ARIA: timer role

- Description
  - Associated WAI-ARIA roles, states, and properties
- Accessibility concerns
- Examples
  - A basic timer
- Specifications
- See also

The `timer` role indicates to assistive technologies that an element is a numerical counter listing the amount of elapsed time from a starting point or the remaining time until an end point. Assistive technologies will not announce updates to a timer as it has an implicit `aria-live` value of `off`.

```
<div role="timer" id="eggtimer">0</div>
```

This defines this `div` element as a timer with no remaining time.

Description

The `timer` role indicates to assistive technologies that this part of the web content is a live region containing a timer listing the time remaining or elapsed time. A timer's inner text should be an updating current time measurement. While the value does not necessarily need to be machine parsable, it should continuously update at regular intervals unless the timer is paused or reaches its end-point.

Along with `alert`, `log`, `marquee`, and `status` the `timer` role is a live region and can be modified by live region attributes.

Associated WAI-ARIA roles, states, and properties

- `aria-label` or `aria-labelledby`
  - : Some screen readers announce the name of a timer element before announcing its contents. If a name is visible, reference it using `aria-labelledby`. Including an `aria-label` provides a method for prefacing the visible content of a timer element with text that is not displayed when a screen reader reads the content. Naming a timer is not required so if nothing is appropriate both these attributes can be omitted.
- `aria-live`
  - : Elements with the role `timer` have an implicit `aria-live` value of `off`.

Accessibility concerns

If a time limit needs to be in place, for example, for security reasons, the user should have the option to turn it off or extend it. This restriction does not apply if the time limit is due to a live event, such as an auction or a game, or if the time to complete the form is essential for a valid submission.

Examples

A basic timer

This example has a timer that counts down from 30 seconds to 0 seconds. The whole time display region has `role="timer"`, and also `aria-atomic` to indicate that the region should be announced as a whole, not just the changed regions. Due to the implicit `aria-live="off"`, changes are not announced by default; we manually change the role to `"alert"` when the timer reaches 10 seconds remaining so it gets announced once.

```
<div id="countdown" role="timer" aria-atomic="true">
  <span id="number">30</span> seconds left!
</div>
```

```
html {
  font-size: 50px;
  text-align: center;
  margin-top: 1em;
  font-family: sans-serif;
}

#number {
```

```
  font-family: monospace;
  color: #cc0000;
  font-weight: bold;
  font-size: 1.25em;
  vertical-align: middle;
}
```

```
const numElement = document.getElementById("number");
const liveRegion = document.getElementById("countdown");
let startTime = new Date().getTime();

function decrement() {
  const timeNow = new Date().getTime();
  const elapsedTime = Math.floor((timeNow - startTime) / 1000);
  let newNumber = 30 - elapsedTime;

  if (newNumber >= 0) {
    numElement.textContent = newNumber;
  }

  if (newNumber === 10) {
    liveRegion.setAttribute("role", "alert");
    setTimeout(() => {
      liveRegion.setAttribute("role", "timer");
    }, 999);
  }
}

window.setInterval(() => {
  decrement();
}, 500);
```

Specifications

See also

- ARIA: `alert` role

- ARIA: `log` role
- ARIA: `marquee` role
- ARIA: `status` role
- ARIA live regions

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