Gatsby uses a React component to server render the `<head>` and other parts of the HTML outside of the core Gatsby application.

Most sites should use the default `html.js` shipped with Gatsby. But if you need to customize your site's html.js, copy the default one into your source tree by running:

```
cp .cache/default-html.js src/html.js
```

And then make modifications as needed.

If you need to insert custom HTML into the `<head>` or `<footer>` of each page on your site, you can use `html.js`.

> Customizing `html.js` is a workaround solution for when the use of the appropriate APIs is not available in `gatsby-ssr.js`. Consider using [onRenderBody](#) or [onPreRenderHTML](#) instead of the method above. As a further consideration, customizing `html.js` is not supported within a Gatsby Theme. Use the API methods mentioned instead.

Required props

Note: the various props that are rendered into pages *are* required e.g. `headComponents`, `preBodyComponents`, `body`, and `postBodyComponents`.

Inserting HTML into the `<head>`

Anything you render in the `html.js` component will *not* be made "live" in the client like other components. If you want to dynamically update your `<head>` we recommend using [React Helmet](#)

Inserting HTML into the `<footer>`

If you want to insert custom HTML into the footer, `html.js` is the preferred way of doing this. If you're writing a plugin, consider using the `setPostBodyComponents` prop in the [Gatsby SSR API](#).

Target container

If you see this error: `Uncaught Error: _registerComponent(...): Target container is not a DOM element.` it means your `html.js` is missing the required "target container". Inside your `<body>` you must have a div with an id of `___gatsby` like:

```
<div
  key={`body`}
  id="___gatsby"
  dangerouslySetInnerHTML={{ __html: this.props.body }}
/>
```

Adding custom JavaScript

You can add custom JavaScript to your HTML document by using React's [dangerouslySetInnerHTML](#) attribute.

```
<script
  dangerouslySetInnerHTML={{
    __html: `
          var name = 'world';
          console.log('Hello ' + name);
      `,
  }}
/>
```