

```
<NavLink
  to="/"
  className={({ isActive }) =>
    isActive ? classes.active : undefined
  }
  style={({ isActive }) => ({
    textAlign: isActive ? "center" : "left",
  })}
end
>
  Home
</NavLink>
```



explain in details whole code in step by step with example

Certainly! Let's break down the entire `<NavLink>` component step by step so you can understand each part and how it works, with examples.

This component is from **React Router**, specifically `react-router-dom`. It's used for navigation in a single-page React application (SPA).

### Full Code:

jsx

 Copy  Edit

```
<NavLink to="/" className={({ isActive }) => isActive ? classes.active : undefined }
style={({ isActive }) => ({ textAlign: isActive ? "center" : "left", })} end > Home
</NavLink>
```

## Step-by-Step Explanation:

### 1. `<NavLink>`

- `NavLink` is a special version of `Link` from `react-router-dom`.
  - It is used to render navigation links.
  - It provides styling or class features based on whether the link is currently "active" (i.e., the route matches the current URL).
- 

### 2. `to="/"`

- This defines the destination path. In this case, it's the **home route** (root).
  - Example:  
If the user clicks this link, the URL becomes `http://yourdomain.com/`.
- 



### 3. `className={({ isActive }) => isActive ? classes.active : undefined}`

- `className` can accept a function in `NavLink`.
- That function receives an object with `isActive` (a boolean).
  - `true`: the current route matches the `to` path
  - `false`: otherwise

Example:

Let's say your CSS module contains:

js

 Copy  Edit

```
const classes = { active: "nav-link-active" };
```

Then:

- If you are on `/` (home), `isActive` is `true`, so the `className` becomes `"nav-link-active"`.
- If you are on `/about`, `isActive` is `false`, so `className` is `undefined` (no class added).



4. `style={({ isActive }) => ({ textAlign: isActive ? "center" : "left" })}`

- Similar to `className`, the `style` prop can accept a function that gets `isActive`.
- It returns a style object that applies different styles based on whether the link is active.

Example:

- If current path is `/` → `isActive: true` → style becomes:



js

 Copy  Edit

```
{ textAlign: "center" }
```

- If current path is `/about` → `isActive: false` → style becomes:

js

 Copy  Edit



```
{ textAlign: "left" }
```

So the alignment of the link text will change based on whether the link is active.

## 5. `end`

- This ensures the route is matched **exactly**.
- Without `end`, paths like `/` would still be considered active when the current URL is `/about`, because `/` is a prefix.
- With `end`, `/` will only be active when the URL is exactly `/`.



Example:

- URL: `/` → link is active 
- URL: `/about` → link is not active 

## 6. `Home`

- This is the child content of the `NavLink`.
- This is what the user sees and clicks.
- Rendered like a normal link:

html

 Copy  Edit

```
<a href="/">Home</a>
```

 **Final Render Example (Assuming you're at `/`):**

DOM Output (Simplified):

html

```
<a href="/" class="nav-link-active" style="text-align: center;" > Home </a>
```



Copy



Edit



## Final Render Example (If you're at `/about`):

html



Copy



Edit

```
<a href="/" class="" style="text-align: left;" > Home </a>
```



## Summary:

Part

Purpose

`to="/"`

Navigates to home (`/`)

`className`

Applies a class if the route is active

`style`

Applies styles conditionally based on `isActive`

`end`

Ensures exact match (e.g., `/` matches only `/`, not `/about`)

`Home`

Visible link text