

[› Menu](#)[Pages Router](#) > ... > [Components](#) > [<Head>](#)

<Head>

► Examples

We expose a built-in component for appending elements to the `head` of the page:

```
1  import Head from 'next/head';
2
3  function IndexPage() {
4    return (
5      <div>
6        <Head>
7          <title>My page title</title>
8        </Head>
9        <p>Hello world!</p>
10     </div>
11   );
12 }
13
14 export default IndexPage;
```

To avoid duplicate tags in your `head` you can use the `key` property, which will make sure the tag is only rendered once, as in the following example:

```
1  import Head from 'next/head';
2
3  function IndexPage() {
4    return (
5      <div>
6        <Head>
7          <title>My page title</title>
8          <meta property="og:title" content="My page title" key="title" />
9        </Head>
10       <Head>
11         <meta property="og:title" content="My new title" key="title" />
12       </Head>
13       <p>Hello world!</p>
14     </div>
15   );
16 }
```

```
17  
18 export default IndexPage;
```

In this case only the second `<meta property="og:title" />` is rendered. `meta` tags with duplicate `key` attributes are automatically handled.

The contents of `head` get cleared upon unmounting the component, so make sure each page completely defines what it needs in `head`, without making assumptions about what other pages added.

`title`, `meta` or any other elements (e.g. `script`) need to be contained as **direct** children of the `Head` element, or wrapped into maximum one level of `<React.Fragment>` or arrays—otherwise the tags won't be correctly picked up on client-side navigations.

We recommend using [next/script](#) in your component instead of manually creating a `<script>` in `next/head`.