Parsing Query Parameters & the Fragment

You learned how to extract route parameters (=> `:id` etc).

But how do you extract **search** (also referred to as "**query**") **parameters** (=> `?something=somevalue` at the end of the URL)? How do you extract the **fragment** (=> `#something` at the end of the URL)?

Query Params:

You can pass them easily like this:

```
<Link to="/my-path?start=5">Go to Start</Link>
```

or

```
<Link
    to={{
        pathname: '/my-path',
        search: '?start=5'
    }}
    >Go to Start</Link>
```

React router makes it easy to get access to the search string: `props.location.search` .

But that will only give you something like `?start=5`

You probably want to get the key-value pair, without the `?` and the `=`. Here's a snippet which allows you to easily extract that information:

```
componentDidMount() {
    const query = new URLSearchParams(this.props.location.search);
    for (let param of query.entries()) {
        console.log(param); // yields ['start', '5']
    }
}
```

`URLSearchParams` is a built-in object, shipping with vanilla JavaScript. It returns an object, which exposes the `entries()` method. `entries()` returns an Iterator - basically a construct which can be used in a `for...of...` loop (as shown above).

When looping through `query.entries()`, you get **arrays** where the first element is the **key name** (e.g. `start`) and the second element is the assigned **value** (e.g. `5`).

Fragment:

You can pass it easily like this:

```
<Link to="/my-path#start-position">Go to Start</Link>
```

or

```
<Link
    to={{
        pathname: '/my-path',
        hash: 'start-position'
```

```
      }}
      >Go to Start</Link>
```

React router makes it easy to extract the fragment. You can simply access `props.location.hash` .