Responda às seguintes questões:

01) O que é um conjunto em Dart?

Um conjunto em Dart é uma coleção não ordenada de itens exclusivos¹. O suporte a dardos para conjuntos é fornecido por literais de conjuntos e pelo tipo Set¹.

Set<String> conjunto = {'a', 'b', 'c'};

O código acima cria um conjunto com três elementos.

02) Como criar um conjunto vazio em Dart?

Set<String> conjunto = Set();

03) Como criar um conjunto com elementos em Dart?

Para criar um conjunto com elementos em Dart, você pode usar o construtor padrão do tipo Set e adicionar os elementos usando o método add.

Set<String> conjunto = Set();

conjunto.add('elemento1');

conjunto.add('elemento2');

O código cria um conjunto com dois elementos.

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

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

Em outras palavras, os elementos em uma lista são acessados por meio de índices, enquanto os elementos em um conjunto não têm ordem e não podem ser acessados por índices¹. Além disso, os conjuntos não permitem elementos duplicados.

05) Como adicionar um elemento a um conjunto em Dart?

Para adicionar um elemento a um conjunto em Dart, você pode usar o método add() ou addAll()¹. O método add() adiciona um único elemento ao conjunto, enquanto o método addAll() adiciona vários elementos ao conjunto de uma só vez¹.

Aqui está um exemplo de como adicionar um elemento a um conjunto em Dart:

```dart

void main() {

var conjunto = <String>{};

conjunto.add('elemento');

print(conjunto);

}

```

Este código cria um conjunto vazio e adiciona o elemento "elemento" ao conjunto usando o método add(). O resultado é a impressão do conjunto com o novo elemento adicionado.

06) Como remover um elemento de um conjunto em Dart?

Para remover um elemento de um conjunto em Dart, você pode utilizar o método remove() passando o valor a ser removido como parâmetro ou o método removeAt() passando a posição do elemento a ser removido como parâmetro.

Por exemplo:

```dart

void main () {

var conjunto = {1, 2, 3, 4};

conjunto.remove(2);

print(conjunto);

}

```

Saída:

```

{1, 3, 4}

```

07) Como verificar se um conjunto contém um determinado elemento em Dart?

Para verificar se um conjunto contém um determinado elemento em Dart, você pode utilizar o método contains() passando o valor a ser verificado como parâmetro.

Por exemplo:

```dart

void main () {

var conjunto = {1, 2, 3, 4};

print(conjunto.contains(2));

}

```

Saída:

```

true

```

08) Como verificar se um conjunto é vazio em Dart?

Para verificar se um conjunto é vazio em Dart, você pode utilizar o método isEmpty().

Por exemplo:

```dart

void main () {

var conjunto = {1, 2, 3, 4};

print(conjunto.isEmpty);

}

```

Saída:

```

false

```

09) Como unir dois conjuntos em Dart?

Para unir dois conjuntos em Dart, você pode utilizar o método union().

Por exemplo:

```dart

void main () {

var conjunto1 = {1, 2, 3};

var conjunto2 = {3, 4, 5};

var conjuntoUniao = conjunto1.union(conjunto2);

print(conjuntoUniao);

}

```

Saída:

```

{1, 2, 3, 4, 5}

```

10) Como obter a interseção de dois conjuntos em Dart?

Para obter a interseção de dois conjuntos em Dart, você pode utilizar o método intersection().

Por exemplo:

```dart

void main () {

var conjunto1 = {1, 2, 3};

var conjunto2 = {3, 4, 5};

var conjuntoIntersecao = conjunto1.intersection(conjunto2);

print(conjuntoIntersecao);

}

```

Saída:

```

{3}

```

11) Como obter a diferença entre dois conjuntos em Dart?

Para obter a diferença entre dois conjuntos em Dart, você pode utilizar o método difference().

Por exemplo:

```dart

void main () {

var conjunto1 = {1, 2, 3};

var conjunto2 = {3, 4, 5};

var conjuntoDiferenca = conjunto1.difference(conjunto2);

print(conjuntoDiferenca);

}

```

Saída:

```

{1, 2}

```

12) Como verificar se um conjunto é subconjunto de outro conjunto em Dart?

Para verificar se um conjunto é subconjunto de outro conjunto em Dart, você pode utilizar o método containsAll().

Por exemplo:

```dart

void main () {

var conjunto1 = {1, 2, 3};

var conjunto2 = {1, 2};

var conjunto3 = {4, 5};

print(conjunto1.containsAll(conjunto2)); // true

print(conjunto1.containsAll(conjunto3)); // false

}

```

Saída:

```

true

false

```

13) Como verificar se dois conjuntos são iguais em Dart?

Para verificar se dois conjuntos são iguais em Dart, você pode utilizar o método containsAll().

Por exemplo:

```dart

void main () {

var conjunto1 = {1, 2, 3};

var conjunto2 = {1, 2, 3};

var conjunto3 = {4, 5};

print(conjunto1.containsAll(conjunto2)); // true

print(conjunto1.containsAll(conjunto3)); // false

}

```

Saída:

```

true

false

```

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

Para criar um conjunto a partir de uma lista em Dart, você pode utilizar o construtor Set().

Por exemplo:

```dart

void main () {

var lista = [1, 2, 3];

var conjunto = Set.from(lista);

print(conjunto);

}

```

Saída:

```

{1, 2, 3}

```

15) Como criar uma lista a partir de um conjunto em Dart?

Para criar um conjunto a partir de uma lista em Dart, você pode utilizar o construtor Set().

Por exemplo:

```dart

void main () {

var lista = [1, 2, 3];

var conjunto = Set.from(lista);

print(conjunto);

}

```

Saída:

```

{1, 2, 3}

```

16) Como transformar um conjunto em uma lista de strings em Dart?

Para transformar um conjunto em uma lista de strings em Dart, você pode utilizar o método map().

Por exemplo:

```dart

void main () {

var conjunto = {1, 2, 3};

var listaStrings = conjunto.map((elemento) => elemento.toString()).toList();

print(listaStrings);

}

```

Saída:

```

[1, 2, 3]

```

17) Como calcular a união de vários conjuntos em Dart?

Para calcular a união de vários conjuntos em Dart, você pode utilizar o método union().

Por exemplo:

```dart

void main () {

var conjunto1 = {1, 2, 3};

var conjunto2 = {3, 4, 5};

var conjunto3 = {5, 6, 7};

var uniao = conjunto1.union(conjunto2).union(conjunto3);

print(uniao);

}

```

Saída:

```

{1, 2, 3, 4, 5, 6, 7}

```

18) Como calcular a interseção de vários conjuntos em Dart?

Para calcular a interseção de vários conjuntos em Dart, você pode utilizar o método intersection().

Por exemplo:

```dart

void main () {

var conjunto1 = {1, 2, 3};

var conjunto2 = {3, 4, 5};

var conjunto3 = {5, 6, 7};

var intersecao = conjunto1.intersection(conjunto2).intersection(conjunto3);

print(intersecao);

}

```

Saída:

```

{}

```

19) Como verificar se dois conjuntos são disjuntos em Dart?

Para verificar se dois conjuntos são disjuntos em Dart, você pode utilizar o método intersection().

Por exemplo:

```dart

void main () {

var conjunto1 = {1, 2, 3};

var conjunto2 = {4, 5, 6};

var conjunto3 = {3, 4, 5};

print(conjunto1.intersection(conjunto2).isEmpty); // true

print(conjunto1.intersection(conjunto3).isEmpty); // false

}

```

Saída:

```

true

false

```

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

Para remover todos os elementos duplicados de uma lista usando um conjunto em Dart, você pode utilizar o construtor Set().

Por exemplo:

```dart

void main () {

var lista = [1, 2, 3, 2, 1];

var conjunto = Set.from(lista);

var listaSemDuplicados = conjunto.toList();

print(listaSemDuplicados);

}

```

Saída:

```

[1, 2, 3]

```