

<b>defaultChecked</b>	boolean	-
<b>defaultValue</b>	string   number   string[]	-
<b>suppressContentEditableWarning</b>	boolean	-
<b>suppressHydrationWarning</b>	boolean	-
<b>accessKey</b>	string	-
<b>contentEditable</b>	boolean   "true"   "false"   "inherit"	-
<b>contextMenu</b>	string	-
<b>dir</b>	string	-
<b>draggable</b>	Booleanish	-
<b>hidden</b>	boolean	-
<b>id</b>	string	-
<b>lang</b>	string	-
<b>placeholder</b>	string	-
<b>slot</b>	string	-
<b>spellCheck</b>	Booleanish	-
<b>style</b>	CSSProperties	-
<b>tabIndex</b>	number	-
<b>title</b>	string	-
<b>translate</b>	"yes"   "no"	-
<b>radioGroup</b>	string	-
<b>role</b>	string	-
<b>about</b>	string	-
<b>datatype</b>	string	-
<b>inlist</b>	any	-
<b>prefix</b>	string	-
<b>property</b>	string	-
<b>resource</b>	string	-
<b>typeof</b>	string	-
<b>vocab</b>	string	-
<b>autoCapitalize</b>	string	-
<b>autoCorrect</b>	string	-
<b>autoSave</b>	string	-
<b>color</b>	string	-
<b>itemProp</b>	string	-
<b>itemScope</b>	boolean	-
<b>itemType</b>	string	-
<b>itemID</b>	string	-
<b>itemRef</b>	string	-
<b>results</b>	number	-
<b>security</b>	string	-
<b>unselectable</b>	"on"   "off"	-

<b>inputMode</b>	Hints at the type of data that might be entered by the user while editing the element or its contents "none"   "text"   "tel"   "url"   "email"   "numeric"   "decimal"   "search"	-
<b>is</b>	Specify that a standard HTML element should behave like a defined custom built-in element string	-
<b>aria-activedescendant</b>	Identifies the currently active element when DOM focus is on a composite widget, textbox, group, or application. string	-
<b>aria-atomic</b>	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. Booleanish	-
<b>aria-autocomplete</b>	Indicates whether inputting text could trigger display of one or more predictions of the user's intended value for an input and specifies how predictions would be presented if they are made. "none"   "inline"   "list"   "both"	-
<b>aria-busy</b>	Indicates an element is being modified and that assistive technologies MAY want to wait until the modifications are complete before exposing them to the user. Booleanish	-
<b>aria-checked</b>	Indicates the current "checked" state of checkboxes, radio buttons, and other widgets. boolean   "true"   "false"   "mixed"	-
<b>aria-colcount</b>	Defines the total number of columns in a table, grid, or treegrid. number	-
<b>aria-colindex</b>	Defines an element's column index or position with respect to the total number of columns within a table, grid, or treegrid. number	-
<b>aria-colspan</b>	Defines the number of columns spanned by a cell or gridcell within a table, grid, or treegrid. number	-
<b>aria-controls</b>	Identifies the element (or elements) whose contents or presence are controlled by the current element. string	-
<b>aria-current</b>	Indicates the element that represents the current item within a container or set of related elements. boolean   "true"   "false"   "page"   "step"   "location"   "date"	-
<b>aria-describedby</b>	Identifies the element (or elements) that describes the object. string	-
<b>aria-details</b>	Identifies the element that provides a detailed, extended description for the object. string	-
<b>aria-disabled</b>	Indicates that the element is perceivable but disabled, so it is not editable or otherwise operable. Booleanish	-
<b>aria-dropeffect</b>	Indicates what functions can be performed when a dragged object is released on the drop target. "none"   "link"   "copy"   "execute"   "move"   "popup"	-
<b>aria-errormessage</b>	Identifies the element that provides an error message for the object. string	-
<b>aria-expanded</b>	Indicates whether the element, or another grouping element it controls, is currently expanded or collapsed. Booleanish	-
<b>aria-flowto</b>	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. string	-
<b>aria-grabbed</b>	Indicates an element's "grabbed" state in a drag-and-drop operation. Booleanish	-
<b>aria-haspopup</b>	Indicates the availability and type of interactive popup element, such as menu or dialog, that can be triggered by an element. boolean   "true"   "false"   "dialog"   "menu"   "listbox"   "tree"   "grid"	-
<b>aria-hidden</b>	Indicates whether the element is exposed to an accessibility API. Booleanish	-

<b>aria-invalid</b>	Indicates the entered value does not conform to the format expected by the application. boolean   "true"   "false"   "grammar"   "spelling"	-
<b>aria-keyshortcuts</b>	Indicates keyboard shortcuts that an author has implemented to activate or give focus to an element. string	-
<b>aria-label</b>	Defines a string value that labels the current element. string	-
<b>aria-labelledby</b>	Identifies the element (or elements) that labels the current element. string	-
<b>aria-level</b>	Defines the hierarchical level of an element within a structure. number	-
<b>aria-live</b>	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. "off"   "assertive"   "polite"	-
<b>aria-modal</b>	Indicates whether an element is modal when displayed. Booleanish	-
<b>aria-multiline</b>	Indicates whether a text box accepts multiple lines of input or only a single line. Booleanish	-
<b>aria-multiselectable</b>	Indicates that the user may select more than one item from the current selectable descendants. Booleanish	-
<b>aria-orientation</b>	Indicates whether the element's orientation is horizontal, vertical, or unknown/ambiguous. "horizontal"   "vertical"	-
<b>aria-owns</b>	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. string	-
<b>aria-placeholder</b>	Defines a short hint (a word or short phrase) intended to aid the user with data entry when the control has no value. A hint could be a sample value or a brief description of the expected format. string	-
<b>aria-posinset</b>	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. number	-
<b>aria-pressed</b>	Indicates the current "pressed" state of toggle buttons. boolean   "true"   "false"   "mixed"	-
<b>aria-readonly</b>	Indicates that the element is not editable, but is otherwise operable. Booleanish	-
<b>aria-relevant</b>	Indicates what notifications the user agent will trigger when the accessibility tree within a live region is modified. "text"   "additions"   "additions text"   "all"   "removals"	-
<b>aria-required</b>	Indicates that user input is required on the element before a form may be submitted. Booleanish	-
<b>aria-roledescription</b>	Defines a human-readable, author-localized description for the role of an element. string	-
<b>aria-rowcount</b>	Defines the total number of rows in a table, grid, or treegrid. number	-
<b>aria-rowindex</b>	Defines an element's row index or position with respect to the total number of rows within a table, grid, or treegrid. number	-
<b>aria-rowspan</b>	Defines the number of rows spanned by a cell or gridcell within a table, grid, or treegrid. number	-
<b>aria-selected</b>	Indicates the current "selected" state of various widgets. Booleanish	-
<b>aria-setsize</b>	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. number	-
<b>aria-sort</b>	Indicates if items in a table or grid are sorted in ascending or descending order. "none"   "ascending"   "descending"   "other"	-

<b>aria-valuemax</b>	Defines the maximum allowed value for a range widget. number	-
<b>aria-valuemin</b>	Defines the minimum allowed value for a range widget. number	-
<b>aria-valuenow</b>	Defines the current value for a range widget. number	-
<b>aria-valuetext</b>	Defines the human readable text alternative of aria-valuenow for a range widget. string	-
<b>dangerouslySetInnerHTML</b>	{ __html: string; }	-
<b>onCopy</b>	((event: ClipboardEvent<HTMLButtonElement>) => void)   ((event: ClipboardEvent<HTMLAnchorElement>) => void)	-
<b>onCopyCapture</b>	((event: ClipboardEvent<HTMLButtonElement>) => void)   ((event: ClipboardEvent<HTMLAnchorElement>) => void)	-
<b>onCut</b>	((event: ClipboardEvent<HTMLButtonElement>) => void)   ((event: ClipboardEvent<HTMLAnchorElement>) => void)	-
<b>onCutCapture</b>	((event: ClipboardEvent<HTMLButtonElement>) => void)   ((event: ClipboardEvent<HTMLAnchorElement>) => void)	-
<b>onPaste</b>	((event: ClipboardEvent<HTMLButtonElement>) => void)   ((event: ClipboardEvent<HTMLAnchorElement>) => void)	-
<b>onPasteCapture</b>	((event: ClipboardEvent<HTMLButtonElement>) => void)   ((event: ClipboardEvent<HTMLAnchorElement>) => void)	-
<b>onCompositionEnd</b>	((event: CompositionEvent<HTMLButtonElement>) => void)   ((event: CompositionEvent<HTMLAnchorElement>) => void)	-
<b>onCompositionEndCapture</b>	((event: CompositionEvent<HTMLButtonElement>) => void)   ((event: CompositionEvent<HTMLAnchorElement>) => void)	-
<b>onCompositionStart</b>	((event: CompositionEvent<HTMLButtonElement>) => void)   ((event: CompositionEvent<HTMLAnchorElement>) => void)	-
<b>onCompositionStartCapture</b>	((event: CompositionEvent<HTMLButtonElement>) => void)   ((event: CompositionEvent<HTMLAnchorElement>) => void)	-
<b>onCompositionUpdate</b>	((event: CompositionEvent<HTMLButtonElement>) => void)   ((event: CompositionEvent<HTMLAnchorElement>) => void)	-
<b>onCompositionUpdateCapture</b>	((event: CompositionEvent<HTMLButtonElement>) => void)   ((event: CompositionEvent<HTMLAnchorElement>) => void)	-
<b>onFocus</b>	((event: FocusEvent<HTMLButtonElement>) => void)   ((event: FocusEvent<HTMLAnchorElement>) => void)	-
<b>onFocusCapture</b>	((event: FocusEvent<HTMLButtonElement>) => void)   ((event: FocusEvent<HTMLAnchorElement>) => void)	-
<b>onBlur</b>	((event: FocusEvent<HTMLButtonElement>) => void)   ((event: FocusEvent<HTMLAnchorElement>) => void)	-
<b>onBlurCapture</b>	((event: FocusEvent<HTMLButtonElement>) => void)   ((event: FocusEvent<HTMLAnchorElement>) => void)	-
<b>onChange</b>	((event: FormEvent<HTMLButtonElement>) => void)   ((event: FormEvent<HTMLAnchorElement>) => void)	-
<b>onChangeCapture</b>	((event: FormEvent<HTMLButtonElement>) => void)   ((event: FormEvent<HTMLAnchorElement>) => void)	-
<b>onBeforeInput</b>	((event: FormEvent<HTMLButtonElement>) => void)   ((event: FormEvent<HTMLAnchorElement>) => void)	-
<b>onBeforeInputCapture</b>	((event: FormEvent<HTMLButtonElement>) => void)   ((event: FormEvent<HTMLAnchorElement>) => void)	-
<b>onInput</b>	((event: FormEvent<HTMLButtonElement>) => void)   ((event: FormEvent<HTMLAnchorElement>) => void)	-
<b>onInputCapture</b>	((event: FormEvent<HTMLButtonElement>) => void)   ((event: FormEvent<HTMLAnchorElement>) => void)	-
<b>onReset</b>	((event: FormEvent<HTMLButtonElement>) => void)   ((event: FormEvent<HTMLAnchorElement>) => void)	-
<b>onResetCapture</b>	((event: FormEvent<HTMLButtonElement>) => void)   ((event: FormEvent<HTMLAnchorElement>) => void)	-













<b>onAnimationIterationCapture</b>	((event: AnimationEvent<HTMLButtonElement>) => void)   ((event: AnimationEvent<HTMLAnchorElement>) => void)	-
<b>onTransitionEnd</b>	((event: TransitionEvent<HTMLButtonElement>) => void)   ((event: TransitionEvent<HTMLAnchorElement>) => void)	-
<b>onTransitionEndCapture</b>	((event: TransitionEvent<HTMLButtonElement>) => void)   ((event: TransitionEvent<HTMLAnchorElement>) => void)	-
<b>href</b>	string	-
<b>target</b>	string	-