React的Tab切換動畫筆記

1. 看這部影片[React.js: Building an animated Tabs component](https://www.youtube.com/watch?v=KlQagQQmwX0&feature=emb_title)

看這部的過程

想要理解React.Childer.map(), componentDidMount(),

官方的文字敘述：  
如果你熟悉 React class 的生命週期方法，你可以把 useEffect 視為 componentDidMount，componentDidUpdate 和 componentWillUnmount 的組合。  
“””你不能在 function component 上使用 ref，因為他們沒有 instance”””

“””然而，只要你想要指到的是 DOM element 或 class component，你也可以在 function component 裡使用 ref”””

想要釐清brigand使用的els的用意

1. [[React] 搭配 React Router 打造一個動態麵包屑（dynamic breadcrumb）](https://pjchender.blogspot.com/2018/11/react-react-router-dynamic-breadcrumb.html)
2. 一個playground的tab底線動畫 即時檢視程式碼：<https://codepen.io/burntcustard/pen/PwGgJz>
3. [一名作者](https://riptutorial.com/reactjs/example/28223/stateless-functional-component)對stateless component優點的列舉：  
   Benefits:
   1. No class overhead
   2. Don't have to worry about this keyword
   3. Easy to write and easy to understand
   4. Don't have to worry about managing state values
   5. Performance improvement
4. 最後的解法

用functional-based去包裹著component-based,

兩者合用

才能取得每個tab的state, 同時又能使用react-router-dom package

製作TabsInnerClass.js , 是component-based, 內容幾乎就是bigrand作者的[Tabs.js](https://github.com/brigand/react-scratch/blob/master/src/tabs/Tabs.js)

製作TabsWrapperClass.js, 是functional-based,

|  |
| --- |
| const TabAnimationFunction = ({ route, location }) => {  /\*\* initialization hooks \*/  const [clickedKey, setClickedKey] = useState('00');  return (  <BrowserRouter>  <div  >  <TabsInnerClass  active={clickedKey}  onChange={active => setClickedKey(active)}  className="Tabs"  >  <div key="00"><Link to='/'>Calender</Link></div>  <div key="01"><Link to='/wardrobe'>Wardrobe</Link></div>  <div key="02"><Link to='/outfits'>Outfits</Link></div>  <div key="03"><Link to ='/setting'>Setting</Link></div>  </TabsInnerClass>  </div>  {/\*\* use renderRoutes to replace the primal map() way \*/}  {renderRoutes(myRoutes)}  </BrowserRouter>  );  } |