4.Pizza

1.Создаем компоненту SortPopup:

import React from "react";

const SortPopup=(props)=>{

return <div className="sort">

<div className="sort\_\_label">

  <svg

    width="10"

    height="6"

    viewBox="0 0 10 6"

    fill="none"

    xmlns="http://www.w3.org/2000/svg"

  >

    <path

      d="M10 5C10 5.16927 9.93815 5.31576 9.81445 5.43945C9.69075 5.56315 9.54427 5.625 9.375 5.625H0.625C0.455729 5.625 0.309245 5.56315 0.185547 5.43945C0.061849 5.31576 0 5.16927 0 5C0 4.83073 0.061849 4.68424 0.185547 4.56055L4.56055 0.185547C4.68424 0.061849 4.83073 0 5 0C5.16927 0 5.31576 0.061849 5.43945 0.185547L9.81445 4.56055C9.93815 4.68424 10 4.83073 10 5Z"

      fill="#2C2C2C"

    />

  </svg>

  <b>Сортировка по:</b>

  <span>популярности</span>

</div>

<div className="sort\_\_popup">

  <ul>

    <li className="active">популярности</li>

    <li>цене</li>

    <li>алфавиту</li>

  </ul>

</div>

</div>

}

export default SortPopup

2.Делаем компоненту интерактивной:

import React, { useState } from "react";

const SortPopupCopy=(props)=>{

    const[activeItem,setActiveItem]=useState(null)

const[visiblePopup,setVisiblePopup]=useState(false)

return <div className="sort">

<div className="sort\_\_label">

  <svg

    width="10"

    height="6"

    viewBox="0 0 10 6"

    fill="none"

    xmlns="http://www.w3.org/2000/svg"

  >

    <path

      d="M10 5C10 5.16927 9.93815 5.31576 9.81445 5.43945C9.69075 5.56315 9.54427 5.625 9.375 5.625H0.625C0.455729 5.625 0.309245 5.56315 0.185547 5.43945C0.061849 5.31576 0 5.16927 0 5C0 4.83073 0.061849 4.68424 0.185547 4.56055L4.56055 0.185547C4.68424 0.061849 4.83073 0 5 0C5.16927 0 5.31576 0.061849 5.43945 0.185547L9.81445 4.56055C9.93815 4.68424 10 4.83073 10 5Z"

      fill="#2C2C2C"

    />

  </svg>

  <b>Сортировка по:</b>

  <span onClick={()=>setVisiblePopup(!visiblePopup)}>популярности</span>

  {window.visiblePopup=visiblePopup}

</div>

{visiblePopup && <div className="sort\_\_popup">

  <ul>

     {props.varies.map((elem,index)=>{

     return <li key={`${elem}\_${index}`} onClick={()=>setActiveItem(index)}

     className={activeItem===index ? "active":""}>{elem}</li>

      })

    }

    </ul>

</div>}

</div>

}

export default SortPopupCopy

3.Более точное значение:

import React, { useState } from "react";

const SortPopupWithHooks=(props)=>{

    const[activeItem,setActiveItem]=useState(0)

const[visiblePopup,setVisiblePopup]=useState(false)

const toggleVisiblePopup=()=>{

  setVisiblePopup(!visiblePopup)

}

const chooseParam=(index)=>{

    setActiveItem(index)

    setVisiblePopup(false)

}

return <div className="sort">

<div className="sort\_\_label">

  <svg

    width="10"

    height="6"

    viewBox="0 0 10 6"

    fill="none"

    xmlns="http://www.w3.org/2000/svg"

  >

    <path

      d="M10 5C10 5.16927 9.93815 5.31576 9.81445 5.43945C9.69075 5.56315 9.54427 5.625 9.375 5.625H0.625C0.455729 5.625 0.309245 5.56315 0.185547 5.43945C0.061849 5.31576 0 5.16927 0 5C0 4.83073 0.061849 4.68424 0.185547 4.56055L4.56055 0.185547C4.68424 0.061849 4.83073 0 5 0C5.16927 0 5.31576 0.061849 5.43945 0.185547L9.81445 4.56055C9.93815 4.68424 10 4.83073 10 5Z"

      fill="#2C2C2C"

    />

  </svg>

  <b>Сортировка по:</b>

  <span onClick={toggleVisiblePopup}>{props.varies[activeItem]}</span>

  {window.visiblePopup=visiblePopup}

</div>

{visiblePopup && <div className="sort\_\_popup">

  <ul>

     {props.varies.map((elem,index)=>{

     return <li key={`${elem}\_${index}`} onClick={()=>chooseParam(index)}

     className={activeItem===index ? "active":""}>{elem}</li>

      })

    }

    </ul>

</div>}

</div>

}

export default SortPopupWithHooks

4.Желательно для оптимизации не использовать анонимные функции в параметрах компонентов,изза того ,что при ререндере реакт каждый раз будет создавать новую функцию и засорять память.

5.Просматриваем вскользь хуки useRef и useEffect.

useEffect это почти аналог классовых методов жизненного цикла.

useEffect( ()=>{ … },[зависимость])

Если массив пустой ,значит нет зависимостей и код в фигурных скобках вызовется один раз.Эмулирует метод жизненного цикла componentDidMount- код в скобках выполняется только один оаз когда компонента будет смонтирована.

Если добавить зависимость в массив ,то код в фигурных скобках будет выполняться каждый раз при изменении переменной зависимости.

useRef –хранит обьект с актуальной ссылкой.

Const sortRef=useRef(сылка)

6.Интересные методы работы с массивами includes и find-аналоги метода some.Проверяют на совпадение.

7.Новое промежуточное значение компоненты:

import React, { useEffect, useRef, useState } from "react";

const SortPopupWithHooks=(props)=>{

    const[activeItem,setActiveItem]=useState(0)

const[visiblePopup,setVisiblePopup]=useState(false)

---------------------------------------------------------------------------------

const sortRef=useRef(null)

---------------------------------------------------------------------------------

const func=(event)=>{

  console.log(event)

  !event.path.some(elem=>elem===sortRef.current) &&  setVisiblePopup(false)

}

useEffect((e)=>{

document.body.addEventListener("click",func)},

[]

)

----------------------------------------------------------------------------------

const toggleVisiblePopup=()=>{

  setVisiblePopup(!visiblePopup)

}

const chooseParam=(index)=>{

    setActiveItem(index)

    setVisiblePopup(false)

}

---------------------------------------------------------------------------------

return <div ref ={sortRef} className="sort">

---------------------------------------------------------------------------------

<div className="sort\_\_label">

  <svg

    width="10"

    height="6"

    viewBox="0 0 10 6"

    fill="none"

    xmlns="http://www.w3.org/2000/svg"

  >

    <path

      d="M10 5C10 5.16927 9.93815 5.31576 9.81445 5.43945C9.69075 5.56315 9.54427 5.625 9.375 5.625H0.625C0.455729 5.625 0.309245 5.56315 0.185547 5.43945C0.061849 5.31576 0 5.16927 0 5C0 4.83073 0.061849 4.68424 0.185547 4.56055L4.56055 0.185547C4.68424 0.061849 4.83073 0 5 0C5.16927 0 5.31576 0.061849 5.43945 0.185547L9.81445 4.56055C9.93815 4.68424 10 4.83073 10 5Z"

      fill="#2C2C2C"

    />

  </svg>

  <b>Сортировка по:</b>

  <span onClick={toggleVisiblePopup}>{props.varies[activeItem]}</span>

  {window.visiblePopup=visiblePopup}

</div>

{visiblePopup && <div className="sort\_\_popup">

  <ul>

     {props.varies.map((elem,index)=>{

     return <li key={`${elem}\_${index}`} onClick={()=>chooseParam(index)}

     className={activeItem===index ? "active":""}>{elem}</li>

      })

    }

    </ul>

</div>}

</div>

}

export default SortPopupWithHooks

8.Использовали три хука:useState,useEffect,useRef:

   const[activeItem,setActiveItem]=useState(0)

const[visiblePopup,setVisiblePopup]=useState(false)

useEffect((e)=>{

document.body.addEventListener("click",func)},

[]

)

const sortRef=useRef(null)

9.Наша логика связанная с эвентом:

const sortRef=useRef(null)

const func=(event)=>{

  !event.path.some(elem=>elem===sortRef.current) &&  setVisiblePopup(false)

}

useEffect((e)=>{

document.body.addEventListener("click",func)},

[]

)

<div ref ={sortRef} className="sort"> в разметке