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. É possível adicionar, remover e acessar elementos da lista.

2) Como criar uma lista vazia em Dart?

Para criar uma lista vazia em Dart, pode-se usar a seguinte sintaxe: List<int> listaVazia = [];

3) Como criar uma lista com elementos em Dart?

Para criar uma lista com elementos em Dart, pode-se usar a seguinte sintaxe: List<String> lista = ['elemento1', 'elemento2', 'elemento3'];

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

A diferença entre uma lista e um conjunto em Dart é que uma lista permite elementos duplicados e é ordenada, enquanto um conjunto não permite elementos duplicados e não é ordenado.

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

Para acessar um elemento específico de uma lista em Dart, pode-se usar o índice do elemento na lista. Por exemplo, para acessar o segundo elemento de uma lista, pode-se usar a seguinte sintaxe: lista[1]

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

Para adicionar um elemento ao final de uma lista em Dart, pode-se usar o método add(). Por exemplo, para adicionar o elemento "elemento4" ao final da lista anterior, pode-se usar a seguinte sintaxe: lista.add('elemento4');

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, pode-se usar o método insert(). Por exemplo, para inserir o elemento "elemento0" na primeira posição da lista anterior, pode-se usar a seguinte sintaxe: lista.insert(0, 'elemento0');

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

Para remover um elemento de uma lista em Dart, pode-se usar o método remove(). Por exemplo, para remover o elemento "elemento1" da lista anterior, pode-se usar a seguinte sintaxe: lista.remove('elemento1');

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

Para verificar se uma lista contém um determinado elemento em Dart, pode-se usar o método contains(). Por exemplo, para verificar se a lista anterior contém o elemento "elemento2", pode-se usar a seguinte sintaxe: lista.contains('elemento2');

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

Para ordenar uma lista em ordem crescente em Dart, pode-se usar o método sort(). Por exemplo, para ordenar a lista anterior em ordem crescente, pode-se usar a seguinte sintaxe: lista.sort();

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

Para ordenar uma lista em ordem decrescente em Dart, pode-se usar o método sort() em conjunto com o método reversed(). Por exemplo, para ordenar a lista anterior em ordem decrescente, pode-se usar a seguinte sintaxe: lista.sort((a, b) => b.compareTo(a));

12) Como copiar uma lista em Dart?

Para copiar uma lista em Dart, pode-se usar o método List.from(). Por exemplo, para copiar a lista anterior em uma nova lista, pode-se usar a seguinte sintaxe: List<String> novaLista = List.from(lista);

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

Para verificar se duas listas são iguais em Dart, pode-se usar o método equals(). Por exemplo, para verificar se a nova lista é igual à lista anterior, pode-se usar a seguinte sintaxe: novaLista.equals(lista);

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

Para criar uma lista a partir de outra lista em Dart, pode-se usar o método List.from(). Por exemplo, para criar uma nova lista com os elementos da lista anterior, pode-se usar a seguinte sintaxe: List<String> novaLista = List.from(lista);

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

Para transformar uma lista em uma lista de strings em Dart, podemos utilizar o método

map() junto com o método toString(). Veja o exemplo abaixo:

```
List<int> numeros = [1, 2, 3, 4, 5];
List<String> numerosString = numeros.map((numero) => numero.toString()).toList();
print(numerosString); // [1, 2, 3, 4, 5]
```

Nesse exemplo, o método map() é usado para transformar cada elemento da lista de números em uma string, utilizando o método toString(). O método toList() é usado para transformar o resultado do map() novamente em uma lista.

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

Para calcular a soma dos elementos de uma lista em Dart, podemos utilizar o método reduce() da classe List. O método reduce() aplica uma função aos elementos da lista de forma cumulativa, reduzindo a lista a um único valor. Veja o exemplo abaixo:

```
List<int> numeros = [1, 2, 3, 4, 5];
int soma = numeros.reduce((valor, elemento) => valor + elemento);
print(soma); // 15
```

Nesse exemplo, o método reduce() é usado para somar os elementos da lista. A função passada para o reduce() recebe dois parâmetros: o valor acumulado até o momento e o próximo elemento da lista.

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

Para calcular a média dos elementos de uma lista em Dart, podemos utilizar a soma dos elementos e o tamanho da lista. Veja o exemplo abaixo:

```
List<double> numeros = [1.0, 2.0, 3.0, 4.0, 5.0];
double soma = numeros.reduce((valor, elemento) => valor + elemento);
double media = soma / numeros.length;
print(media); // 3.0
```

Nesse exemplo, a soma dos elementos é calculada utilizando o método reduce(), e a média é calculada dividindo a soma pelo tamanho da lista.

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, podemos utilizar os métodos max() e min() da classe List. Veja o exemplo abaixo:

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

```
int maximo = numeros.max();
int minimo = numeros.min();
print(maximo); // 5
print(minimo); // 1
```

Nesse exemplo, os métodos max() e min() são usados para obter o valor máximo e mínimo da lista, respectivamente.

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, podemos utilizar o método where() da classe List em conjunto com o método length. Veja o exemplo abaixo:

```
List<int> numeros = [1, 2, 3, 2, 4, 2, 5];
int elemento = 2;
int ocorrencias = numeros.where((numero) => numero == elemento).length;
print(ocorrencias); // 3
```

Nesse exemplo, o método where() é usado para filtrar a lista e retornar apenas os elementos que são iguais ao elemento desejado. Em seguida, o método length é usado para obter o número de elementos na lista resultante.

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

Para remover todos os elementos duplicados de uma lista em Dart, podemos utilizar o método toSet() para converter a lista em um conjunto, e depois convertê-lo de volta para uma lista usando o método toList(). O conjunto irá automaticamente remover todos os elementos duplicados:

```
List<int> numeros = [1, 2, 3, 4, 5, 3, 4, 6];
List<int> numerosSemDuplicatas = numeros.toSet().toList();
print(numerosSemDuplicatas); // Output: [1, 2, 3, 4, 5, 6]
```

Neste exemplo, a lista original numeros possui alguns elementos duplicados (o número 3 e 4 aparecem duas vezes cada). Após a conversão para um conjunto e volta para uma lista, a lista resultante numerosSemDuplicatas contém apenas uma ocorrência de cada número.