Element: scrollend event

- Syntax
- Event type
- Example
  - Using scrollend with an event listener
  - Using onscrollend event handler property
- Specifications
- Browser compatibility
- See also

The **`scrollend`** event fires when element scrolling has completed. Scrolling is considered completed when the scroll position has no more pending updates and the user has completed their gesture.

Scroll position updates include smooth or instant mouse wheel scrolling, keyboard scrolling, scroll-snap events, or other APIs and gestures which cause the scroll position to update. User gestures like touch panning or trackpad scrolling aren't complete until pointers or keys have released. If the scroll position did not change, then no scrollend event fires.

For detecting when scrolling inside a Document is complete, see the scrollend event of `Document`.

Syntax

Use the event name in addEventListener():

```
addEventListener("scrollend", (event) => { })
```

Alternatively, assign to the event handler property of the interface:

```
onscrollend = (event) => { }
```

Event type

A generic Event.

Example

Using `scrollend` with an event listener

The following example shows how to use the `scrollend` event to detect when the user has stopped scrolling:

```
#scroll-box {
  height: 100px;
  width: 100px;
  float: left;
  overflow: scroll;
  outline: 4px dotted;
  margin: 4px;
}

#scroll-box-title {
  position: fixed;
  top: 5px;
  left: 5px;
  transform: translateX(0);
}

#large-element {
  height: 200px;
  width: 200px;
}

#output {
  text-align: center;
}
```

```
<div id="scroll-box">
  <p id="scroll-box-title">Scroll me!</p>
  <p id="large-element"></p>
</div>
<p id="output">Waiting on scroll events...</p>
```

```
const element = document.querySelector("div#scroll-box");
const output = document.querySelector("p#output");

element.addEventListener("scroll", (event) => {
  output.textContent = "scroll event fired, waiting for scrollend...";
});

element.addEventListener("scrollend", (event) => {
  output.textContent = "scrollend event fired!";
});
```

Using `onscrollend` event handler property

The following example shows how to use the `onscrollend` event handler property to detect when the user has stopped scrolling (but note that this replaces other event listeners added using `onscroll` and `onscrollend`, if any):

```
#scroll-box {
  height: 100px;
  width: 100px;
  float: left;
  overflow: scroll;
  outline: 4px dotted;
  margin: 4px;
}

#scroll-box-title {
  position: fixed;
  top: 5px;
  left: 5px;
  transform: translateX(0);
}
```

```
#large-element {
  height: 200px;
  width: 200px;
}

#output {
  text-align: center;
}
```

```
<div id="scroll-box">
  <p id="scroll-box-title">Scroll me!</p>
  <p id="large-element"></p>
</div>
<p id="output">Waiting on scroll events...</p>
```

```
const element = document.querySelector("div#scroll-box");
const output = document.querySelector("p#output");

element.onscroll = (event) => {
  output.textContent = "Element scroll event fired, waiting for scrollend...";
};

element.onscrollend = (event) => {
  output.textContent = "Element scrollend event fired!";
};
```

Specifications

Browser compatibility

See also

- Element `scroll` event
- Document `scrollend` event
- Document `scroll` event

□□□ #3
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ cocobook-sync □□□ 2026-11-04 17:34:34 UTC