

Welcome, If you are using a screen reader we recommend switching to “Full Access Mode”.

This mode is designed to help different types of navigation:

Each page is divided into sections and each section is described by a title (headings navigation). The most important sections are assigned to a role (landmark navigation). On the top of each page you will find the quick links menu (internal link navigation). Each link is assigned to an accesskey (keyboard shortcut).

Bạn đang xem: Aria-hidden là gì

Night mode. Night mode: this mode sets low luminance. It increases readability in dark environments. It is also useful for some people with reading disabilities such as dyslexia.

Best Practices Design Editorial & Language Elements &
Widgets Forms Structure Tables ARIA About Accessibility
Characteristics

Type:

String

See Related:

aria-disabled

Used in Roles:

All elements of the base markup (Global) No role required

Value:

true/false true: Indicates that this section of the document and its children are hidden. false

(default): Indicates that this section of the document is visible.

Indicates that the element and all of its descendants are not visible or perceivable to any user as implemented by the author.

If an element is only visible after some user action, authors must set

the `aria-hidden` attribute to `true`. When the element is presented, authors must set the `aria-hidden` attribute to `false` or remove the attribute, indicating that the element is visible.

Some assistive technologies access ariainformation directly through the DOM and not through platform Supported by user’s assistive technologies as well as the accessibility features in browsers and other user agents.

”

```
data-tiptheme="tipthemeflatdarklight" data-tipdelayclose="500" data-tipeventout="mouseout" data-tipmouseleave="false" data-tipcontent="html" class="jqeasytooltip jqeasytooltip1"
```

```
id="jqeasytooltip1" title="Accessibility Supported">accessibility supported by the browser.
```

Authors must set `aria-hidden="true"` on content that is not displayed, regardless of the mechanism used to hide it. This allows assistive technologies or user agents to properly skip hidden elements in the document.

Xem thêm: Cách Nấu Sữa Ngô Tại Nhà Ngon, Cách Làm Sữa Bắp Thơm Ngon Và Đơn Giản Nhất

It is recommended that authors key visibility of elements off this attribute, rather than change visibility and separately have to remember to update this property. CSS 2 provides a way to select on attribute values. The following CSS declaration makes content visible unless the `aria-hidden` attribute is `true`; scripts need only update the value of this attribute to change visibility:

However, in order to support assistive technologies that access the DOM directly, or other authoring techniques to visibly hide content (for example, opacity or off-screen positioning), authors need to ensure the `aria-hidden` attribute is always updated accordingly when an element is shown or hidden, unless the intent of using off-screen positioning is to make the content

visible only to screen reader users and not others.

Authors may, with caution, use aria-hidden to hide visibly rendered content from assistive technologies only if the act of hiding this content is intended to improve the experience for users of assistive technologies by removing redundant or extraneous content.

Authors using aria-hidden to hide visible content from screen readers must ensure that identical or equivalent meaning and functionality is exposed to assistive technologies.

Note: Authors are advised to use extreme caution and consider a wide range of disabilities when hiding visibly rendered content from assistive technologies. For example, a sighted, dexterity-impaired individual may use voice-controlled assistive technologies to access a visual interface.

If an author hides visible link text “Go to checkout” and exposes similar, yet non-identical link text “Check out now” to the Operating systems and other platforms provide a set of interfaces that expose information about objects and events to assistive technologies. Assistive technologies use these interfaces to get information about and interact with those widgets. Examples of accessibility APIs are the Microsoft Active Accessibility

data-tiptheme="tipthemeflatdarklight" data-tipdelayclose="500" data-tipeventout="mouseout" data-tipmouseleave="false" data-tipcontent="html" class="jqeasytooltip jqeasytooltip0" id="jqeasytooltip0" title="Accessibility API">accessibility API, the user may be unable to access the interface they perceive using voice control. Similar problems may also arise for screen reader users. For example, a sighted telephone support technician may attempt to have the blind screen reader user click the “Go to checkout” link, which they may be unable to find using a type-ahead item search (“Go to...”).

Note: Authors are advised to avoid using aria-hidden="false" with styles or attributes that have historically prevented rendering in all modalities, such as display:none or visibility:hidden in CSS, or the hidden attribute in HTML 5.

At the time of this writing, aria-hidden="false" is known to work inconsistently when used in conjunction with such features. As future implementations improve, use caution and test thoroughly before relying on this approach.

Xem thêm: It Help Desk Là Gì – Sự Khác Nhau Giữa It Service Desk Và

Note: For cross-browser compatibility, always use the aria attribute syntax to access and modify aria properties, for example object.setAttribute(“aria-valuenow”, newValue).

Syntax

HTMLJavaScriptobject.setAttribute(“aria-hidden”,value);var value = object.getAttribute(“aria-hidden”);

Microsoft Active Accessibility Properties:

Chuyên mục: Hỏi Đáp

THAM KHẢO THÊM: <https://bongdanews.top/>

The post [Aria-Hidden Là Gì – Sử Dụng Thuộc Tính Aria](#) appeared first on [BONGDANEWS.TOP](https://bongdanews.top/).

via BONGDANEWS.TOP <https://bongdanews.top/aria-hidden-la-gi-su-dung-thuoc-tinh-aria/>