

01) O que é uma exceção em Dart?

- a) Um erro que ocorre durante a execução de um programa
- b) Um aviso que é exibido ao usuário
- c) Uma mensagem de depuração para desenvolvedores

02) Como uma exceção é lançada em Dart?

- a) Usando o comando `throw`
- b) Usando o comando `catch`
- c) Usando o comando `try`

03) Qual é a estrutura básica de um bloco try-catch em Dart?

- a) `try {} finally {}`
- b) `try {} catch () {}`
- c) `try {} catch () {} finally {}`

04) Qual é a diferença entre uma exceção e um erro em Dart?

- a) Não há diferença entre exceção e erro em Dart
- b) Uma exceção é lançada manualmente pelo programador, enquanto um erro é um problema interno do sistema
- c) Um erro é lançado manualmente pelo programador, enquanto uma exceção é um problema interno do sistema

05) Como você pode lidar com exceções não tratadas em Dart?

- a) Usando a cláusula `onError`
- b) Usando a cláusula `catchError`
- c) Usando a cláusula `handleError`

06) O que é uma exceção assíncrona em Dart?

- a) Uma exceção que ocorre em código síncrono
- b) Uma exceção que ocorre em código assíncrono
- c) Não há exceções assíncronas em Dart

07) Como você pode lançar uma exceção assíncrona em Dart?

- a) Usando o comando `throw`
- b) Usando o comando `throwAsync`
- c) Usando o comando `throwError`

08) Qual é a função do bloco finally em um bloco try-catch em Dart?

a) Executar um bloco de código após a conclusão do bloco try-catch, independentemente de ocorrer ou não uma exceção

b) Lidar com exceções que não foram tratadas no bloco try-catch

c) Interromper a execução do bloco try-catch

09) Como você pode lidar com várias exceções em Dart?

a) Usando vários blocos try-catch

b) Usando a cláusula onError

c) Usando a cláusula catchError

10) Como você pode criar sua própria classe de exceção em Dart?

a) Estendendo a classe Exception

b) Estendendo a classe Error

c) Não é possível criar sua própria classe de exceção em Dart