### Совместное использование состояний и ссылки

### Цель работы

Познакомить со способами использования ссылок на значения React компонент, а также попрактиковаться в совместном использовании состояний между компонентами.

### Задания для выполнения

1. Организуйте таймер на странице с использованием ссылок.
2. Реализуйте слайдер, используя ссылки для манипуляции с DOM.
3. Реализуйте управление состояниями в приложении React, так чтобы один компонент зависел от состояния другого компонента.
4. Сделать свое СМИ для программистов с новостными слайдерами, таймером отдыха и возможностью просмотра разных статей за счет связанных и несвязных состояний.



1. Загрузить созданную страницу на GitHub в репозиторий Student, используя формат в названии Фамилия (латинскими буквами)\_8.

### Методические указания

1. Вы можете объединить ссылки и состояние в одном компоненте. Например, давайте создадим секундомер, который пользователь может запустить или остановить нажатием кнопки. Чтобы отобразить, сколько времени прошло с тех пор, как пользователь нажал “Пуск”, вам нужно будет отслеживать, когда была нажата кнопка "Пуск" и каково текущее время.

import { useState, useRef } from 'react';

export default function Stopwatch() {

 const [startTime, setStartTime] = useState(null);

 const [now, setNow] = useState(null);

 const intervalRef = useRef(null);

 function handleStart() {

 setStartTime(Date.now());

 setNow(Date.now());

 clearInterval(intervalRef.current);

 intervalRef.current = setInterval(() => {

 setNow(Date.now());

 }, 10);

 }

 function handleStop() {

 clearInterval(intervalRef.current);

 }

 let secondsPassed = 0;

 if (startTime != null && now != null) {

 secondsPassed = (now - startTime) / 1000;

 }

 return (

 <>

 <h1>Time passed: {secondsPassed.toFixed(3)}</h1>

 <button onClick={handleStart}>

 Start

 </button>

 <button onClick={handleStop}>

 Stop

 </button>

 </>

 );

}

1. React автоматически обновляет [DOM](https://developer.mozilla.org/docs/Web/API/Document_Object_Model/Introduction) в соответствии с вашими результатами рендеринга, поэтому вашим компонентам не придется часто манипулировать им. Однако иногда вам может потребоваться доступ к элементам DOM, управляемым React, — например, для фокусировки узла, прокрутки до него или измерения его размера и положения. В React нет встроенного способа делать эти вещи, поэтому вам понадобится *ссылка* на узел DOM.

import { useRef } from 'react';

export default function CatFriends() {

 const firstCatRef = useRef(null);

 const secondCatRef = useRef(null);

 const thirdCatRef = useRef(null);

 function handleScrollToFirstCat() {

 firstCatRef.current.scrollIntoView({

 behavior: 'smooth',

 block: 'nearest',

 inline: 'center'

 });

 }

 function handleScrollToSecondCat() {

 secondCatRef.current.scrollIntoView({

 behavior: 'smooth',

 block: 'nearest',

 inline: 'center'

 });

 }

 function handleScrollToThirdCat() {

 thirdCatRef.current.scrollIntoView({

 behavior: 'smooth',

 block: 'nearest',

 inline: 'center'

 });

 }

 return (

 <>

 <nav>

 <button onClick={handleScrollToFirstCat}>

 Tom

 </button>

 <button onClick={handleScrollToSecondCat}>

 Maru

 </button>

 <button onClick={handleScrollToThirdCat}>

 Jellylorum

 </button>

 </nav>

 <div>

 <ul>

 <li>

 <img

 src="https://placekitten.com/g/200/200"

 alt="Tom"

 ref={firstCatRef}

 />

 </li>

 <li>

 <img

 src="https://placekitten.com/g/300/200"

 alt="Maru"

 ref={secondCatRef}

 />

 </li>

 <li>

 <img

 src="https://placekitten.com/g/250/200"

 alt="Jellylorum"

 ref={thirdCatRef}

 />

 </li>

 </ul>

 </div>

 </>

 );

}

1. Для организации связанных стейтов нужно убрать State из родительского компонента и передавать State как Proops в каждый дочерний компонент:

import { useState } from 'react';

export default function Accordion() {

 const [activeIndex, setActiveIndex] = useState(0);

 return (

 <>

 <h2>Almaty, Kazakhstan</h2>

 <Panel

 title="About"

 isActive={activeIndex === 0}

 onShow={() => setActiveIndex(0)}

 >

 With a population of about 2 million, Almaty is Kazakhstan's largest city. From 1929 to 1997, it was its capital city.

 </Panel>

 <Panel

 title="Etymology"

 isActive={activeIndex === 1}

 onShow={() => setActiveIndex(1)}

 >

 The name comes from <span lang="kk-KZ">алма</span>, the Kazakh word for "apple" and is often translated as "full of apples". In fact, the region surrounding Almaty is thought to be the ancestral home of the apple, and the wild <i lang="la">Malus sieversii</i> is considered a likely candidate for the ancestor of the modern domestic apple.

 </Panel>

 </>

 );

}

function Panel({

 title,

 children,

 isActive,

 onShow

}) {

 return (

 <section className="panel">

 <h3>{title}</h3>

 {isActive ? (

 <p>{children}</p>

 ) : (

 <button onClick={onShow}>

 Show

 </button>

 )}

 </section>

 );

}

### Контрольные вопросы

1. Что делает функция **componentWillUnmount()**?
2. Вызовется ли **componentDidUpdate(prevProps, prevState)** если shouldComponentUpdate возвращает false?

###

### Полезные ссылки

1. https://react.dev/learn/referencing-values-with-refs - использование ссылок
2. https://react.dev/learn/manipulating-the-dom-with-refs - манипулирование DOM с помощью ссылок
3. https://react.dev/learn/sharing-state-between-components - совместное использование состояния между компонентами