



React JS

React - Φόρμες & Χειρισμός Συμβάντων DOM

Δαμιανός Μαρινάτος



Controlled Inputs (1)

React

- Η τιμή (value) του input **αποθηκεύεται στο React state** μέσω του useState Hook.
- Το input είναι **πάντα συγχρονισμένο με την React** (value + onChange).
- Επιτρέπει **εύκολη επικύρωση (validation)**.
- Υποστηρίζει **δυναμικές αλλαγές περιεχομένου**.
- Κάνει πιο εύκολο το **conditional rendering** ανάλογα με την τιμή.



Controlled Inputs (2)

React

```
import { useState } from "react";

const ControlledInput : () => Element = () : Element => { Show usages new *
  const [name, setName] = useState( initialState: "" );

  const handleChange : (e: ChangeEvent<HTMLInputElement>) => voi... = (e: React.ChangeEvent<HTMLInputElement>) : void => {
    setName(e.target.value);
  };

  return (
    <>
      <div className="text-center">
        <input
          type="text"
          value={name}
          onChange={handleChange}
          className="border rounded px-4 py-2"
        />
        <p className="mt-2">Value: {name}</p>
      </div>
    </>
  );
};

export default ControlledInput; Show usages new *
```



Controlled Inputs (3)

- **const [name, setName] = useState("");**
Δημιουργούμε ένα state μεταβλητή name που κρατάει την τιμή του input. Το setName αλλάζει την τιμή του name.
- **onChange={handleChange}**
Κάθε φορά που ο χρήστης πληκτρολογεί στο input, τρέχει η handleChange. Η handleChange παίρνει την τιμή από το event (e.target.value) και την αποθηκεύει στο name.
- **value={name}**
Η τιμή του input είναι **πάντα** συγχρονισμένη με τη React. Αν αλλάξει το name, το input ενημερώνεται.
- **<p>Value: {name}</p>**
Κάθε φορά που αλλάζει το name, το component κάνει re-render και δείχνει το νέο κείμενο.



Uncontrolled Inputs (1)

React

- Η τιμή του input **δεν αποθηκεύεται στο React state.**
- Η React **δεν παρακολουθεί τις αλλαγές** στο input.
- Χρησιμοποιούμε το **ref** για να προσπελάσουμε την τιμή του input.
- Δεν υποστηρίζουν εύκολα **επικύρωση ή δυναμικό έλεγχο.**
- Είναι πιο κοντά στη "παραδοσιακή" HTML συμπεριφορά.



Uncontrolled Inputs (2)

React

```
import { useRef } from "react";

const UncontrolledInput : () => Element = () : Element => { Show usages new *
  const inputRef : RefObject<HTMLInputElement | null> = useRef<HTMLInputElement>( initialValue: null);

  const handleClick : () => void = () : void => { Show usages new *
    alert( message: `Value: ${inputRef.current?.value}`);
  };

  return (
    <div>
      <input type="text" ref={inputRef} className="border rounded px-4 py-2" />
      <button
        onClick={handleClick}
        className="ml-2 bg-cf-dark-red text-white px-3 py-1 rounded"
      >
        Show Value
      </button>
      <p className="mt-2">Value : {inputRef.current?.value}</p>
    </div>
  );
};
```



Uncontrolled Inputs (3)

React

- **const inputRef = useRef<HTMLInputElement>(null);**
Χρησιμοποιούμε το useRef για να δημιουργήσουμε ένα reference στο input element. Η αρχική τιμή του current είναι null, αλλά μόλις γίνει render το στοιχείο, το inputRef.current θα δείχνει στο actual DOM element.
- **ref={inputRef}**
Συνδέει το input με το inputRef, ώστε να μπορούμε να έχουμε πρόσβαση στο DOM στοιχείο.
- **onClick={handleClick}**
Όταν ο χρήστης πατήσει το κουμπί, τρέχει η handleClick\
- **handleClick = () => { alert(...); }**
Η handleClick παίρνει την **τρέχουσα τιμή** του input από το DOM μέσω inputRef.current?.value και εμφανίζει ένα alert.



Controlled vs Uncontrolled

React

Χαρακτηριστικό	Controlled Input	Uncontrolled Input
Διαχείριση Τιμής	Μέσω useState	Μέσω ref και του DOM
Αντίδραση σε αλλαγές	Προκαλεί re-render	Δεν προκαλεί re-render
Ανάγνωση τιμής	Από το state (value)	Από το DOM (ref.current?.value)
Ιδανικό για	Πολύπλοκες φόρμες, validation, preview	Απλές ή legacy φόρμες
Δυνατότητα validation	Ναι	Περιορισμένη



HTML Element Τύποι

React

HTML Στοιχείο	Τύπος TypeScript	Παράδειγμα HTML
<code><input></code>	<code>HTMLInputElement</code>	<code><input type="text" /></code>
<code><textarea></code>	<code>HTMLTextAreaElement</code>	<code><textarea></textarea></code>
<code><select></code>	<code>HTMLSelectElement</code>	<code><select>...</select></code>
<code><option></code>	<code>HTMLOptionElement</code>	<code><option>Option</option></code>
<code><form></code>	<code>HTMLFormElement</code>	<code><form>...</form></code>
<code><button></code>	<code>HTMLButtonElement</code>	<code><button>Click</button></code>
<code><output></code>	<code>HTMLOutputElement</code>	<code><output></output></code>



Form & Input Events

React

Τύπος Event	React TypeScript Τύπος	Περιγραφή
onChange	React.ChangeEvent<T>	Όταν αλλάζει η τιμή ενός input
onInput	React.FormEvent<T>	Όταν εισάγεται κείμενο
onSubmit	React.FormEvent<HTMLFormElement>	Υποβολή φόρμας
onReset	React.FormEvent<HTMLFormElement>	Reset φόρμας
onInvalid	React.InvalidEvent<T>	Όταν input δεν περνά validation
onSelect	React.SyntheticEvent<T>	Όταν γίνεται επιλογή κειμένου

*T = τύπος στοιχείου (π.χ. HTMLInputElement, HTMLDivElement, HTMLTextAreaElement)



Mouse Events

React

Τύπος Event	React TypeScript Τύπος	Περιγραφή
onClick	React.MouseEvent<T>	Κλικ
onDoubleClick	React.MouseEvent<T>	Διπλό κλικ
onMouseDown	React.MouseEvent<T>	Πάτημα κουμπιού ποντικιού
onMouseUp	React.MouseEvent<T>	Απελευθέρωση κουμπιού
onmouseenter	React.MouseEvent<T>	Όταν περνάει το ποντίκι
onmouseleave	React.MouseEvent<T>	Όταν φεύγει το ποντίκι
onMouseMove	React.MouseEvent<T>	Όταν κινείται το ποντίκι



Keyboard Events

React

Τύπος Event	React TypeScript Τύπος	Περιγραφή
onKeyDown	React.KeyboardEvent<T>	Όταν πατηθεί κουμπί
onKeyPress	React.KeyboardEvent<T> (<i>deprecated</i>)	Παλαιότερο API για typing
onKeyUp	React.KeyboardEvent<T>	Όταν απελευθερωθεί κουμπί



Focus, Blur & Clipboard Events

React

Τύπος Event	React TypeScript Τύπος	Περιγραφή
onFocus	React.FocusEvent<T>	Όταν ένα πεδίο αποκτήσει focus
onBlur	React.FocusEvent<T>	Όταν ένα πεδίο χάσει focus
onCopy	React.ClipboardEvent<T>	Όταν αντιγράφεις
onCut	React.ClipboardEvent<T>	Όταν κόβεις
onPaste	React.ClipboardEvent<T>	Όταν κάνεις επικόλληση



Drag & Drop Events

React

Τύπος Event	React TypeScript Τύπος	Περιγραφή
onDrag	React.DragEvent<T>	Όταν ξεκινά drag
onDragStart	React.DragEvent<T>	Ξεκινά drag
onDragEnd	React.DragEvent<T>	Τελειώνει drag
onDragEnter	React.DragEvent<T>	Μπαίνει σε περιοχή drop
onDragLeave	React.DragEvent<T>	Φεύγει από περιοχή drop
onDragOver	React.DragEvent<T>	Περνάει πάνω από περιοχή drop
onDrop	React.DragEvent<T>	Τοποθέτηση αντικειμένου



Misc Events

React

Τύπος Event	React TypeScript Τύπος	Περιγραφή
onScroll	React.UIEvent<T>	Scroll
onWheel	React.WheelEvent<T>	Τροχός ποντικιού
onTouchStart	React.TouchEvent<T>	Touch σε κινητά
onTouchEnd	React.TouchEvent<T>	Τέλος touch
onTouchMove	React.TouchEvent<T>	Κίνηση touch