

HowTo: Components – howto-tooltip



By [Ewa Gasperowicz](#)

Ewa is a contributor to WebFundamentals



By [Rob Dodson](#)

Rob is a contributor to WebFundamentals



By [Surma](#)

Surma is a contributor to WebFundamentals

Summary

A `<howto-tooltip>` is a popup that displays information related to an element when the element receives keyboard focus or the mouse hovers over it. The element that triggers the tooltip references the tooltip element with `aria-describedby`.

The element self-applies the role `tooltip` and sets `tabindex` to `-1`, as the tooltip itself can never be focused.

Reference

- [HowTo: Components on GitHub](#)
- [Tooltip pattern in ARIA Authoring Practices 1.1](#)

Demo

[View live demo on GitHub](#)

Example usage

```
<div class="text">
  <label for="name">Your name:</label>
  <input id="name" aria-describedby="tp1"/>
  <howto-tooltip id="tp1">Ideally your name is Batman</howto-tooltip>
  <br>
  <label for="cheese">Favourite type of cheese: </label>
  <input id="cheese" aria-describedby="tp2"/>
  <howto-tooltip id="tp2">Help I am trapped inside a tooltip message</howto-tool
</div>
```

Code

```
class HowtoTooltip extends HTMLElement {
```

The constructor does work that needs to be executed *exactly* once.

```
  constructor() {
    super();
```

These functions are used in a bunch of places, and always need to bind the correct `this` reference, so do it once.

```
    this._show = this._show.bind(this);
    this._hide = this._hide.bind(this);
  }
```

`connectedCallback()` fires when the element is inserted into the DOM. It's a good place to set the initial `role`, `tabindex`, internal state, and install event listeners.

```
  connectedCallback() {
    if (!this.hasAttribute('role'))
```

```
    this.setAttribute('role', 'tooltip');
```

```
    if (!this.hasAttribute('tabindex'))  
        this.setAttribute('tabindex', -1);
```

```
    this._hide();
```

The element that triggers the tooltip
references the tooltip element with
aria-describedby.

```
    this._target = document.querySelector('[aria-describedby=' + this.id +  
    if (!this._target)  
        return;
```

The tooltip needs to listen to
focus/blur events from the target,
as well as **hover** events over the
target.

```
    this._target.addEventListener('focus', this._show);  
    this._target.addEventListener('blur', this._hide);  
    this._target.addEventListener('mouseenter', this._show);  
    this._target.addEventListener('mouseleave', this._hide);  
}
```

disconnectedCallback()
unregisters the event listeners that
were set up in
connectedCallback().

```
disconnectedCallback() {
```

```
    if (!this._target)  
        return;
```

Remove the existing listeners, so
that they don't trigger even though
there's no tooltip to show.

```
    this._target.removeEventListener('focus', this._show);
```

```
    this._target.removeEventListener('blur', this._hide);
    this._target.removeEventListener('mouseenter', this._show);
    this._target.removeEventListener('mouseleave', this._hide);
    this._target = null;
}

_show() {
    this.hidden = false;
}

_hide() {
    this.hidden = true;
}
}

window.customElements.define('howto-tooltip', HowtoTooltip);
```

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 2, 2018.