Generics

In Dart, you can use generics to avoid using specific data types in your code. For instance, you can write a class that allows you to perform arithmetic operations on an integer but soon you will realize that the operations might not only be applicable to integers, but also on doubles or even other similar numeric data types. In that case, you might want to use generics to avoid having to re-coding your implementation.

Further reading:

- Generics - dart.dev
- Generics in Dart and Flutter - dart.academy
- Dart Generics - techLog
- Dart Programming - Generics - Tutorialspoint
- Dart Generics - Javatpoint

Examples

```dart
// here we have a class named Stack that works
// in such a way that generic values of type T,
// a type we have just named T for the sake of
// simplicity, can be stored in a first-in-last-out
// fashion with the push and pop functions
class Stack<T> {
  final List<T> _values = [];
  // the push function accepts values of type T
  void push(T value) => _values.insert(0, value);
  // and the pop function returns values of
  // type T too
  T? pop() => _values.isNotEmpty ? _values.removeAt(0) : null;
}
```