Inherit Widget!

1. UpdateShouldNotify - он говорит обновились ли данные в дочерних виджетах или нет, сравнивая value (значение) с oldwidget.value и возвращая true если изменилось или false если не изменилось.

Методы контекста инхерита:

1) dependOnInheritElement - подписывается на значения инхерита.

2) dependOnInheritWidgetOfExactType - получает инхерит и одновременно подписывается на него.

3) getElementForInheritWidgetOfExactType - получает инхерит виджета, из него можно получить сам инхерит без подписи на изменения.

Provider (библиотека)

1. ChangeNotifierProvider

У тебя должна быть класс (модель) наследуемая от changeNotifier с данными которые ты хочешь изменить и отобразить у себя на экране, далее в главном экране (родительском) ты должен обратиться с билда стрелочной функцией к ChangeNotifierProvider, первый аргумент у ChangeNotifierProvider это create (снова замыкание) в него передается контекст и от сюда надо передать нашу модель, далее нужно передать наш дочерний класс (child). Далее чтоб использовать наш провайдер в дочерних классах, надо у них создать переменную model (к примеру) и передать туда context.Функция<Наша модель>.функцияНашейМодели;

Пример использования в родительском классе:

Widget build(BuildContext context) => ChangeNotifierProvider(

create: (context) => Model(),

child: const \_View()

);

Пример с получением данных в дочерних классах:

final model = context.read<Model>();

Пример с получением и отображением данных:

final model = context.watch<Model>();

у ChangeNotifierProvider по мимо create и child есть еще lazy, автоматический он стоит true, это значит что модель в create он создаст не сразу, а только когда ты вызовешь ее в дочерних классах, но можно поставить и false это значит что он создастся сразу же, при первом его написании в create. Эта штука стоит в false автоматический для оптимизации, вдруг произойдет так что к этой модели в приложении пользователь обращаться не будет, тогда зачем нам ее создавать? Поэтому лучше не менять стандартное значение и оставлять false.