Responda às seguintes questões:

1) O que é uma lista em Dart?

Uma lista em Dart é uma coleção ordenada de elementos do mesmo tipo.

```dart

void main () {

var lista = [1, 2, 3];

print(lista);

}

```

Saída:

```

[1, 2, 3]

```

2) Como criar uma lista vazia em Dart?

Para criar uma lista vazia em Dart, você pode utilizar o construtor List().

Por exemplo:

```dart

void main () {

var lista = List();

print(lista);

}

```

Saída:

```

[]

```

3) Como criar uma lista com elementos em Dart?

Para criar uma lista com elementos em Dart, você pode utilizar o construtor List().

Por exemplo:

```dart

void main () {

var lista = [1, 2, 3];

print(lista);

}

```

Saída:

```

[1, 2, 3]

```

4) Qual a diferença entre uma lista e um conjunto em Dart?

A principal diferença entre uma lista e um conjunto em Dart é que uma lista é uma coleção ordenada de elementos do mesmo tipo, enquanto um conjunto é uma coleção não ordenada de elementos únicos do mesmo tipo.

Por exemplo:

```dart

void main () {

var lista = [1, 2, 3];

var conjunto = {1, 2, 3};

print(lista);

print(conjunto);

}

```

Saída:

```

[1, 2, 3]

{1, 2, 3}

```

5) Como acessar um elemento específico de uma lista em Dart?

Para acessar um elemento específico de uma lista em Dart, você pode utilizar o operador [].

Por exemplo:

```dart

void main () {

var lista = [1, 2, 3];

print(lista[0]);

}

```

Saída:

```

1

```

6) Como adicionar um elemento ao final de uma lista em Dart?

Para adicionar um elemento ao final de uma lista em Dart, você pode utilizar o método add().

Por exemplo:

```dart

void main () {

var lista = [1, 2, 3];

lista.add(4);

print(lista);

}

```

Saída:

```

[1, 2, 3, 4]

```

7) Como inserir um elemento em uma posição específica de uma lista em Dart?

Para inserir um elemento em uma posição específica de uma lista em Dart, você pode usar o método `insert(index, element)` da classe `List`. O primeiro argumento é o índice da posição onde você deseja inserir o elemento e o segundo argumento é o elemento que você deseja inserir.

Por exemplo, para inserir o número 5 na posição 2 da lista, você pode fazer assim:

```dart

var list = [1, 2, 3, 4];

list.insert(2, 5);

print(list); // [1, 2, 5, 3, 4]

8) Como remover um elemento de uma lista em Dart?

Para remover um elemento de uma lista em Dart, você pode utilizar o método `remove()` passando o valor a ser removido como parâmetro. Por exemplo:

```dart

var lista = [1, 2, 3];

lista.remove(2); // remove o número 2 da lista

```

Você também pode utilizar outros métodos como `removeAt()`, `removeRange()`, `removeLast()` e `removeWhere()` para remover elementos de uma lista em Dart¹.

9) Como verificar se uma lista contém um determinado elemento em Dart?

Para verificar se uma lista contém um determinado elemento em Dart, você pode utilizar o método `contains()` passando o valor a ser verificado como parâmetro. Por exemplo:

```dart

var lista = [1, 2, 3];

print(lista.contains(2)); // true

```

Você também pode utilizar outros métodos como `isEmpty()`, `isNotEmpty()`, `first`, `last` e `clear()` para verificar se uma lista contém elementos em Dart¹⁴.

10) Como ordenar uma lista em ordem crescente em Dart?

Para ordenar uma lista em ordem crescente em Dart, você pode usar o método `sort()` que é fornecido pela classe List do Dart. O método `sort()` classifica os elementos da lista em ordem crescente.

Aqui está um exemplo de como usar o método `sort()` para ordenar uma lista de inteiros em ordem crescente:

```dart

void main() {

List<int> numeros = [5, 2, 9, 1, 3];

numeros.sort();

print(numeros);

}

```

Este código irá imprimir `[1, 2, 3, 5, 9]`.

Se você quiser ordenar uma lista de objetos em ordem crescente com base em um campo específico do objeto, você pode passar uma função de comparação personalizada para o método `sort()`. Aqui está um exemplo:

```dart

class Pessoa {

String nome;

int idade;

Pessoa(this.nome, this.idade);

}

void main() {

List<Pessoa> pessoas = [

Pessoa('Keila', 48),

Pessoa('Isabel', 20),

Pessoa('Karol', 14),

];

pessoas.sort((a, b) => a.idade.compareTo(b.idade));

print(pessoas.map((p) => p.nome).toList());

}

```

Este código irá imprimir `[Karol, Isabel, Keila]`, que é a lista de nomes das pessoas ordenada por idade em ordem crescente.

11) Como ordenar uma lista em ordem decrescente em Dart?

Para ordenar uma lista em ordem decrescente em Dart, você pode usar o método `sort()` e passar uma função de comparação que retorna um valor negativo se o primeiro argumento for menor que o segundo argumento, um valor positivo se o primeiro argumento for maior que o segundo argumento e zero se os dois argumentos forem iguais. Para ordenar em ordem decrescente, você pode inverter a ordem dos argumentos na função de comparação.

Exemplo de código que ordena uma lista de inteiros em ordem decrescente:

```dart

void main() {

var numeros = [5, 1, 3, 4, 2];

numeros.sort((a, b) => b.compareTo(a));

print(numeros); // [5, 4, 3, 2, 1]

}

```

A função de comparação é `(a, b) => b.compareTo(a)`, que retorna um valor negativo se `a` for menor que `b`, um valor positivo se `a` for maior que `b` e zero se `a` e `b` forem iguais.

12) Como copiar uma lista em Dart?

Para copiar uma lista em Dart, você pode usar o método `List.from()`. O método cria uma nova lista com os mesmos elementos da lista original.

Exemplo de como copiar uma lista em Dart:

```dart

void main() {

var listaOriginal = [1, 2, 3];

var listaCopia = List.from(listaOriginal);

print('Lista original: $listaOriginal');

print('Lista cópia: $listaCopia');

}

```

Este código imprimirá:

```

Lista original: [1, 2, 3]

Lista cópia: [1, 2, 3]

```

13) Como verificar se duas listas são iguais em Dart?

Para verificar se duas listas são iguais em Dart, você pode usar o método `ListEquality().equals(list1, list2)`. Esse método retorna `true` se as duas listas tiverem os mesmos elementos na mesma ordem e com a mesma quantidade. Caso contrário, ele retorna `false`.

Por exemplo:

```

import 'package:collection/collection.dart';

void main() {

var list1 = [1, 2, 3];

var list2 = [1, 2, 3];

var list3 = [3, 2, 1];

print(ListEquality().equals(list1, list2)); // true

print(ListEquality().equals(list1, list3)); // false

}

```

14) Como criar uma lista a partir de outra lista em Dart?

Para criar uma lista a partir de outra lista em Dart, você pode usar o método `map()` que é usado para transformar cada elemento de uma lista em outro elemento. O método `toList()` é usado para converter o resultado de volta em uma lista.

Aqui está um exemplo de como criar uma nova lista a partir de outra lista em Dart:

```dart

List<int> numeros = [1, 2, 3];

List<String> numerosEmString = numeros.map((numero) => numero.toString()).toList();

```

Neste exemplo, criamos uma lista chamada `numeros` com três elementos e depois usamos o método `map()` para transformar cada número em uma string. Em seguida, usamos o método `toList()` para converter o resultado de volta em uma lista.

15) Como transformar uma lista em uma lista de strings em Dart?

Para transformar uma lista em uma lista de strings em Dart, você pode usar o método `map()` que é usado para transformar cada elemento de uma lista em outro elemento. No caso, você pode usar o método `toString()` para transformar cada elemento da lista em uma string.

Aqui está um exemplo de como fazer isso:

```dart

List<int> lista = [1, 2, 3];

List<String> listaDeStrings = lista.map((i) => i.toString()).toList();

```

Neste exemplo, a lista `[1, 2, 3]` é convertida em uma lista de strings `['1', '2', '3']`.

16) Como calcular a soma dos elementos de uma lista em Dart?

Para calcular a soma dos elementos de uma lista em Dart, você pode usar o método `reduce()` que é aplicado em uma lista e recebe uma função que define como os elementos da lista serão combinados para produzir um único valor. A função passada para o método `reduce()` recebe dois parâmetros, o primeiro é o valor acumulado até o momento e o segundo é o próximo elemento da lista. A função deve retornar a soma desses dois valores.

Aqui está um exemplo de como usar o método `reduce()` para calcular a soma dos elementos de uma lista:

```dart

void main() {

List<int> numeros = [1, 2, 3, 4, 5];

int soma = numeros.reduce((a, b) => a + b);

print(soma); // Output: 15

}

```

Neste exemplo, a variável `numeros` contém a lista de números que queremos somar. Em seguida, usamos o método `reduce()` para calcular a soma dos elementos da lista e armazenamos o resultado na variável `soma`. Finalmente, imprimimos o valor da variável `soma` na tela.

17) Como calcular a média dos elementos de uma lista em Dart?

Para calcular a média dos elementos de uma lista em Dart, você pode usar o seguinte código:

```dart

double average(List<int> list) {

var sum = 0;

list.forEach((element) => sum += element);

return sum / list.length;

}

```

O código acima calcula a soma de todos os elementos da lista e divide pelo número de elementos para obter a média.

18) Como calcular o valor máximo e mínimo de uma lista em Dart?

Para calcular o valor máximo e mínimo de uma lista em Dart, você pode usar os métodos `reduce()` e `fold()`. O método `reduce()` é usado para reduzir todos os elementos de uma lista a um único valor, enquanto o método `fold()` é usado para combinar todos os elementos de uma lista em um único valor.

Para encontrar o valor máximo de uma lista em Dart, você pode usar o método `reduce()`. O método `reduce()` recebe uma função que compara dois elementos da lista e retorna o maior deles. Aqui está um exemplo:

```

var numbers = [1, 2, 3, 4, 5];

var max = numbers.reduce((a, b) => a > b ? a : b);

print(max); // Output: 5

```

Para encontrar o valor mínimo de uma lista em Dart, você pode usar o método `fold()`. O método `fold()` recebe um valor inicial e uma função que combina esse valor com cada elemento da lista para produzir um único resultado. Aqui está um exemplo:

```

var numbers = [1, 2, 3, 4, 5];

var min = numbers.fold(numbers[0], (a, b) => a < b ? a : b);

print(min); // Output: 1

```

19) Como contar quantas vezes um elemento aparece em uma lista em Dart?

Para contar quantas vezes um elemento aparece em uma lista em Dart, você pode usar o método `where()` para filtrar os elementos iguais ao elemento que você deseja contar e depois usar o método `length` para obter o número de elementos iguais.

Aqui está um exemplo de código que conta quantas vezes o número 1 aparece na lista:

```dart

void main() {

List<int> numeros = [1, 2, 3, 1, 4, 5, 1];

int numeroProcurado = 1;

int quantidade = numeros.where((numero) => numero == numeroProcurado).length;

print('O número $numeroProcurado aparece $quantidade vezes na lista.');

}

```

Este código imprime:O número 1 aparece 3 vezes na lista.

20) Como remover todos os elementos duplicados de uma lista em Dart?

Para remover todos os elementos duplicados de uma lista em Dart, você pode utilizar o método `distinct()` que remove os itens duplicados gerando um novo objeto. Por exemplo, para remover os elementos duplicados de uma lista de inteiros chamada `listaDeInteirosVelha`, você pode fazer o seguinte:

```dart

List<int> novaListaDeInteiros = listaDeInteirosVelha.distinct().toList();

```Isso irá retornar uma sequência `List<int>` preenchida sem repetir os dados.

21) Espero que essa prova ajude a avaliar o conhecimento dos alunos sobre Listas em Dart!