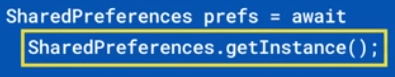
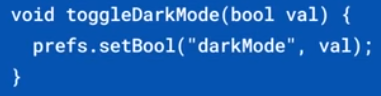
**shared\_preferences**

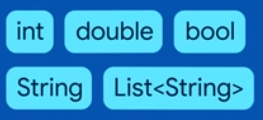
**shared\_preferences: ^2.2.2**



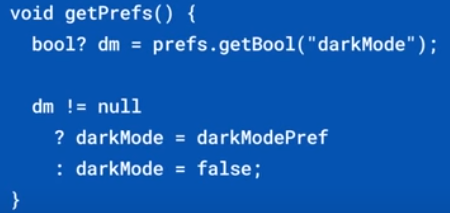
Para guardar un valor:



Tipos de datos que se pueden guardar:



Para buscar un valor:



Creamos el archivo **preferences.dart**

|  |  |
| --- | --- |
| **Código** | **Comentarios** |
| import 'package:shared\_preferences/shared\_preferences.dart';  class Preferences {    static late SharedPreferences \_prefs;    static bool \_isDarkmode = false;  //----------------------------------------------------    static Future init() async {      \_prefs = await SharedPreferences.getInstance();    }  //----------------------------------------------------    static bool get isDarkmode {      return \_prefs.getBool('isDarkmode') ?? \_isDarkmode;    }    static set isDarkmode(bool isDarkmode) {      \_isDarkmode = isDarkmode;      \_prefs.setBool('isDarkmode', isDarkmode);    }  } |  |

En **main.dart** hacemos:

|  |  |
| --- | --- |
| **Código** | **Comentarios** |
| …  void main() async {    WidgetsFlutterBinding.ensureInitialized();    await Preferences.init();    runApp(const MyApp());  }  … |  |

Luego:

|  |  |
| --- | --- |
| **Código** | **Comentarios** |
| import 'package:flutter/material.dart';  import 'package:widgetsoftheweek/share\_preferences/preferences.dart';  class SharedPreferencesScreen extends StatefulWidget {    const SharedPreferencesScreen({Key? key}) : super(key: key);    @override    State<SharedPreferencesScreen> createState() =>        \_SharedPreferencesScreenState();  }  class \_SharedPreferencesScreenState extends State<SharedPreferencesScreen> {    late bool? isDarkMode;      @override    void initState() {      super.initState();      isDarkMode = Preferences.isDarkmode;    }    Widget build(BuildContext context) {      return Scaffold(        backgroundColor: (isDarkMode == null || isDarkMode == false)            ? Colors.amber            : Colors.black,        appBar: AppBar(          title: const Text('SharedPreferences'),          centerTitle: true,        ),        body: Center(          child: (isDarkMode == null || isDarkMode == false)              ? Text('Modo Claro',                  style: TextStyle(                      color: (isDarkMode == null || isDarkMode == false)                          ? Colors.black                          : Colors.amber,                      fontSize: 28,                      fontWeight: FontWeight.bold))              : Text('Modo Oscuro',                  style: TextStyle(                      color: (isDarkMode == null || isDarkMode == false)                          ? Colors.black                          : Colors.amber,                      fontSize: 28,                      fontWeight: FontWeight.bold)),        ),        floatingActionButton: FloatingActionButton(            child: (isDarkMode == null || isDarkMode == false)                ? const Icon(Icons.toggle\_off)                : const Icon(Icons.toggle\_on),            onPressed: () {              if (isDarkMode != null) {                isDarkMode = !isDarkMode!;                Preferences.isDarkmode = isDarkMode!;                setState(() {});              }            }),      );    }  } |  |

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente Gráfico

Descripción generada automáticamente con confianza media

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -