

ACCESSIBLE BUTTON DROPDOWNS

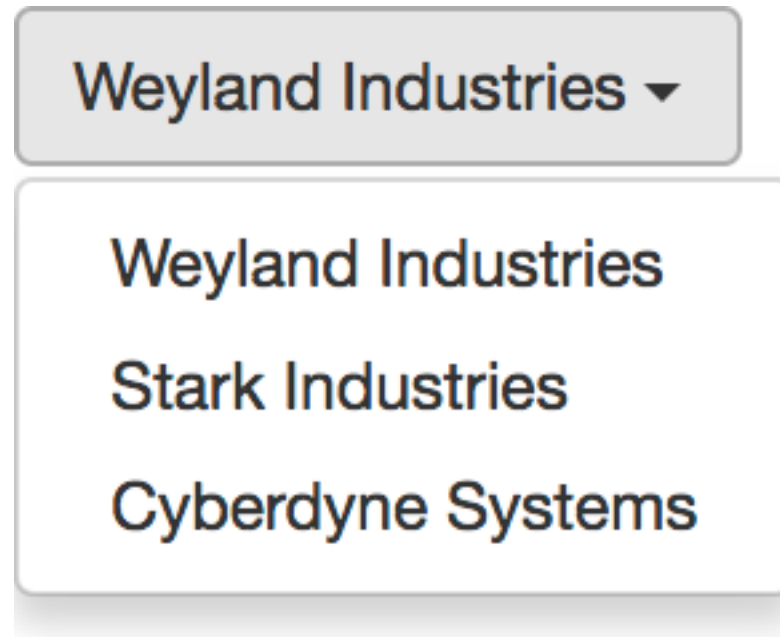
Introduction

One web component that can present accessibility issues is the **button dropdown**.

A button dropdown is where a button is used to **trigger the display of contextual menus** - such as a list of links.

Weyland Industries ▼

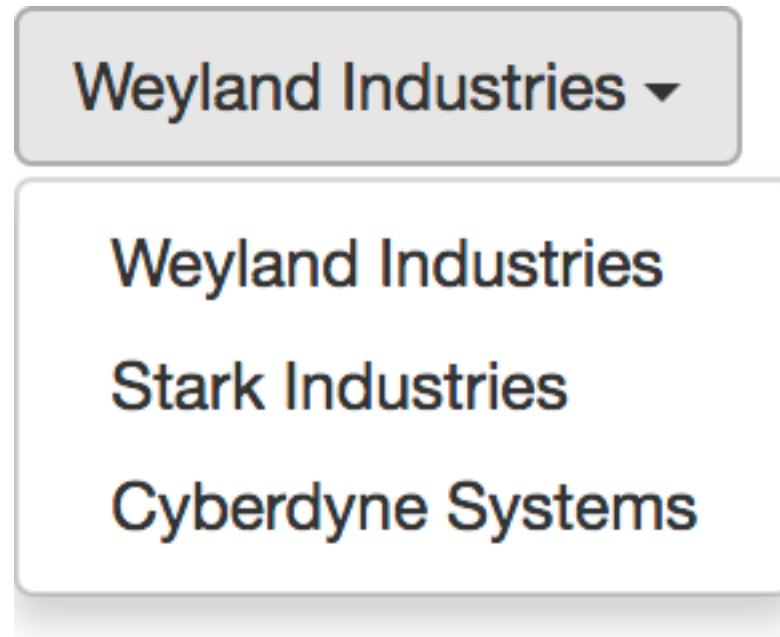
A diagram showing a button “Weyland Industries” with no dropdown displayed



A diagram showing a button “Weyland Industries” with the dropdown displayed below. The dropdown displays three options: “Weyland Industries”, “Stark Industries” and “Cyberdyne Systems”

We're going to look at how to make
button dropdowns **more accessible**.

We'll use an imaginary example of a button dropdown that allows users to **switch their current organisation.**



A diagram showing a button “Weyland Industries” with the dropdown displayed below. The dropdown displays three options: “Weyland Industries”, “Stark Industries” and “Cyberdyne Systems”

1. Semantics

As the name suggests, the ideal element that **should be used for the trigger**, is the `<button>` element.

```
<button type="button">  
  Weyland Industries  
</button>
```

The `<button>` element should be **used to trigger something** (submit a form, open an accordion).

The `<a>` element should **send users somewhere** (go to a different location).

The markup for the dropdown depends on the role of the items inside. If the items are a list of links, then **an unordered list of links is ideal.**

```
<ul>
```

```
  <li><a href="#">Weyland Industries</a></li>
```

```
  <li><a href="#">Stark Industries</a></li>
```

```
  <li><a href="#">Cyberdyne Systems</a></li>
```

```
</ul>
```


2.

Getting to the
button

Keyboard-only users **should be able to**
TAB to the button, which then receives
focus.

3.

Announcing the
button

The `aria-label` attribute can be used to announce the button value **along with any additional information** to screen reader users.

For modern screen readers, the `aria-label` value will be announced **instead of** the button value.

```
<button  
  type="button"  
  aria-label="Current company: Weyland Industries. Use the  
  dropdown menu to switch companies"  
>  
  Weyland Industries  
</button>
```

The `aria-haspopup="true"` attribute can be used to announce the button as a **“popup button”** to screen readers.

This is important, as it tells screen reader users that it is **a different type of button** - not a normal button associated with submitting a form etc.


```
<button  
  type="button"  
  aria-label="Current company: Weyland Industries. Use the  
  dropdown menu to switch companies"  
  aria-haspopup="true"  
>  
  Weyland Industries  
</button>
```

The `aria-expanded="false"` attribute can be used to announce the **current state of the popup button** to screen readers - i.e the dropdown below the button is not currently expanded.

The "false" value would **need to be changed** to "true" via JavaScript as soon as the user triggers the button.

```
<button
  type="button"
  aria-label="Current company: Weyland Industries. Use the
  dropdown menu to switch companies"
  aria-haspopup="true"
  aria-expanded="false"
>
  Weyland Industries
</button>
```

4.

Triggering the
button

For keyboard-only users, **ENTER** or **SPACEBAR** strokes should **trigger the dropdown to appear**.

5.

After the button is
triggered

If the `aria-expanded="false"` attribute **is present** in the default button state, it should be changed to `aria-expanded="true"` via JavaScript.


```
<button
  type="button"
  aria-label="Current company: Weyland Industries. Use the
  dropdown menu to switch companies"
  aria-haspopup="true"
  aria-expanded="true"
>
  Weyland Industries
</button>
```

Focus should **immediately shift** to the `<u1>` element and the dropdown should become visible.

This is something that most button dropdown solutions **do not solve elegantly**. In many cases, users trigger the button but the focus does not shift at all.

Users are either given silence after they trigger the button, or the button information is repeated again. This can cause confusion for users who cannot see that the dropdown has been triggered, but **nothing has been announced.**

The `` element could be given an `aria-label` value, which means that when it receives focus, **it's purpose is announced.**

```
<ul aria-label="Switch Companies">  
  <li><a href="#">Weyland Industries</a></li>  
  <li><a href="#">Stark Industries</a></li>  
  <li><a href="#">Cyberdyne Systems</a></li>  
</ul>
```

If the current organisation exists in the long list of dropdown items, it may be a good idea to **flag this item as the current organisation** for screen reader users.

This can be achieved with a range of different methods, including **providing additional information** that is hidden off-screen.


```
<ul aria-label="Switch Companies">
  <li>
    <a href="#">
      <span class="hidden">Current company: </span>
      Weyland Industries
    </a>
  </li>
  <li><a href="#">Stark Industries</a></li>
  <li><a href="#">Cyberdyne Systems</a></li>
</ul>
```

8.

To escape the
dropdown

For keyboard-only users, the **ESC** keystroke should close the dropdown and **return focus to the button.**

7.

To navigate through
items within the
dropdown

When focus has shifted to the `` element, keyboard-only users should be able to use `TAB`, `SHIFT TAB`, `UP ARROW` or `DOWN ARROW` to move **forwards or backwards through the list items.**

When users reach the start or end of the list, UP ARROW and DOWN ARROW keystrokes should then have **not have any effect.**

8.

Selecting a
dropdown item

Keyboard-only users should be able to **select a dropdown menu item** using **ENTER** and possibly the **SPACEBAR** keystrokes.

9.

To leave the
dropdown

Keyboard-only users should be able to **TAB** forward through the dropdown items and then **on to other focusable items outside the dropdown.**

As soon focus leave the last dropdown item, **the dropdown should disappear.**

Users should be able to **SHIFT TAB**
backwards through the dropdown items
and **back to the button**.

The dropdown should **remain open** when the button receives focus. (Users can close the dropdown by triggering the button again).