SharedPreferences

SharedPreferences

SharedPreferences

- ***SharedPreferences*** é um recurso de armazenamento de dados bem simples e muito útil quando precisamos salvar, em persistência local, dados primitivos (**long**, **int**, **short**, **char**, **double**, **float**, **boolean**, **String**) que não sejam muito dinâmicos.

SharedPreferences

- O recurso ***SharedPreferences*** consiste em uma interface que permite acessar e modificar dados de preferência de usuário.
- O valor armazenado apresenta-se sob formato chave-valor ou ***key-value***, ou seja, cada preferência armazenada possui uma identificação ou chave e associada a ela está um valor.

SharedPreferences

- Para usar este recurso você deve adicionar a dependência “**shared_preferences**”[1] no arquivo “**pubspec.yaml**”

```
dependencies:
  flutter:
    sdk: flutter
  shared_preferences: "<newest version>"
```

[1]https://pub.dev/packages/shared_preferences

SharedPreferences

- Para usar o recurso *SharedPreferences*, precisamos usar basicamente 2 tipos de métodos
  - Métodos Getter
  - Métodos Setter
- Além disso temos o método de exclusão, que serve para excluir os dados armazenados

Métodos Getter

- Métodos usados para recuperar dados
  - bool getBool(String key)
  - int getInt(String key)
  - double getDouble(String key)
  - String getString(String key)
  - List<String> getStringList(String key)

Métodos Setter

- Métodos usados para recuperar dados
  - Future<bool> setBool(String key, bool value)
  - Future<bool> setInt(String key, int value)
  - Future<bool> setDouble(String key, double value)
  - Future<bool> setString (String key, String value)
  - Future<bool> setStringList(String key, List<String> value)

Método de Exclusão

- Método de exclusão, que serve para remover os dados armazenados

Future < bool > remove(String key)

Armazenando Dados

- Os métodos ***setter*** fazem duas coisas:
  - atualizam de forma síncrona o par de valores-chave na memória.
  - persista os dados no disco.

Armazenando Dados

```
// obtain shared preferences
final prefs = await SharedPreferences.getInstance();

 // set value
prefs.setInt('counter', counter);
```

Recuperando Dados

- Para ler os dados, use o método ***getter*** apropriado fornecido pelo ***SharedPreferences***.
- Para cada ***setter***, há um ***getter*** correspondente.

```
final prefs = await SharedPreferences.getInstance();

// Try reading data from the counter key. If it doesn't exist, return 0.
final counter = prefs.getInt('counter') ?? 0;
```

Excluindo Dados

- Para excluir dados, use o método **remove**().

```
final prefs = await SharedPreferences.getInstance();

prefs.remove('counter');
```

Referências Bibliográficas

- **Flutter:** https://flutter.dev/
- **Store key-value data on disk:** https://flutter.dev/docs/cookbook/persistence/key-value
- **shared_preferences:** https://pub.dev/packages/shared_preferences