# 04-05 Context

L’idea è quella di avere una variabile che possa essere utilizzata da tutta l’applicazione.

* Impostare tema chiaro/scuro
* Infomrazioni di contesto (utente loggato o meno)
* Lingua in applicazione multi-lingua

Tree componenti principali:

* **Definizione del contesto**
  + Definisce un context object e lo salva nel riferimento ExContect
    - Crea un nuovo oggetto context, contenente due proprietà:
      * Provider
      * Consumer
  + I componenti possono iscrivere (consumare) il contesto
* **Provider del contesto**
  + Fornisce il contesto a tutti i componenti
  + Contiene props: **Value**
    - Che diventa disponibile a tutti i componenti annidati
* **Consumer del contesto**
  + Tramite componente
  + Tramite hook

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

**Language** → stato che da valore al contesto

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

*Se la props value cambia, tutti i consumer vengono ri-renderizzati*

### **Context Provider:**

il componente che fornisce il contesto deve essere al di fuori dei componenti che vuole aggiornare:

* **Value** **prop**: disponibile per ogni consumer
* *Immagine che contiene testo, Carattere, software, schermata

  Descrizione generata automaticamenteQuando una props value del provider cambia, tutti i consumer vengono aggiornati*

### Immagine che contiene testo, schermata, Carattere, linea Descrizione generata automaticamente**Context Consumer:**

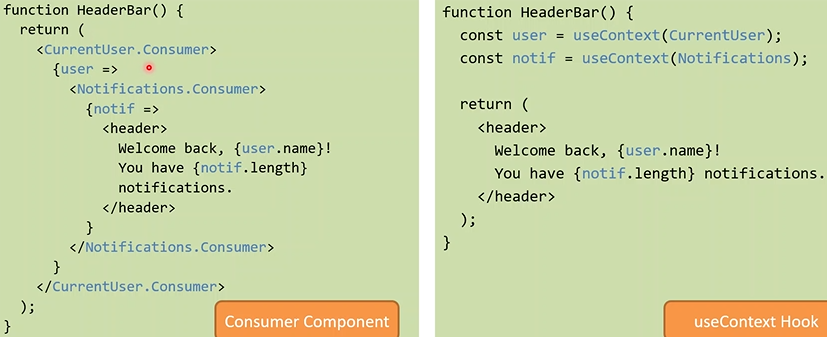
Necessita di un callback, restituisce l’elemento react da ri-renderizzare

**Hook** mi permette di accedere al valore

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

#### Definire contesti multipli:



quella di dx,   
  
migliore

Per cambiare il valore di contesto, si può fare con:

* Props drilling
* Metodo per aggiornarlo creato insieme al contesto

*Nota: Lo stato dove di solito salviamo i valore di contesto è parte del componente provider*

***Nota:*** *lo stato in cui salviamo solitamente il contresto, è nel provider*

#### **Caveats**

* Non abusare del contesto
  + Componente diventa strettamente dipendente dal contesto
  + Perdo portabilità
* Non usare per pigrizia
  + In alcuni casi serve passare una props a 10 componenti ma il contesto non è scelta saggia
* Non usare il contesto per correggere errori di design