

01) O que é Fluxo de Controle em Dart?

O fluxo de controle em Dart se refere à maneira como o programa segue uma sequência de instruções, pulando ou repetindo blocos de código, com base em certas condições e comandos.

02) Quais são as três estruturas básicas de Fluxo de Controle em Dart?

- Estruturas de decisão (if-else, switch)
- Loops (while, do-while, for)
- Estruturas de controle de exceção (try-catch-finally)

03) O que é uma instrução if em Dart?

Uma estrutura de decisão que permite que o programa execute um bloco de código se uma determinada condição for verdadeira.

04) O que é uma instrução if-else em Dart?

Uma estrutura de decisão que permite executar um bloco de código se uma condição for verdadeira, e outro bloco de código se a condição for falsa.

05) O que é uma instrução switch em Dart?

É uma estrutura de controle que permite avaliar uma expressão e compará-la com várias possibilidades de valores. Cada possibilidade é representada por um caso case.

06) Como usar a instrução switch em Dart?

```
switch (expressão) {  
  case valor1:  
    // bloco de código a ser executado se a expressão for igual a valor1  
    break;  
  case valor2:  
    // bloco de código a ser executado se a expressão for igual a valor2  
    break;  
  case valor3:  
    // bloco de código a ser executado se a expressão for igual a valor3  
    break;  
  // outros casos  
  default:  
    // bloco de código a ser executado se a expressão não for igual a nenhum dos casos  
}
```

07) O que é uma instrução for em Dart?

É uma estrutura de controle que permite executar um bloco de código repetidamente por um número definido de vezes.

08) Como usar a instrução for em Dart?

```
for (inicialização; condição; atualização) {  
    // bloco de código a ser executado repetidamente  
}
```

09) O que é uma instrução while em Dart?

É uma estrutura de controle que permite executar um bloco de código repetidamente enquanto uma condição for verdadeira.

10) Como usar a instrução while em Dart?

```
var i = 1;  
while (i <= 5) {  
    print(i);  
    i++;  
}
```

11) O que é uma instrução do-while em Dart?

Também é uma estrutura de controle que permite executar um bloco de código repetidamente enquanto uma condição for verdadeira. A principal diferença entre a instrução while e a instrução do-while é que o bloco de código dentro do loop da instrução do-while é executado pelo menos uma vez, mesmo que a condição seja falsa na primeira iteração.

12) Como usar a instrução do-while em Dart?

```
do {  
    // bloco de código a ser executado repetidamente  
} while (condição);  
  
var i = 1;  
do {  
    print(i);  
    i++;  
} while (i <= 5);
```

13) O que é uma instrução break em Dart?

É utilizada para interromper a execução de um loop (for, while, do-while) ou de um switch case. Quando a instrução break é executada dentro de um loop, o controle é transferido imediatamente para a próxima linha de código após o loop.

14) Como usar a instrução break em Dart?

```
for (var i = 0; i < 10; i++) {  
  if (i == 5) {  
    break;  
  }  
  print(i);  
}
```

OU

```
var opcao = 'b';  
switch (opcao) {  
  case 'a':  
    print('Opção A selecionada');  
    break;  
  case 'b':  
    print('Opção B selecionada');  
    break;  
  case 'c':  
    print('Opção C selecionada');  
    break;  
  default:  
    print('Opção inválida');  
}
```

15) O que é uma instrução continue em Dart?

A instrução continue é utilizada dentro de laços de repetição para interromper a execução do bloco de código atual e avançar para a próxima iteração do laço.

16) Como usar a instrução continue em Dart?

```
for (var i = 0; i < 10; i++) {  
  if (i == 5) {  
    continue;  
  }  
  print(i);  
}
```

//0 1 2 3 4 5 6 7 8 9

17) O que é uma instrução try-catch em Dart?

A instrução try-catch em Dart é usada para lidar com exceções que podem ocorrer durante a execução do código. Ela é usada para "tentar" executar um bloco de código e, se uma exceção for lançada, o controle é transferido para o bloco "catch" correspondente para lidar com a exceção.

18) Como usar a instrução try-catch em Dart?

```
try {  
  // Código que pode gerar uma exceção  
} catch (exception) {  
  // Código para lidar com a exceção  
}
```

19) O que é uma instrução finally em Dart?

A instrução finally é usada juntamente com a instrução try-catch em Dart. Ela é usada para executar um bloco de código, independentemente de ocorrer uma exceção ou não. O bloco de código dentro da instrução finally será sempre executado após a conclusão do bloco de código dentro do bloco try, mesmo que ocorra uma exceção.

20) Como usar a instrução finally em Dart?

```
try {  
  // bloco de código que pode gerar exceções  
} catch (e) {  
  // tratamento de exceções  
} finally {  
  // bloco de código que será executado sempre  
}
```