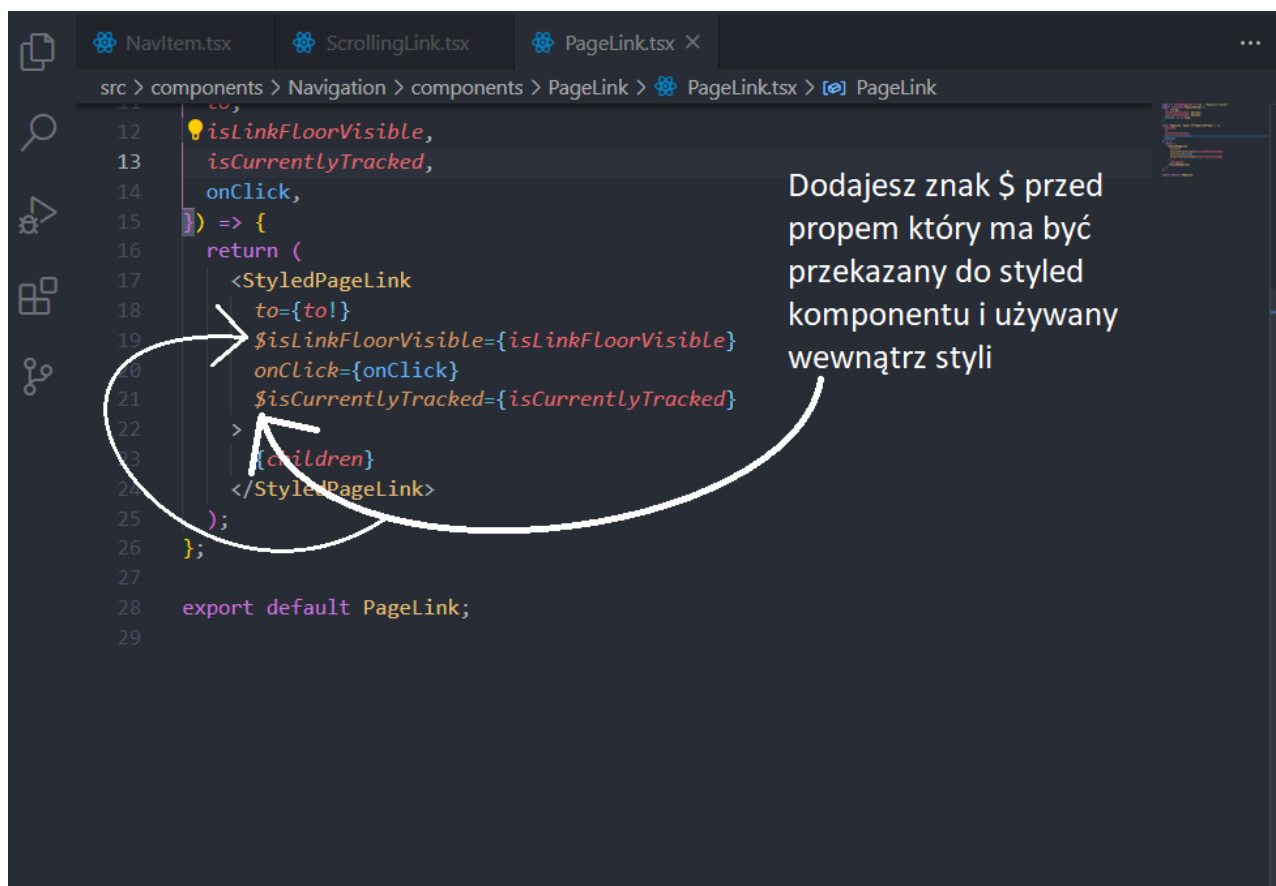


Jeśli masz błąd „React does not recognize the `isLinkFloorVisible` prop on a DOM element...”

```
Warning: React does not recognize the `isLinkFloorVisible` prop on a DOM element. If you intentionally want it to appear in the DOM as a custom attribute,
a
LinkElement@http://localhost:3000/visual-novel/static/js/0.chunk.js:40125:20
Link@http://localhost:3000/visual-novel/static/js/0.chunk.js:41405:22
Q@http://localhost:3000/visual-novel/static/js/0.chunk.js:48654:6
ScrollingLink@http://localhost:3000/visual-novel/static/js/main.chunk.js:4889:23
span
Q@http://localhost:3000/visual-novel/static/js/0.chunk.js:48654:6
li
Q@http://localhost:3000/visual-novel/static/js/0.chunk.js:48654:6
NavItem@http://localhost:3000/visual-novel/static/js/main.chunk.js:1452:17
ul
Q@http://localhost:3000/visual-novel/static/js/0.chunk.js:48654:6
NavItemsList@http://localhost:3000/visual-novel/static/js/main.chunk.js:3850:22
div
Q@http://localhost:3000/visual-novel/static/js/0.chunk.js:48654:6
ConnectFunction@http://localhost:3000/visual-novel/static/js/0.chunk.js:36639:75
div
Q@http://localhost:3000/visual-novel/static/js/0.chunk.js:48654:6
nav
Q@http://localhost:3000/visual-novel/static/js/0.chunk.js:48654:6
Navigation@http://localhost:3000/visual-novel/static/js/main.chunk.js:735:83
Route@http://localhost:3000/visual-novel/static/js/0.chunk.js:39481:29
div
Ge@http://localhost:3000/visual-novel/static/js/0.chunk.js:48565:67
Router@http://localhost:3000/visual-novel/static/js/0.chunk.js:39116:30
BrowserRouter@http://localhost:3000/visual-novel/static/js/0.chunk.js:38736:35
App
Provider@http://localhost:3000/visual-novel/static/js/0.chunk.js:36352:15
```

to znaczy, że przekazywałeś do stylowanego `<a>` czy np `<Link>` z react-router prop którego normalnie taki `<a>` nie posiada po wyrenderowaniu w HTML. Aby temu zapobiec możesz nazwę takiego propa poprzedzić znakiem `$` i potem w styled komponencie używać tej nazwy ze znakiem `$`:



```
12  isLinkFloorVisible,
13  isCurrentlyTracked,
14  onClick,
15  }) => {
16    return (
17      <StyledPageLink
18        to={to!}
19        $isLinkFloorVisible={isLinkFloorVisible}
20        onClick={onClick}
21        $isCurrentlyTracked={isCurrentlyTracked}
22      >
23        {children}
24      </StyledPageLink>
25    );
26  };
27
28  export default PageLink;
29
```

Dodajesz znak \$ przed propem który ma być przekazany do styled komponentu i używany wewnątrz styli

I używasz w następujący sposób w styled komponencie:

src > components > Navigation > components > PageLink > PageLink.styled.ts > StyledPageLink

```
1 import styled from 'styled-components';
2 import { Link, LinkProps } from 'react-router-dom';
3 import MediaQueries from 'utils/MediaQueries';
4 export interface StyledPageLinkProps extends LinkProps {
5   $isLinkFloorVisible?: boolean;
6   $isCurrentlyTracked: boolean;
7 }
8
9 export const StyledPageLink = styled(Link)<StyledPageLinkProps>`
10   text-transform: capitalize;
11   align-items: center;
12   display: flex;
13   height: 100%;
14   color: rgb(${({ theme }) => theme.lightModeColors.textPrimary});
15   position: relative;
16   overflow: hidden;
17   padding: 15px 20px;
18   text-decoration: none;
19
20   &::before {
21     content: '';
22     position: absolute;
23     left: 0;
24     bottom: 0;
25     width: 100%;
26     height: 3px;
27     border-radius: 5px;
28     background-color: rgb(
29       ${({ theme }) => theme.lightModeColors.textSecondary}
30     );
31     transition: transform 0.3s cubic-bezier(0.39, 0.99, 0.18, 1.35);
32     transform: translateX(-100%);
33
34     ${MediaQueries.laptop} {
35       transform: translateX(
36         ${({ $isLinkFloorVisible, $isCurrentlyTracked }) =>
37           $isCurrentlyTracked ? '0' : $isLinkFloorVisible ? '0' : '-100%'}
38       );
39     }
40   }
```

Link do artykułu gdzie znalazłem to rozwiązanie:

<https://stackoverflow.com/questions/49784294/warning-received-false-for-a-non-boolean-attribute-how-do-i-pass-a-boolean-f>

Screen odpowiedzi podającej takie rozwiązanie:



As of 5.1 you can now use [transient props](#) which prevents the props being passed to the DOM element.

6



```
const Comp = styled.div`
  color: ${props =>
    props.$draggable || 'black'};
`;

render(
  <Comp $draggable="red" draggable="true">
    Drag me!
  </Comp>
);
```

share improve this answer follow

answered Oct 1 '20 at 14:34



MustKillBill

303 • 3 • 14

add a comment

opisanie tego mechanizmu na oficjalnej stronie styled-components:

Transient props v5.1

If you want to prevent props meant to be consumed by styled components from being passed to the underlying React node or rendered to the DOM element, you can prefix the prop name with a dollar sign (\$), turning it into a transient prop.

In this example, `$draggable` isn't rendered to the DOM like `draggable` is.

```
const Comp = styled.div`
  color: ${props =>
    props.$draggable || 'black'};
`;

render(
  <Comp $draggable="red" draggable="true">
    Drag me!
  </Comp>
);
```

Drag me!