ARIA: aria-multiselectable attribute

- Description
  - Used with aria-selected
- Example
- Values
- Associated interfaces
- Associated roles
- Specifications
- See also

The `aria-multiselectable` attribute indicates that the user may select more than one item from the current selectable descendants.

Description

The default behavior of selection lists, such as select, is to be able to choose only one item or option. By default or by convention, when a user is presented with list from which they must select an item, they assume they can only select a single item unless otherwise notified. The `aria-multiselectable` attribute is the way to inform assistive technology users that they may select more than one item from the current selectable items if they so choose. Lists and trees are examples of roles that might allow users to select more than one item at a time.

> **NOTE:** When allowing for multiple selections, inform the user that multiple values are allowed and provide directions on how to provide multiple values, such as "to select more than one value, keep the control key pressed during selection."

Used with `aria-selected`

When the user does select one or more items, remember to set the selected descendants as selected with `aria-selected="true"`, and selectable descendants that are not selected have `aria-selected="false"` set. If an element is not selectable, omit the `aria-selected` attribute altogether as its presence informs the user the item is selectable.

If a tree, grid, tab list, or list box supports selection of more than one node, the element with role `grid`, `listbox`, `tablist`, or `tree` has `aria-multiselectable` set to `true`. Otherwise, `aria-multiselectable` is either set to `false` or the default value of false is implied.

Example

```
<p id="colorOptions">Choose the colors for your flag.</p>
<ul
  tabindex="0"
  role="listbox"
  aria-labelledby="colorOptions"
  aria-multiselectable="true">
  <li id="red" role="option" aria-selected="false">Red</li>
  <li id="orange" role="option" aria-selected="false">Orange</li>
  <li id="yellow" role="option" aria-selected="false">Yellow</li>
  <li id="green" role="option" aria-selected="false">Green</li>
  <li id="blue" role="option" aria-selected="false">Blue</li>
  <li id="purple" role="option" aria-selected="false">Purple</li>
  <li id="magenta" role="option" aria-selected="false">Hot pink</li>
  <li id="lightpink" role="option" aria-selected="true">Light pink</li>
  <li id="white" role="option" aria-selected="true">White</li>
  <li id="lightblue" role="option" aria-selected="true">Light blue</li>
  <li id="black" role="option" aria-selected="false">Black</li>
  <li id="brown" role="option" aria-selected="false">Brown</li>
</ul>
```

This listbox supports multiple selection so we set the element with role `listbox` with `aria-multiselectable="true"`. All selected options have `aria-selected` set to `true`. All options that are not selected but are selectable have `aria-selected` set to false. Had we included options that were disabled or otherwise not selectable, we would have omitted the `aria-selected` attribute altogether. Including the attribute, even with no value or explicitly set to `false` indicates to assistive technology users that the item is selectable.

The first rule of ARIA use is "if you can use a native feature with the semantics and behavior you require already built in, instead of repurposing an element and **adding** an ARIA role, state or property to make it accessible, then do so." Instead of creating an unordered list requiring `tabindex`, ARIA and JavaScript to turn text into selectable options, we could have used a native multiselect: the select element has a Boolean `multiple` attribute. If included, the user can select multiple options. If not, only a single option can be selected.

```
<label for="flagcolors"> Choose the colors for your flag. </label>
<select multiple id="flagcolors">
  <option value="red">Red</option>
  <option value="orange">Orange</option>
  <option value="yellow">Yellow</option>
  <option value="green">Green</option>
  <option value="blue">Blue</option>
  <option value="purple">Purple</option>
  <option value="magenta">Hot pink</option>
  <option value="lightpink" selected>Light pink</option>
  <option value="white" selected>White</option>
  <option value="lightblue" selected>Light blue</option>
  <option value="black">Black</option>
  <option value="brown">Brown</option>
</select>
```

This HTML `<select>` version is accessible and interactive, and needs no ARIA or JavaScript to function.

If the above isn't stylable to your liking, you can also create a list of selectable options with HTML checkboxes, which is also semantic, focusable, and infinitely stylable with CSS:

```
<fieldset>
  <legend>Choose the colors for your flag.</legend>
  <ul>
    <li>
      <label><input type="checkbox" name="fc" value="red" />Red</label>
    </li>
    <li>
      <label><input type="checkbox" name="fc" value="orange" />Orange</label>
    </li>
    <li>
      <label><input type="checkbox" name="fc" value="yellow" />Yellow</label>
    </li>
```

```
    <li>
      <label><input type="checkbox" name="fc" value="green" />Green</label>
    </li>
    <li>
      <label><input type="checkbox" name="fc" value="blue" />Blue</label>
    </li>
    <li>
      <label><input type="checkbox" name="fc" value="purple" />Purple</label>
    </li>
    <li>
      <label><input type="checkbox" name="fc" value="magenta" />Hot pink</label>
    </li>
    <li>
      <label
        ><input type="checkbox" name="fc" value="lightpink" checked />Light
        pink</label
      >
    </li>
    <li>
      <label
        ><input type="checkbox" name="fc" value="white" checked />White</label
      >
    </li>
    <li>
      <label
        ><input type="checkbox" name="fc" value="lightblue" checked />Light
        blue</label
      >
    </li>
    <li>
      <label><input type="checkbox" name="fc" value="black" />Black</label>
    </li>
    <li>
      <label><input type="checkbox" name="fc" value="brown" />Brown</label>
    </li>
  </ul>
</fieldset>
```

Instead of `aria-selected="true"`, include the `checked` attribute. The browser does the rest.

Values

- `true`
  - : More than one item in the widget may be selected at a time
- `false`
  - : Only one item can be selected

Associated interfaces

- Element.ariaMultiSelectable
  - : The `ariaMultiSelectable` property, part of the Element interface, reflects the value of the `aria-multiselectable` attribute.
- ElementInternals.ariaMultiSelectable
  - : The `ariaMultiSelectable` property, part of the ElementInternals interface, reflects the value of the `aria-multiselectable` attribute.

Associated roles

Used in roles:

- `grid`
- `listbox`
- `tablist`
- `tree`

Inherited into roles:

- `treegrid`

Specifications

See also

- HTML select element

- HTML option element
- HTML input element
- multiple attribute
- `aria-selected`

□□□ #3
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ cocobook-sync □□□ 2026-11-04 17:33:07 UTC