

## ***Operadores matemáticos***

---

O Dart, assim como praticamente qualquer linguagem de programação existente, possui uma série de operadores matemáticos, indo desde as 4 operações básicas da matemática até operadores complexos como resto da divisão e pré-incremento e pós-incremento.

Na tabela podemos ver esse operadores, assim como uma breve explicação sobre seu significado na linguagem.

**Tabela - Operadores aritméticos no Dart**

Operador	Significado
+	Adição
-	Subtração
-expressão	Menos unário, também conhecido como negação (inverter o sinal da expressão)
*	Multiplicação
/	Divisão
~/	Divisão, retornando o inteiro resultante
%	Resto da divisão, também conhecido como módulo da divisão

```
1 void main() {
2     const int primeiro_numero = 10;
3     const int segundo_numero = 3;
4     // Soma
5     print(primeiro_numero + segundo_numero);
6     // Divisão
7     print(primeiro_numero / segundo_numero);
8     // Divisão, retornando o inteiro resultante
9     print(primeiro_numero ~/ segundo_numero);
10    // Resto da divisão
11    print(primeiro_numero % segundo_numero);
12 }
13
```

## ***IconButton***

---

O *Widget IconButton* é uma fusão entre o *Widget TextButton* e o *Widget Icon*. Como o próprio nome já sugere, ele cria um botão com um ícone como elemento clicável, ao contrário do *TextButton*, que faz uso de um texto.

## ***icon***

A propriedade *icon*, como o próprio nome já sugere, é a propriedade responsável por receber o ícone que irá representar a ação do botão.



```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     width: double.infinity,  
7     height: double.infinity,  
8     child: IconButton(  
9       icon: Icon(Icons.add_a_photo),  
10    ),  
11 ),  
12 )
```

Você deve ter percebido que ao tentar interagir com o botão, o mesmo estará desabilitado. Isso ocorre devido ao fato da propriedade *onPressed* não ter sido declarada.

## ***onPressed***

A propriedade *onPressed* como já vimos antes, é a propriedade responsável por manter a ação a ser executada no momento da interação com o botão. Vejamos um exemplo prático.

Para nosso exemplo prático, iremos adicionar uma ação didática, onde ao clicar no botão, iremos apresentar uma mensagem no console informando que o mesmo foi clicado.

□□ Mão no código

```

1 Scaffold(
2   appBar: AppBar(
3     title: Text("Aplicação Exemplo"),
4   ),
5   body: Container(
6     width: double.infinity,
7     height: double.infinity,
8     child: IconButton(
9       icon: Icon(Icons.add_a_photo),
10      onPressed: (){
11        print("Clicou no botão!!!");
12      },
13    ),
14  ),
15)

```

Perceba que agora o botão está ativado e ao clicar nele, vemos uma mensagem em nosso console informando que o mesmo foi clicado.

### *iconSize*

A propriedade *iconSize*, como o próprio nome já sugere, define o tamanho do ícone do botão. E consequentemente, o tamanho do botão como um todo. Para tal, basta informar um valor *double* correspondente ao tamanho desejado.

```

1 Scaffold(
2   appBar: AppBar(
3     title: Text("Aplicação Exemplo"),
4   ),
5   body: Container(
6     width: double.infinity,
7     height: double.infinity,
8     child: IconButton(
9       icon: Icon(Icons.add_a_photo),
10      onPressed: (){
11        print("Clicou no botão!!!");
12      },
13      iconSize: 80.0,
14    ),
15  ),
16)

```

### *color*

A propriedade *color*, como o próprio nome já sugere, nos permite controlar a cor do ícone que será apresentado. Essa propriedade receberá um objeto *Color*. Normalmente

utilizamos umas das constantes previstas na classe *Colors*. Mas podemos utilizar uma cor personalizada, se desejarmos.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     width: double.infinity,  
7     height: double.infinity,  
8     child: IconButton(  
9       icon: Icon(Icons.add_a_photo),  
10      onPressed: () {  
11        print("Clicou no botão!!!");  
12      },  
13      iconSize: 80.0,  
14      color: Colors.green,  
15    ),  
16  ),  
17)
```