

# Popover, Toast, and Utilities

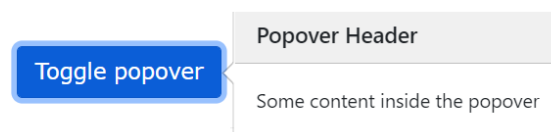
## Topics Covered:

- What is Popover?
- How to create a Popover?
- Positioning Popover.
- Closing Popover.
- Hoverable Popover.
- What is Toast?
- How to open Toast?
- Utilities - Helper Class.

## Topics in Detail:

### What is Popover?

- The Popover component is similar to tooltips
- Popover is a pop-up box that appears when the user clicks on an element.
- The difference is that the popover can contain much more content.



### How to create a Popover?

- To create a popover, add the ***data-bs-toggle="popover"*** attribute to an element.
- The ***title*** attribute is used to specify the header text of the popover, and use the ***data-bs-content*** attribute to specify the text that should be displayed inside the popover's body.
- Popovers must be initialized with JavaScript to work.
- HTML:

```
<button type="button" class="btn btn-primary" data-bs-toggle="popover" title="Popover Header" data-bs-content="Some content inside the popover">Toggle popover</button>
```

- JS:

```
<script>
var popoverTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="popover"]'))
var popoverList = popoverTriggerList.map(function (popoverTriggerEl) {
  return new bootstrap.Popover(popoverTriggerEl)
})
</script>
```

## Positioning Popovers:

- By default, the popover will appear on the right side of the element.
- The **`data-bs-placement`** attribute is used to set the position of the popover on top, bottom, left or the right side of the element.
- The placement attributes do not work as you expect if there is not enough room for them.
- If you use the top placement at the top of a page (where there is no room for it), it will instead display the popover below the element or to the right (wherever there is room for it).
- HTML:

```
<div class="container mt-3">
  <a href="#" title="Header" data-bs-toggle="popover" data-bs-placement="top" data-content="Content">Top</a>
  <a href="#" title="Header" data-bs-toggle="popover" data-bs-placement="bottom" data-content="Content">Bottom</a>
  <a href="#" title="Header" data-bs-toggle="popover" data-bs-placement="left" data-content="Content">Left</a>
  <a href="#" title="Header" data-bs-toggle="popover" data-bs-placement="right" data-content="Content">Right</a>
</div>
```

- JS:

```
<script>
var popoverTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="popover"]'))
var popoverList = popoverTriggerList.map(function (popoverTriggerEl) {
  return new bootstrap.Popover(popoverTriggerEl)
})
</script>
```

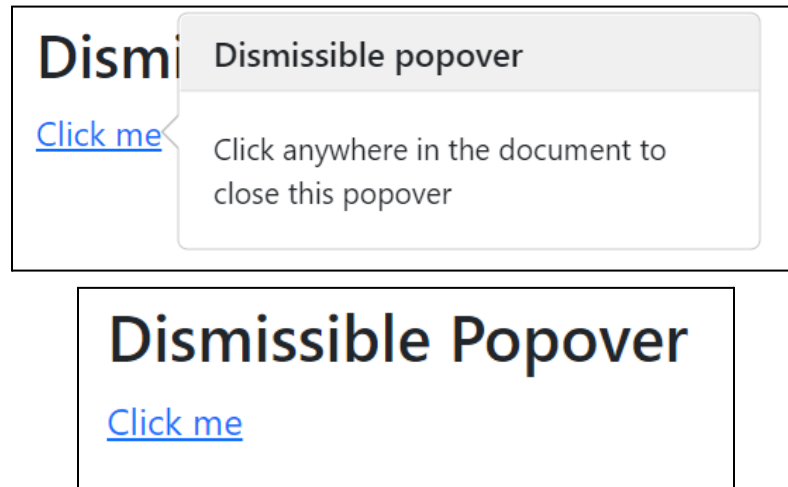
- Output:



## Closing Popover:

- By default, the popover is closed when you click on the element again.

- The ***data-bs-trigger="focus"*** attribute can be used to close the popover when clicking outside the element.

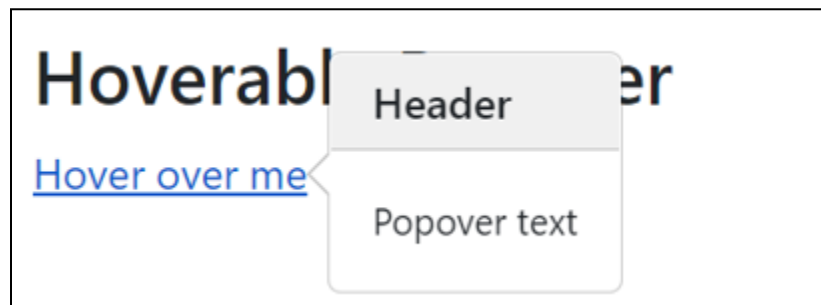


### Hoverable Popover:

- The ***data-bs-trigger*** attribute with a value of ***"hover"*** can be used to display the popover when you move the mouse pointer over the element.
- Code:

```
<a href="#" title="Header" data-bs-toggle="popover" data-bs-trigger="hover" data-bs-content="Popover text">Hover over me</a>
```

- Output:



### What is Toast?

The toast component is like an alert box that is only shown for a couple of seconds when something happens (i.e. when the user clicks on a button, submits a form, etc.).

How to create a Toast?

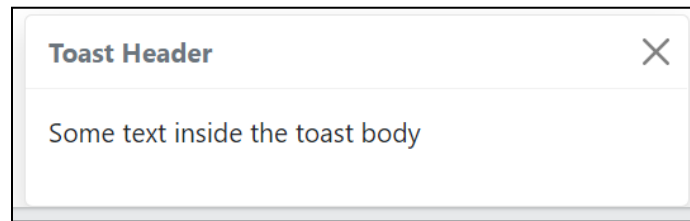
- The ***.toast*** class is used to create a Toast. And to add the header and body use a ***.toast-header*** and a ***.toast-body*** class inside it.
- Toasts are hidden by default. The ***.show*** class should be used to display it.
- To close it, a ***<button>*** element should be used and attribute ***data-bs-dismiss="toast"*** should be added.
- Code:

```

<div class="toast show">
  <div class="toast-header">
    Toast Header
    <button type="button" class="btn-close" data-bs-dismiss="toast"></button>
  </div>
  <div class="toast-body">
    Some text inside the toast body
  </div>
</div>

```

- Output:



## How to Open a Toast?

- To show a toast with a click of a button, you must initialize it with JavaScript: select the specified element and call the **toast()** method.
- HTML:

```

<div class="container mt-3">
  <h3>Toast Example</h3>
  <p>In this example, we use a button to show the toast message.</p>

  <button type="button" class="btn btn-primary" id="toastbtn">Show Toast</button>

  <div class="toast">
    <div class="toast-header">
      <strong class="me-auto">Toast Header</strong>
      <button type="button" class="btn-close" data-bs-dismiss="toast"></button>
    </div>
    <div class="toast-body">
      <p>Some text inside the toast body</p>
    </div>
  </div>
</div>

```

- JS:

```

<script>
document.getElementById("toastbtn").onclick = function() {
  var toastElList = [].slice.call(document.querySelectorAll('.toast'))
  var toastList = toastElList.map(function(toastEl) {
    return new bootstrap.Toast(toastEl)
  })
  toastList.forEach(toast => toast.show())
}
</script>

```

- Output:

## Toast Example

In this example, we use a button to show the toast message.

Show Toast

Toast Header



Some text inside the toast body

Utilities - Helper Class:

Class Name	Description
.border	To add or remove borders of the element.
.border-1 to .border-5	To change the width of the border.
.float-end	Float an element to the right.
.float-start	Float an element to the left.
.clearfix	To clear floats to the next element.
.mx-auto	To center align - margin-left and margin-right: auto.
.align	To change the alignment of elements.
.visible	To control the visibility of elements.
.invisible	To control the visibility of elements.
.btn-close	To style a close icon. This is often used for alerts and modals.
.visually-hidden	To hide an element on all devices, except screen readers.