**1. Типизация события изменения (onChange)**

Для обработки событий изменения (например, события изменения текста в поле ввода input), используется тип React.ChangeEvent:

import React, { ChangeEvent } from 'react';

function MyComponent() {

const handleChange = (event: ChangeEvent<HTMLInputElement>): void => {

console.log(event.target.value); // доступ к значению input

};

Здесь HTMLInputElement указывает, что событие связано с элементом `. Если вы используете другие HTML-элементы (например, textarea), то тип значения в дженерике () нужно изменить на HTMLTextAreaElement`.

**2. Типизация события отправки формы (onSubmit)**

Для обработки событий отправки формы используется тип React.FormEvent:

import React, { FormEvent } from 'react';

function MyComponent() {

const handleSubmit = (event: FormEvent<HTMLFormElement>): void => {

event.preventDefault(); // предотвращаем обновление страницы

console.log('Form submitted');

};

Здесь HTMLFormElement указывает, что событие связано с элементом ``.

**3. Типизация событий мыши (onClick, onMouseEnter, и т.д.)**

Для событий мыши, таких как клики или перемещения курсора, используется тип React.MouseEvent:

import React, { MouseEvent } from 'react';

function MyComponent() {

const handleClick = (event: MouseEvent<HTMLButtonElement>): void => {

console.log('Button clicked');

};

Здесь HTMLButtonElement указывает, что событие связано с элементом ``.

**4. Типизация событий клавиатуры (onKeyDown, onKeyUp)**

Для событий клавиатуры, таких как нажатия клавиш, используется тип React.KeyboardEvent:

import React, { KeyboardEvent } from 'react';

function MyComponent() {

const handleKeyDown = (event: KeyboardEvent<HTMLInputElement>): void => {

if (event.key === 'Enter') {

console.log('Enter key pressed');

}

event.key содержит символ клавиши, нажатой пользователем.

**5. Типизация событий фокуса (onFocus, onBlur)**

Для событий, связанных с потерей или получением фокуса, используется React.FocusEvent:

import React, { FocusEvent } from 'react';

function MyComponent() {

const handleFocus = (event: FocusEvent<HTMLInputElement>): void => {

console.log('Input focused');

};

**6. Типизация произвольного события (универсальный React.SyntheticEvent)**

Если конкретный тип события неизвестен или событие не связано с общепринятым действием, можно использовать тип React.SyntheticEvent:

import React, { SyntheticEvent } from 'react';

function MyComponent() {

const handleEvent = (event: SyntheticEvent<HTMLDivElement>): void => {

console.log('Synthetic event triggered');

};

| Событие | Тип событий | Пример HTML-элемента |

|-------------------|------------------------|----------------------------|

| onChange | React.ChangeEvent | HTMLInputElement, HTMLTextAreaElement |

| onSubmit | React.FormEvent | HTMLFormElement |

| onClick | React.MouseEvent | HTMLButtonElement, HTMLDivElement |

| onKeyDown/onKeyUp | React.KeyboardEvent | HTMLInputElement |

| onFocus/onBlur| React.FocusEvent | HTMLInputElement |

| Универсальные | React.SyntheticEvent | Любой элемент |