

Flutter e Dart - Le basi

Flutter - Le basi

Luigi Durso

`luigi.durso@si2001.it`



SI2001

21 giugno 2022



Sommario

- 1 Widget: Cosa sono
- 2 Widget: Tipi

- 3 Stato dell'applicazione
- 4 Fine



E' tutta una questione di widgets

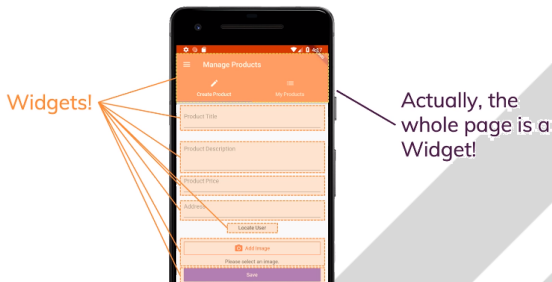


Figura: Esempio widgets



Cos'è un widget?

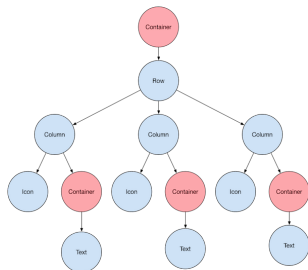


Figura: Esempio albero [1]

L'idea dietro alla composizione della UI in Flutter è la stessa di **React**, tutto ciò che compone una schermata è un widget! Quando lo stato di un widget cambia, il widget e tutti i suoi discendenti sono ridisegnati.



Esistono differenti tipo di widgets

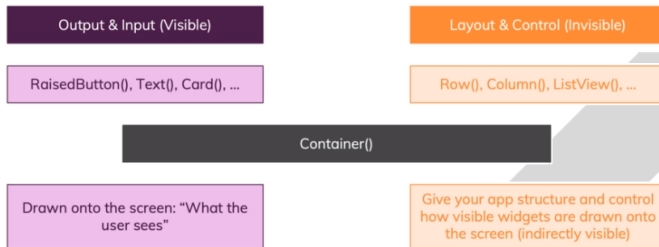


Figura: Tipi di widgets



Diamo una prima occhiata

Alcuni esempi di layout per tipo:

- Visibili: Text, Icon... [► Documentazione](#)
- Invisibili: Column, Row... [► Documentazione](#)

Esiste un widget per ogni necessità

[► Catalogo](#)



Un po' di pratica



Creiamo il nostro primo progetto



Come gestiamo lo stato

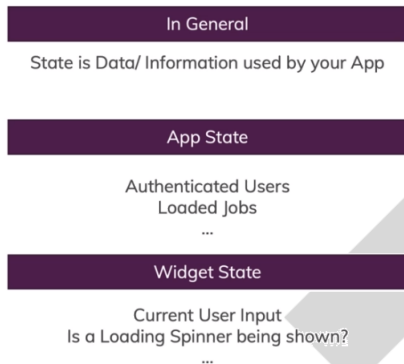


Figura: Stato di un'applicazione



Rendering dello stato

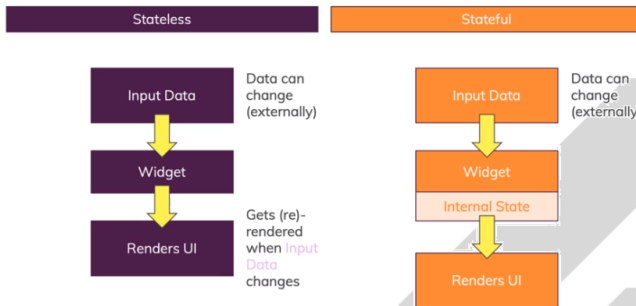


Figura: Statefull-Stateless widgets



Ciclo di vita dei widget

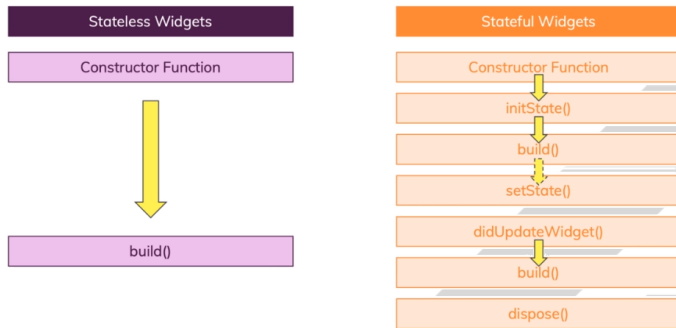


Figura: Ciclo di vita dei widget



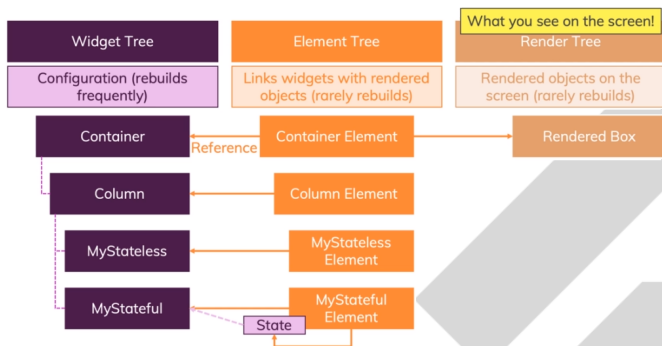
Ciclo di vita di un' app

Lifecycle State Name	When is it hit?
inactive	App is inactive, no user input received
paused	App not visible to user, running in background
resumed	App is (again) visible and responding to user input
suspending	App is about to be suspended (exited)

Figura: Ciclo di vita dell'app



Diversi livelli di gestione



Un po' di pratica



Ritocchiamo un po' progetto.



Idee di ottimizzazione



Dart

Scrivere del codice ottimizzato potrebbe fare la differenza:

- Utilizzare dei widget **const** quando possibile (riduzione delle rebuild)
- Estrazione di componenti in widget separati



Riferimenti I



flutter.dev. **Flutter layout.** 2022.



Grazie per l'attenzione!

