Correção "can't have a value of 'null'"

Fala pessoal!

Junto com a nova versão do Flutter foi lançada uma nova funcionalidade chamada **NullSafety**, que nos obriga a fazer uma verificação sobre um valor para evitar que ele seja nulo e cause problemas no código e no app em alguma situação.

Durante os próximos projetos, vocês podem se deparar algumas vezes com a mensagem: "**The parameter 'valor' can't have a value of 'null'**".

Para corrigir isso existem 3 formas. Em alguns casos mais de uma irá funcionar, em outros apenas uma delas irá funcionar.

Nesse primeiro exemplo abaixo será gerado o erro descrito lá no getter:

1. VendoItem({this.produto, this.quantidade = 1});
3. double get preco {
4. if (produto != null && \_preco == 0) {
5. \_preco = produto.precoComDesconto;
6. }
7. return \_preco;
8. }

Para corrigir, basta adicionar o símbolo ! lá dentro do Getter:

1. \_preco = produto!.precoComDesconto;

No próximo exemplo as duas outras formas de correção funcionam, o erro acontece lá nos parâmetros da função “saudacao”:

1. main() {
2. saudacao(idade: 23, nome: "maria");
3. }
5. saudacao({String nome, int idade}) {
6. print(nome);
7. print(idade);
8. }

A primeira forma de se resolver é adicionando o símbolo ? na tipagem dos parâmetros. Dessa forma:

1. main() {
2. saudacao(idade: 23, nome: "maria");
3. }
5. saudacao({String? nome, int? idade}) {
6. print(nome);
7. print(idade);
8. }

A segunda forma é adicionando a palavra reservada “required” nos parâmetros. Dessa forma:

1. main() {
2. saudacao(idade: 23, nome: "maria");
3. }
5. saudacao({required String nome, required int idade}) {
6. print(nome);
7. print(idade);
8. }

Tendo isso como base, você conseguirá lidar com quaisquer problemas similares durante o curso.

Bons estudos!

Atualização na entrada do usuário no Dart

Fala pessoal, tranquilo?

Recentemente, foi lançada uma nova versão do Dart. Agora, ao salvar valores feitos via Input pelo usuário do código, é preciso usar uma interrogação na tipagem do dado. Por exemplo:

1. import 'dart:io';
3. main() {
4. String? teste = stdin.readLineSync();
5. print(teste);
6. }

Essa interrogação significa que o dado que vai vir pode não ser uma String, mas que será convertido assim que for recebido. Ou seja, você pode digitar números e mesmo assim eles serão tratados como String depois.

Caso você utilize tipagem dinâmica (ou seja, utilize 'var' ao invés do tipo exato de dado), é preciso fazer essa indicação no input mesmo, com uma '!' ao invés de uma '?'. Essa solução é válida também ao se utilizar a tipagem.

Dessa forma:

1. import 'dart:io';
3. main() {
4. String teste = stdin.readLineSync()!;
5. print(teste);
6. }

Bons estudos, galera!

Correção no readLineSync()

Fala, pessoal!

Passando apenas para avisar que por conta de atualizações no Dart uma pequena mudança vai ser necessária na próxima aula. Aonde for utilizada a função:

1. stdin.readLineSync();

vai ser necessário adicionar o método toString() no final. Ou seja, deixe dessa forma:

1. stdin.readLineSync().toString();

Bons estudos!

CRIANDO PROJETO FLUTTER

flutter create projeto\_perguntas