DART

SET

A Set is a collection of unique items (no duplicates). It's like a List but automatically ensures no repeated elements.

Creating a Set

```
Set<String> fruits = {'apple', 'banana', 'orange'};
Set<int> numbers = Set(); // empty set
```

SET METHODS

1. add()

```
void main () {
  Set<String> fruits = {'apple', 'banana'};
  fruits.add ('orange');
  print(fruits); // {apple, banana, orange}
}
```

2. addAll()

```
void main() {
  Set<String> fruits = {'apple'};
  fruits.addAll(['banana', 'orange']);
  print(fruits); // {apple, banana, orange}
}
```

3. remove()

```
void main() {
  Set<String> fruits = {'apple', 'banana', 'orange'};
  fruits.remove('banana');
  print(fruits); // {apple, orange}
```

```
}
```

4. contains()

```
void main() {
  Set<String> fruits = {'apple', 'banana'};
  print(fruits.contains('banana')); // true
  print(fruits.contains('grape'));  // false
}
```

5. clear()

```
void main() {
  Set<String> fruits = {'apple', 'banana'};
  fruits.clear();
  print(fruits); // {}
}
```

6. length

```
void main() {
  Set<String> fruits = {'apple', 'banana', 'orange'};
  print(fruits.length); // 3
}
```

7. isEmpty & isNotEmpty

```
void main() {
  Set<String> fruits = {};
  print(fruits.isEmpty);    // true
  print(fruits.isNotEmpty); // false
}
```

8. forEach ()

```
void main () {
  Set<String> fruits = {'apple', 'banana'};
  fruits.forEach((fruit) {
    print(fruit);
  });
}
```

9. toList ()

```
void main () {
  Set<String> fruits = {'apple', 'banana'};
  List<String> fruitList = fruits.toList();
  print(fruitList); // [apple, banana]
}
```

10. Union ()

```
void main () {
  Set<String> a = {'apple', 'banana'};
  Set<String> b = {'banana', 'orange'};
  print(a.union(b)); // {apple, banana, orange}
}
```

11. Intersection ()

```
void main () {
  Set<String> a = {'apple', 'banana'};
  Set<String> b = {'banana', 'orange'};
  print(a.intersection(b)); // {banana}
}
```

12. Difference ()

```
void main () {
  Set<String> a = {'apple', 'banana'};
  Set<String> b = {'banana', 'orange'};
  print(a.difference(b)); // {apple}
}
```

13. Single ()

```
void main () {
  Set<String> items = {'onlyOne'};

  print(items.single);  // Output: onlyOne
}
```

Error Example (More than one element)

```
void main () {
  Set<String> items = {'one', 'two'};

  print(items.single);  // Throws StateError: Too many elements
}
```