# 04-01pt 2 FORMS

I form html sono inconsistenti → sono diversi in base all’elemento di input e il browser.   
React semplifica il tutto cercando di unifromare il comportamento attraverso JSX.

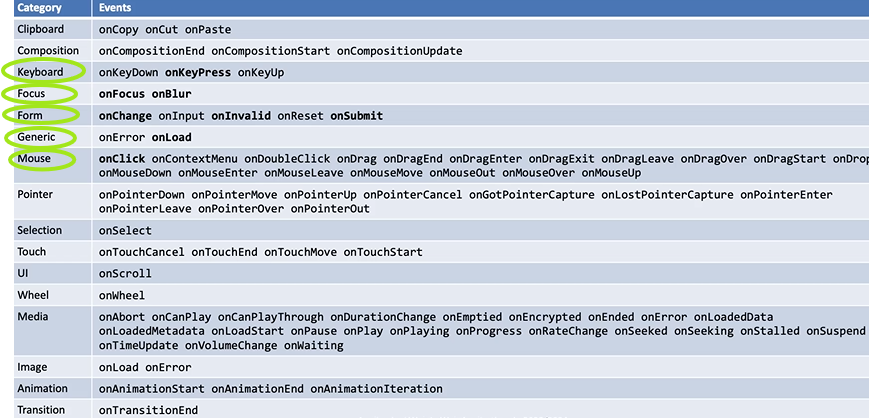
Per **textArea e select** (menù a tendina):

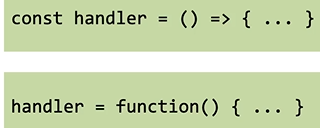
* Attributo **value** contiene sempre il valore corrente del campo
* **Default value** contiene il valore di default che è stato dato al campo quando è stato creato

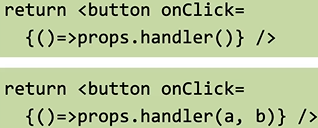
Per gestire gli eventi che cambiano, react fornisce l’elemento **onChange** che gestisce tutto ciò che cambia quando un dato cambia. Passando una funzione ad un onChange si possono gestire gli eventi che cambiano

Le callback sono event handler e sono chiamate con un parametro event object avente un insieme di proprietà:

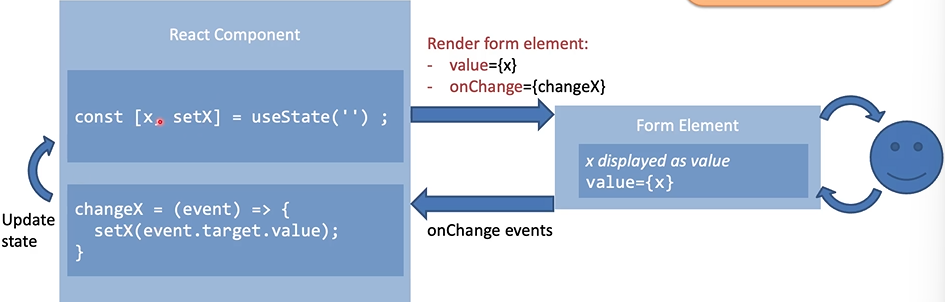
* event.target → sorgente di evento
* alcuni eventi oltre al targer possono avere alcune proprietà aggiuntive.

**Eventi sintetici:**

**Event handler**: sono funzioni, tipicamente definite come arrow function

* Quando si vuole passare un event handler, si passa il riferimento alla funzione *-senza parentesi tonde, altrimenti la stai eseguendo-*.
* Se si vogliono passare dei parametri, si può usare un arrow function

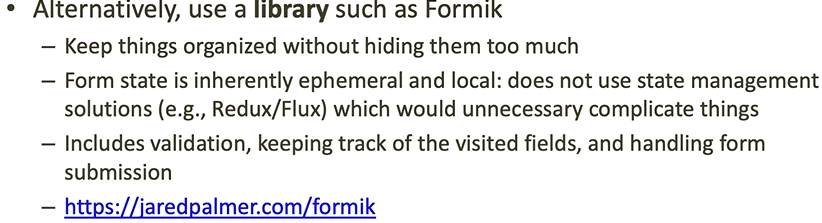
Gli **elementi di un form hanno sempre uno stato**, ogni elemento di uno stato ha bisogno di avere il proprio **onChange**.



**onSubmit** → cosa capita quando l’intero form viene mandato.

**Nota**: un tipo/elemento di type submit scatena in automatico l’evento di onClick e l’evento di submit del form → sul form possiamo usare onSubmit che reagisce al submit (senza che sia necessario reagire all’onClick)  
**Nota**: bisogna sempre chiamare event.preventDefault() per evitare la sottomissione e il ricaricamento della pagina da 0.

Nei form, i dati devono essere validati, usando i validatori interni o usando validator.js.

 → libreria aggiuntiva per form complessi

**Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteGestire array nello stato → ultime slide**

→ aggiornare elmeenmti in un array di oggetti:

crea un nuovo stato quado va a modificarsi lo stato e ogni nuovo oggetto all’interno dell’array dello stato. Ci serve fare una copia profonda dello stato (array che contiene lo stato + oggetti contenuti in esso)

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

→ aggiungere ad uno stato

Immagine che contiene testo, schermata, Carattere, documento

Descrizione generata automaticamente

→ rimuovere elementi in uno stato contenente un array