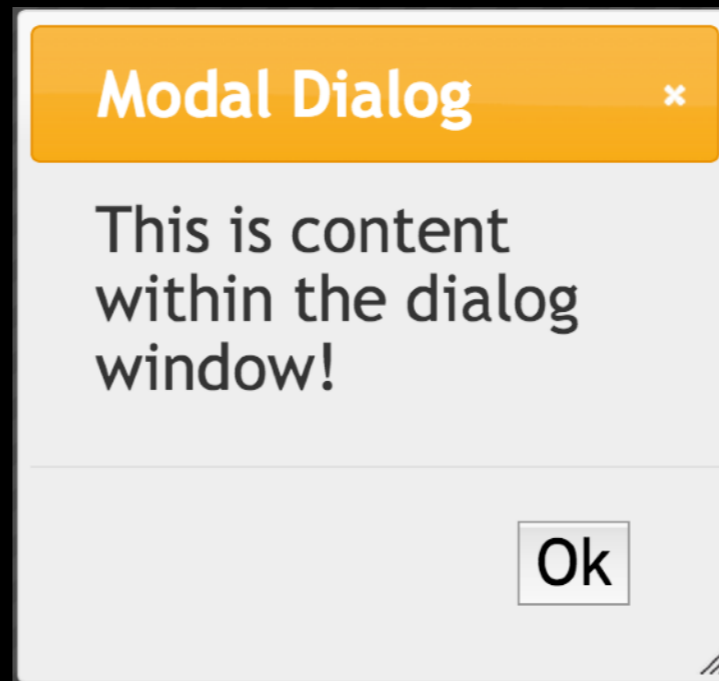
```
<div role="button" tabindex="0">  
  Submit Search</div>
```

If you don't
Just Use A Button
you must...

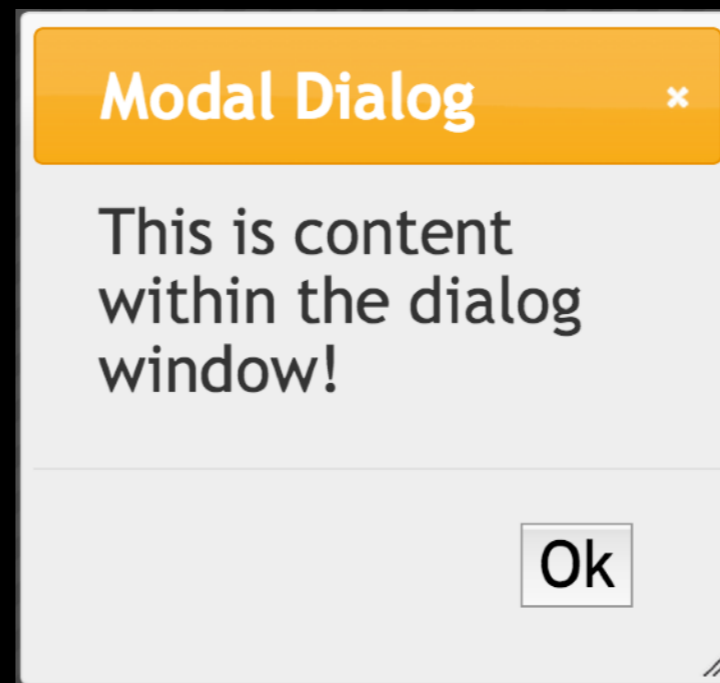
- ... ensure the proper semantics
- ... make it respond to mouse click, Enter, and Space
 - ... provide focus styling

`tabindex="-1"`

- Allows non-focusable elements to receive programmatic focus (by scripting, links, etc.)
- Necessary for focusing dialog boxes, error messages, etc.
- **WARNING:** This removes the element from the default tab order.

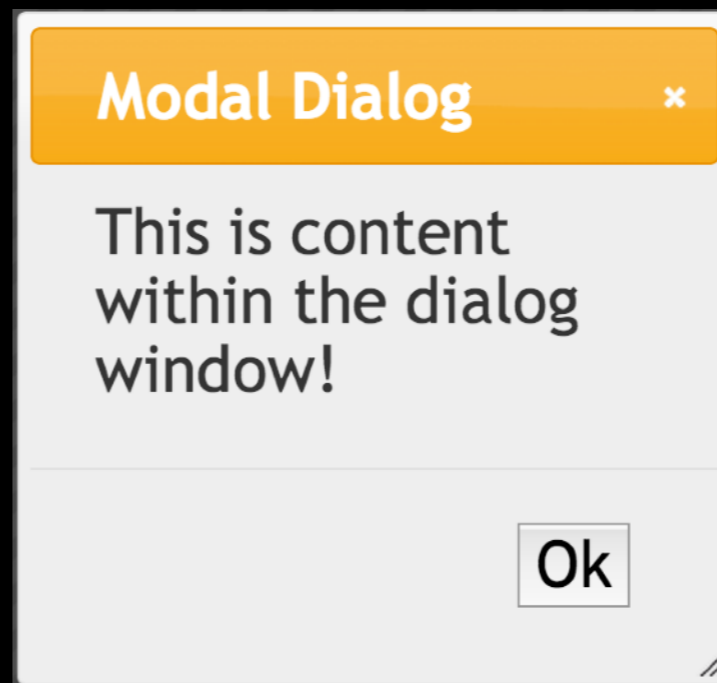


Link or button (with
`aria-haspopup="dialog"`)



`tabindex="-1"` then set focus to dialog (or focus a control or element inside the dialog)

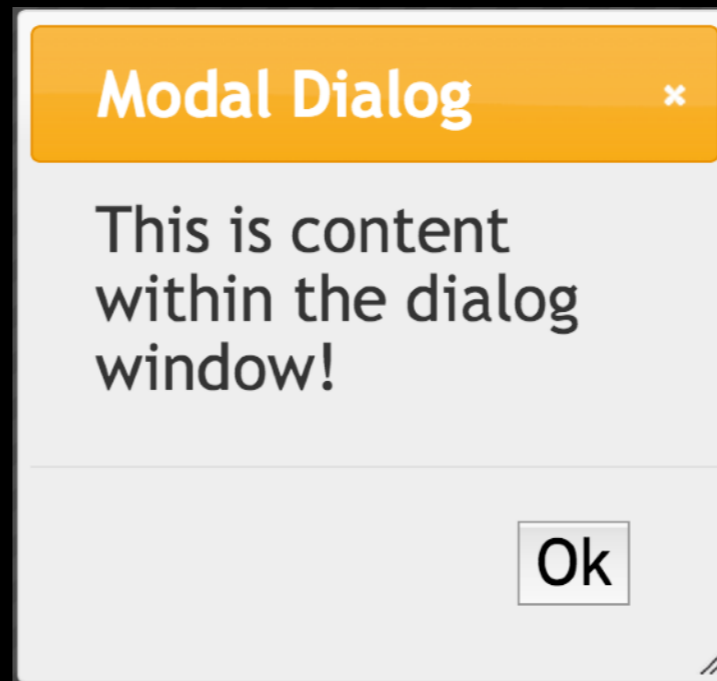
Link or button (with `aria-haspopup="dialog"`)




```
role="dialog"  
w/ aria-labelledby
```

tabindex="-1" then set focus
to dialog (or focus a control or
element inside the dialog)

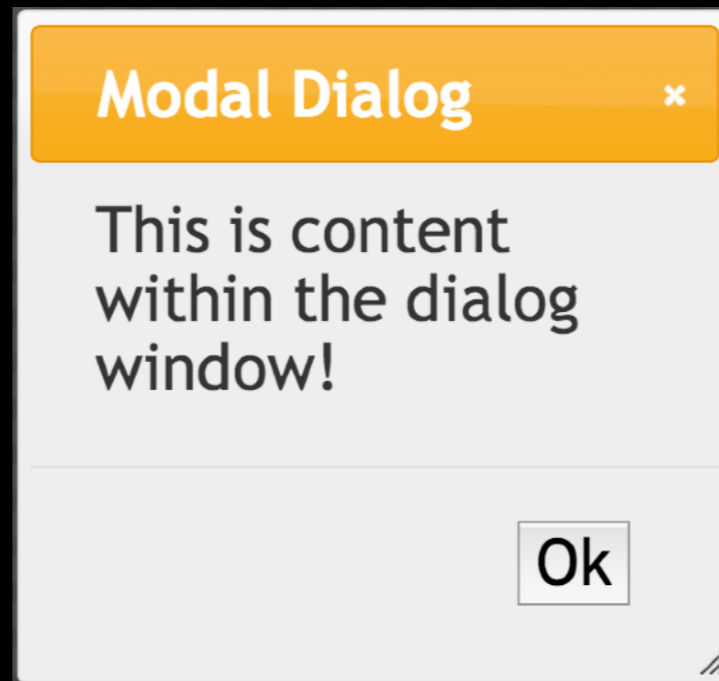
Link or button (with
aria-haspopup="dialog")



```
role="dialog"  
w/ aria-labelledby
```

tabindex="-1" then set focus
to dialog (or focus a control or
element inside the dialog)

Maintains keyboard
focus if modal



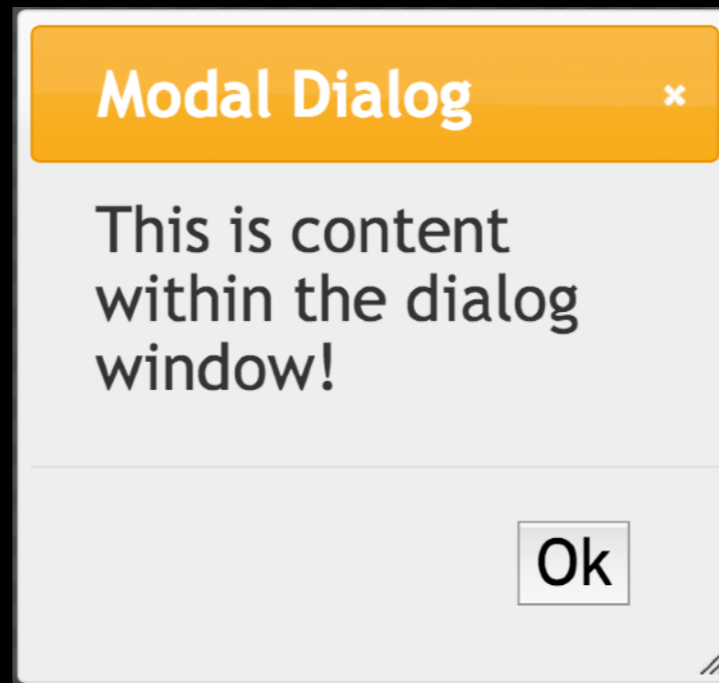
Link or button (with
aria-haspopup="dialog")



`role="dialog"`
w/ `aria-labelledby`

`tabindex="-1"` then set focus
to dialog (or focus a control or
element inside the dialog)

Maintains keyboard
focus if modal



Link or button (with
`aria-haspopup="dialog"`)

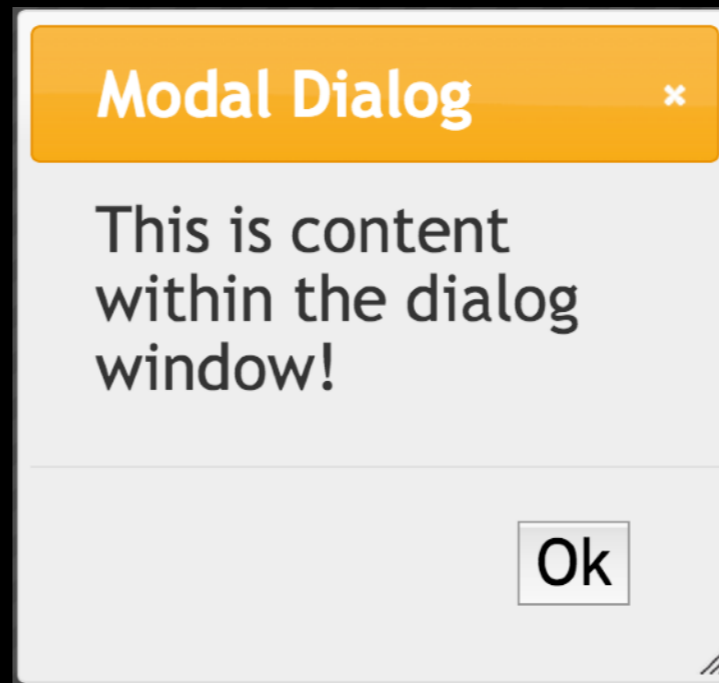


Closes with
ESC key

`role="dialog"`
w/ `aria-labelledby`

`tabindex="-1"` then set focus
to dialog (or focus a control or
element inside the dialog)

Maintains keyboard
focus if modal



Link or button (with
`aria-haspopup="dialog"`)



Returns
focus when
dismissed

Closes with
ESC key

Trapping keyboard focus

- Typically a very bad idea (except for dialogs, menus, etc.)
- Foreground content
 - Manage focus (account for Shift + Tab) and `aria-modal="true"`, or (soon?) HTML5 `modal`
- Background content
 - `aria-hidden="true"` (dialog must be sibling to the remaining content), `tabindex="-1"`, `inert` polyfill, `hidden`, etc.

“Freak-out” Mode

When the currently focused element disappears
or is significantly modified



Avoid it or address it with
`focus () ;`

Non-modal Dialogs

"ARIA" Pronunciation ▾

Area (like "area")

Arrrr-ee-uh (like a pirate)

Ehr-EE-uh (like "diarrhea")

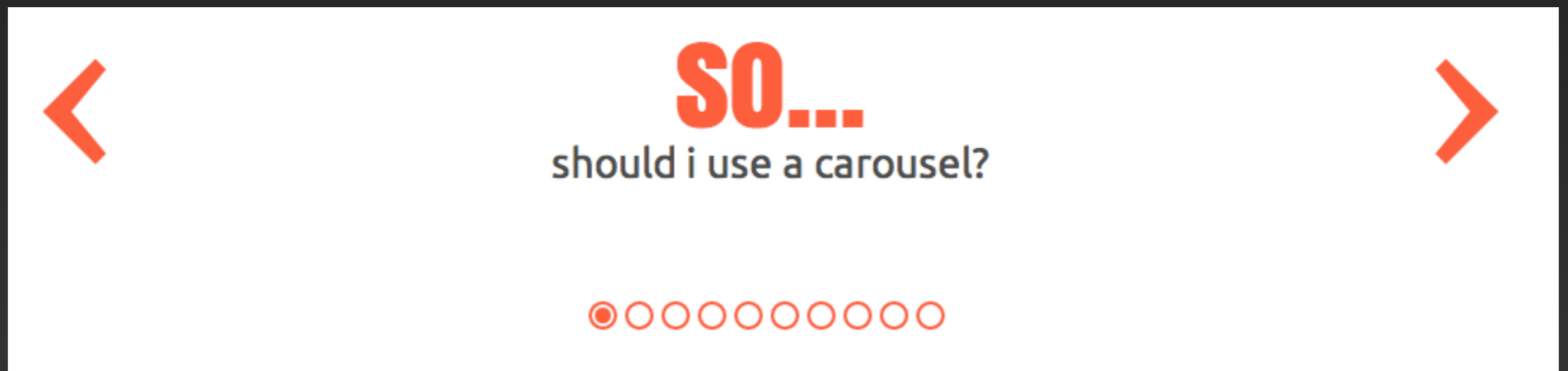
Date:

May 2017						
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			

- Follow the ARIA design patterns
- When possible, use HTML native input types
- `<input type="date">`

Carousels

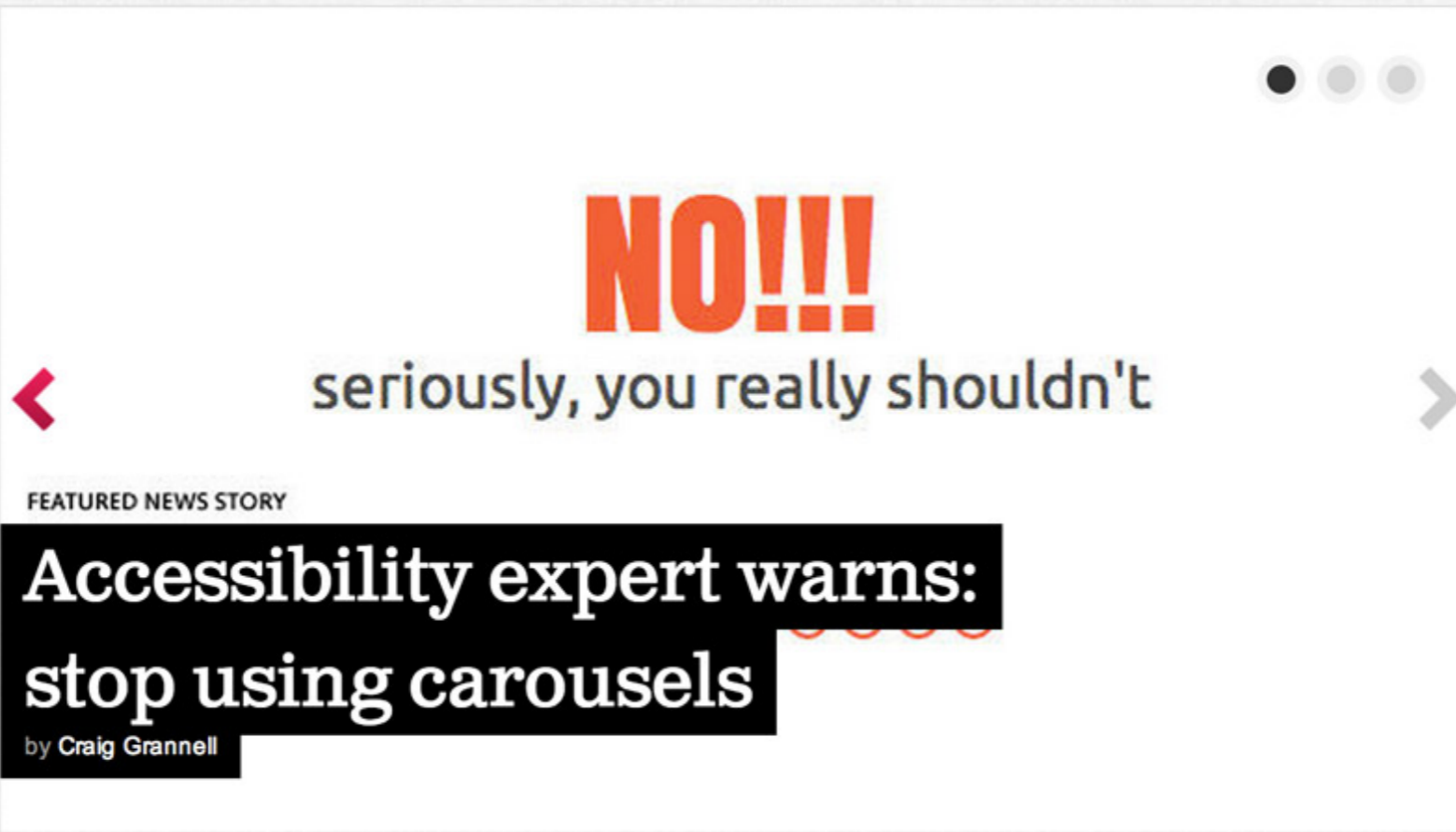
Carousels



<http://shouldiuseacarousel.com/>

Q&A by .net magazine

<http://www.netmagazine.com/news/accessibility-expert-warns-stop-using-carousels-132875>



NO!!!

seriously, you really shouldn't

FEATURED NEWS STORY

**Accessibility expert warns:
stop using carousels**

by Craig Grannell

This is a carousel containing a single slide. The slide features a large red 'NO!!!' and the text 'seriously, you really shouldn't'. Below this is a 'FEATURED NEWS STORY' section with a headline 'Accessibility expert warns: stop using carousels' and a byline 'by Craig Grannell'. The carousel has navigation arrows and a progress indicator.



65 PAGES OF PRACTIC
develop discover c
ne
> BU
PER
MOB
A
YouTube | RV
for brands | fron

This is a sidebar navigation menu with various links and icons.

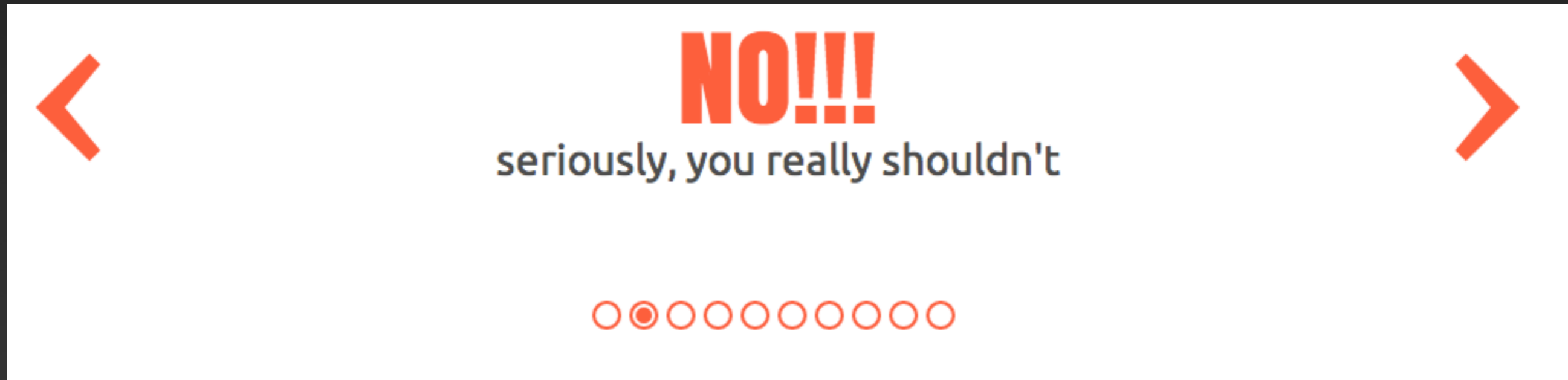
NEWSFLASH

Accessibility expert Open Device Lab reaches 1,000 bits of kit

THE C

An anti-carousel carousel featured in a carousel

Carousels



<http://shouldiuseacarousel.com/>

Carousel Issues

- Distracting and confusing
- Automated carousels violate WCAG 2.0 Success Criteria 2.2.2 (Level A) - Pause, Stop, Hide
- “Freak-out mode” when carousel changes
- Difficult interaction model
 - No relationship between controls and content
- Allow poor content decisions

Carousel Accessibility Solutions?

- Avoid auto-playing (optimal) or include a visible pause button (preferably) before the carousel
- Pause carousel on mouse hover and on keyboard focus
- Ensure focused items do not disappear, or manage focus when they do
- Provide context and instructions for controls
- Ensure accessible content



<section> and <article>

- Each should generally begin with a heading that describes that section
- <article> is self-contained, syndicable
- HTML5 outlining algorithm is not supported and is likely to be removed, so do not rely on it. Use proper heading levels!

ARIA Roles

- Avoid duplicating default roles
(`<button role="button">`)
- ARIA roles override HTML native roles, but do NOT change functionality.
- Be very careful! You can destroy accessibility by adding one attribute.

```
<input type="checkbox" role="radio">
```



Landmark Roles

```
<div role="navigation"  
aria-label="Main navigation">
```

You can add `aria-label` to differentiate multiple landmarks of the same type.

Generic Regions

```
<div role="region"  
aria-label="Filters">
```

Generic regions must have a label.

ARIA States and Properties

- Elements can have more than one aria- attribute.
- ARIA states and properties should not override HTML properties if there's a conflict.
- Be very careful!



```
<input type="checkbox"
  aria-checked="false">
```


ARIA Headings

```
<div role="heading" aria-level="2">
```

Just use `<h2>!!!`

ARIA Lists

```
<h2 id="listheader">The Spice Girls</h2>
<div role="list" aria-describedby="listheader">
  <div role="listitem">Victoria Beckham</div>
  ...
  <div role="listitem">Melanie Brown</div>
</div>
```

Could use `aria-checked=true/false/mixed` to make it interactive. Also `role=switch`.

ARIA 'fieldset/legend'

```
<p id="question">What is the air-speed velocity of  
an unladen swallow?</p>  
<div role="radiogroup" aria-labelledby="question">  
<input type="radio"...>  
</div>
```

ARIA Table Caption

```
<h2 id="salescaption">2013 Sales Data</h2>  
<p>Sales increased by ...</p>  
<table aria-labelledby="salescaption">...
```

SVG

```

```

```
<svg role="img" aria-labelledby="title">  
  <title id="title">Sales increased 10%  
  from 2010 to 2015</title>  
  ...  
</svg>
```

```
<svg role="img" aria-label="Sales increased  
10% from 2010 to 2015">  
  ...  
</svg>
```

href-less Links

```
<a id="login" onclick="login();" >Login</a>
```

Treated identical to a ``. No keyboard interactivity.

ARIA Dialogs, Expanded, Etc.

```
<button aria-haspopup="true | menu | listbox  
| tree | grid | dialog"></button>
```

```
<button aria-expanded="false">Details</  
button>
```

<https://webaim.org/presentations/2019/aria/disclosure.htm>

```
<button aria-pressed="true">Toggle  
Highlights</button>
```

<https://webaim.org/presentations/2019/aria/ariapressed.htm>

```
<a aria-current="page | step | location |  
date | time | true">
```

Windows Screen Reader Modes

- Reading / Virtual Cursor / Document
- Forms / Application

The current mode determines whether the screen reader or browser handles most keyboard commands

```
role="application"
```

Triggers forms/application mode and disables
standard screen reader shortcuts

```
role="document" enables reading mode.
```

Some ARIA roles (tree, slider, grid, tabpanel, menu, etc.) trigger forms/application mode.

You must ensure the user is aware and that the proper keyboard interactions are implemented.

Test with AND without a Windows screen reader.

Navigation menus are not application menus!

Navigation tabs (links) are not application tabs!

Data tables are not grids!

etc.

```
role="presentation"
```

```
<table role="presentation">
```

Hides native roles of elements (and all required descendants) from assistive technology. Useful on layout tables, lists, etc.

Is ignored if the element is navigable (e.g., links and controls).

```
aria-hidden="true"
```

You can't unhide a child element.

Ignored on navigable elements.

Use ARIA attributes to control visual appearance?

```
[aria-hidden=true] {display:none;}
```


Alert Role

```
<div role="alert" aria-  
live="assertive">Read me  
right now</div>
```

Element must be present in the DOM at page load, then updated.

Also `role="alertdialog"`

ARIA Live Regions

- `aria-live=assertive` - read now
- `aria-live=polite` - read at a pause
- `aria-live=off` - read when the user encounters it.

- `aria-busy`
- `aria-atomic` - read the entire region or only what has changed
- `aria-relevant` - additions, removals, text, or all
- `aria-controls`
- **Special live regions:** `alert` (important), `status` (not important), `timer` (always changing), `marquee` (same as `aria-live="polite"`), and `log` (updates added to the bottom)

ARIA Live Regions

Some highly dynamic content updates simply cannot be made accessible using ARIA

Give users control over content updates

Single Page Apps

- Document structure
 - Use structural elements (`<main>` or `role="region"`)
- Update page titles to reflect content/state
- Keyboard navigation
 - Ensure only visible elements are navigable
 - Set `focus()` when necessary
 - `$('#main').attr("tabIndex", -1).focus();`
 - Use live regions for messaging, if necessary

ARIA

With great power comes great responsibility!

Questions?



WebAIM.org