

Providing non-Objects

Typically, when working with the Provider package, you provide objects based on your own custom classes.

This makes sense because you can implement the `ChangeNotifier` mixin in your classes to then trigger `notifyListeners()` whenever you want to update all places in your app that listen to your data.

But you're not limited to providing objects - you can provide ANY kind of value (lists, numbers, strings, objects without `ChangeNotifier` mixing, ...).

Example:

1. `Provider<String>(builder: (ctx) => 'Hi, I am a text!', child: ...);`

Of course, if you're using Provider Package v4 or greater, it would be `create: ...` instead of `builder: ...`.

You might wonder, how this text can change though - it's a constant text after all. It certainly doesn't implement the `ChangeNotifier` mixin (the `String` class, which is built-into Dart, indeed doesn't - just like numbers, booleans etc.).

It's important to note, that the above snippet uses `Provider`, NOT `ChangeNotifierProvider`. The latter indeed only works with objects based on classes that use the `ChangeNotifier` mixin. And this is the most common use-case, because you typically want to be your global data changeable (and have the app UI react to that).

But in case you just want to provide some global (constant) value which you can then conveniently use like this:

1. `print(Provider.of<String>(context));` // prints 'Hi, I am a text!'; does never update!

you can do that.