Polymorphic components

PRACTICAL USE CASES

1. Flexibility and accessibility

Imagine if you built a **Text** component to be reused by other engineers.

```
<Text />
```

Instead of exposing different components for different text types, your single component can be flexible enough to accommodate varying text types

Before

```
<Text />

<H1 />

<H2 />

<H3 />
```

After

```
<Text />

<Text as=`h1`/>

<Text as=`h2`/>

<Text as=`h3`/>
```

Apart from great flexibility, this also lets your user pass in a more accessible element type depending on their specific use case e.g., **as="blockquote"**

2. Interoperability

Imagine if you built a **ListItem** component
to be reused by other engineers.

```
<ListItem />
```

Instead of tying the user to a specific render element type,
they could go ahead and their favourite open source library

```
import { Link } from 'react-router-dom';
```

```
<ListItem as={Link}/>
```

A polymorphic component NOT only renders element
types, but can also render custom components

FLEXIBILITY +

ACCESSIBILTY +

INTEROPERABILITY