

# Attività per Pasqua

## Ripasso Teoria

### Components

I componenti sono i mattoni fondamentali di un'app React. Possono essere funzionali o basati su classi e permettono di suddividere l'interfaccia utente in parti riutilizzabili e indipendenti.

- Documentazione: <https://reactjs.org/docs/components-and-props.html> (<https://reactjs.org/docs/components-and-props.html>).

### Props

Le props (proprietà) sono il modo in cui si passano dati da un componente genitore a un componente figlio. Sono simili agli argomenti delle funzioni e sono **sola lettura**.

- Documentazione: <https://reactjs.org/docs/components-and-props.html> (<https://reactjs.org/docs/components-and-props.html>).

### useState

`useState` è un Hook che permette di gestire lo stato nei componenti funzionali. Restituisce una coppia: lo stato attuale e una funzione per aggiornarlo.

- Documentazione: <https://reactjs.org/docs/hooks-state.html> (<https://reactjs.org/docs/hooks-state.html>).

### useEffect

`useEffect` è un Hook che consente di eseguire effetti collaterali nei componenti, come fetch di dati, intervalli, o aggiornamenti manuali del DOM.

- Documentazione: <https://reactjs.org/docs/hooks-effect.html> (<https://reactjs.org/docs/hooks-effect.html>).

### map

`map()` è un metodo degli array in JavaScript che consente di trasformare ogni elemento in un nuovo elemento. In React viene usato per generare dinamicamente liste di elementi JSX.

- Documentazione: [https://developer.mozilla.org/it/docs/Web/JavaScript/Reference/Global\\_Objects/Array/map](https://developer.mozilla.org/it/docs/Web/JavaScript/Reference/Global_Objects/Array/map) ([https://developer.mozilla.org/it/docs/Web/JavaScript/Reference/Global\\_Objects/Array/map](https://developer.mozilla.org/it/docs/Web/JavaScript/Reference/Global_Objects/Array/map)).

### Controlled Form

I form controllati in React sono quelli in cui il valore degli input è gestito tramite lo stato del componente. Ogni modifica dell'input aggiorna lo stato e viceversa.

□ Documentazione: <https://reactjs.org/docs/forms.html> (<https://reactjs.org/docs/forms.html>)

---

# Esercizi

## Esercizio 1: Contatore click

**Argomenti:** Components, useState

Crea un componente `ClickCounter` che visualizzi un numero e un bottone. Al click sul bottone, il numero deve aumentare di 1.

## Esercizio 2: Lista di nomi

**Argomenti:** map, props

Crea un componente `NameList` che riceva via props un array di nomi e visualizzi ciascun nome come elemento di una lista.

## Esercizio 3: Form controllato per messaggi

**Argomenti:** Controlled Form, useState

Crea un componente con un campo di input e un bottone. Quando l'utente scrive e preme il bottone, il testo deve essere aggiunto sotto come elenco.

## Esercizio 4: Fetch utenti da API

**Argomenti:** useEffect, map

Crea un componente che faccia fetch da `https://jsonplaceholder.typicode.com/users` al montaggio e mostri i nomi in una lista.

## Esercizio 5: Componente Timer

**Argomenti:** useState, useEffect

Crea un componente che mostri un timer che aumenta di 1 ogni secondo. Aggiungi due bottoni per fermare e riprendere il timer.

---

## Esercizio Extra: Mini rubrica

**Argomenti:** components, props, useState, useEffect, map, controlled form

Crea un'app con due componenti:

- Un `ContactForm` che permetta di aggiungere nome e numero di telefono, usando un form controllato.

- Una `ContactList` che visualizzi i contatti sotto forma di lista.

I contatti devono essere salvati nello stato. Bonus: salva i contatti nel `localStorage` usando `useEffect` per mantenere i dati dopo il refresh.