

O Grimório dos Widgets



Dominando a Magia do Flutter



Marcelo Manoel de Lima Filho

01

SIZEDBOX

SIZEDBOX

SizedBox: Espaçamento Simples

O `SizedBox` é usado para adicionar um espaço fixo entre widgets ou definir um tamanho específico para um widget.

Exemplo de uso:

```
SizedBox.dart

Column(
  children: <Widget>[
    Text('Texto Superior'),
    SizedBox(height: 20.0),
    Text('Texto Inferior'),
  ],
)
```

Nesse exemplo, o `SizedBox` adiciona um espaço vertical de 20 pixels entre os textos.



02

CONTAINER

CONTAINER

Container: O Alicerce da Interface

O Container é um dos widgets mais versáteis do Flutter. Ele pode ser usado para adicionar padding, margens, bordas, cores de fundo e tamanhos personalizados.

Exemplo de uso:

```
Container.dart

Container(
  padding: EdgeInsets.all(16.0),
  margin: EdgeInsets.symmetric(vertical: 10.0, horizontal: 20.0),
  decoration: BoxDecoration(
    color: Colors.blue,
    borderRadius: BorderRadius.circular(10.0),
  ),
  child: Text(
    'Hello, Flutter!',
    style: TextStyle(color: Colors.white, fontSize: 20.0),
  ),
)
```

Neste exemplo, o Container tem padding, margens e um fundo azul com cantos arredondados, exibindo o texto "Hello, Flutter!".



03

ROW E COLUMN

BOM E COTRNN

Row e Column: Organizadores de Layout

Os widgets Row e Column são usados para organizar outros widgets horizontalmente e verticalmente, respectivamente.

Exemplo de uso:

```
Row(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: <Widget>[  
    Text('Texto 1'),  
    SizedBox(width: 20.0),  
    Text('Texto 2'),  
  ],  
)
```

Aqui, dois textos são organizados horizontalmente com um espaçamento entre eles.



04

LISTVIEW

ListView: Rolagem Simples e Eficiente

O ListView é ideal para exibir uma lista de itens que podem rolar verticalmente.

Exemplo de uso:

```
ListView.dart

ListView(
  children: <Widget>[
    ListTile(
      leading: Icon(Icons.map),
      title: Text('Map'),
    ),
    ListTile(
      leading: Icon(Icons.photo),
      title: Text('Photos'),
    ),
    ListTile(
      leading: Icon(Icons.phone),
      title: Text('Phone'),
    ),
  ],
)
```

Este ListView exibe uma lista de ícones com títulos.



05

STACK

STACK

Stack:

Sobreposição de Widgets

O Stack permite sobrepor widgets, útil para criar layouts complexos como banners e cards com imagens.

Exemplo de uso:

```
Stack.dart

Stack(
  children: <Widget>[
    Container(
      width: 300.0,
      height: 200.0,
      color: Colors.blue,
    ),
    Positioned(
      bottom: 10.0,
      right: 10.0,
      child: Text(
        'Sobreposição',
        style: TextStyle(color: Colors.white, fontSize: 20.0),
      ),
    ),
  ],
)
```

Aqui, um Container azul é sobreposto por um texto branco no canto inferior direito.



06

SCAFFOLD

Scaffold: Estrutura Básica de uma Página

O Scaffold fornece uma estrutura básica para uma página, incluindo a barra de aplicativos, corpo e rodapé.

Exemplo de uso:

```
Scaffold.dart

Scaffold(
  appBar: AppBar(
    title: Text('Página Inicial'),
  ),
  body: Center(
    child: Text('Bem-vindo ao Flutter!'),
  ),
  floatingActionButton: FloatingActionButton(
    onPressed: () {},
    child: Icon(Icons.add),
  ),
)
```

Este exemplo cria uma página com uma barra de aplicativos, um texto centralizado e um botão de ação flutuante.



07

GESTUREDETECTOR

CE2108EDEC108

GestureDetector: Detecção de Gestos

O GestureDetector permite detectar gestos como toques, arrastes e gestos de pinça.

Exemplo de uso:

```
GestureDetector.dart

GestureDetector(
  onTap: () {
    print('Container foi tocado!');
  },
  child: Container(
    padding: EdgeInsets.all(16.0),
    color: Colors.red,
    child: Text('Toque aqui'),
  ),
)
```

Neste exemplo, um toque no Container imprime uma mensagem no console.



08

FORM

FORM

Form:

Coleta de Dados do Usuário

O Form é usado para agrupar e validar campos de entrada.

Exemplo de uso:

```
Form.dart

final _formKey = GlobalKey<FormState>();

Form(
  key: _formKey,
  child: Column(
    children: <Widget>[
      TextFormField(
        decoration: InputDecoration(labelText: 'Nome'),
        validator: (value) {
          if (value == null || value.isEmpty) {
            return 'Por favor, insira seu nome';
          }
          return null;
        },
      ),
      ElevatedButton(
        onPressed: () {
          if (_formKey.currentState!.validate()) {
            print('Formulário válido');
          }
        },
        child: Text('Enviar'),
      ),
    ],
  ),
)
```

Este formulário valida se o campo "Nome" está preenchido antes de enviar.



09

LAYOUTBUILDER

LAYOUTBUILDER

LayoutBuilder:

Adapte o Layout ao Tamanho da Tela

O LayoutBuilder permite construir layouts que podem se adaptar às mudanças de tamanho de seu pai.

Exemplo de uso:

```
LayoutBuilder.dart

LayoutBuilder(
  builder: (context, constraints) {
    if (constraints.maxWidth > 600) {
      return Text('Tela grande');
    } else {
      return Text('Tela pequena');
    }
  },
)
```

Este exemplo mostra como adaptar o texto exibido com base na largura disponível.



FIM

AGRADECIMENTOS

AGRADECIMENTOS

Obrigado por ler até aqui!

Esse Ebook foi produzido com a assistência de IA's generativas, e diagramado por humano. O passo a passo se encontra no meu Github.

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.



<https://github.com/Marcelo-Filho112/My-prompts-recipe-to-create-a-ebook?tab=readme-ov-file>

