Links

The Link component allows you to easily customize anchor elements with your theme colors and typography styles.

{{"component": "modules/components/ComponentLinkHeader.js"}}

Basic links

The Link component is built on top of the [Typography](#) component, meaning that you can use its props.

{{"demo": "Links.js"}}

However, the Link component has some different default props than the Typography component:

- `color="primary"` as the link needs to stand out.
- `variant="inherit"` as the link will, most of the time, be used as a child of a Typography component.

Underline

The `underline` prop can be used to set the underline behavior. The default is `always`.

{{"demo": "UnderlineLink.js"}}

Security

When you use `target="_blank"` with Links, it is [recommended](#) to always set `rel="noopener"` or `rel="noreferrer"` when linking to third party content.

- `rel="noopener"` prevents the new page from being able to access the `window.opener` property and ensures it runs in a separate process. Without this, the target page can potentially redirect your page to a malicious URL.
- `rel="noreferrer"` has the same effect, but also prevents the *Referer* header from being sent to the new page. ⚠️ Removing the referrer header will affect analytics.

Third-party routing library

One frequent use case is to perform navigation on the client only, without an HTTP round-trip to the server. The `Link` component provides the `component` prop to handle this use case. Here is a [more detailed guide](#).

Accessibility

(WAI-ARIA: https://www.w3.org/TR/wai-aria-practices/#link)

- When providing the content for the link, avoid generic descriptions like "click here" or "go to". Instead, use [specific descriptions](#).
- For the best user experience, links should stand out from the text on the page. For instance, you can keep the default `underline="always"` behavior.
- If a link doesn't have a meaningful href, [it should be rendered using a `<button>` element](#).

{{"demo": "ButtonLink.js"}}