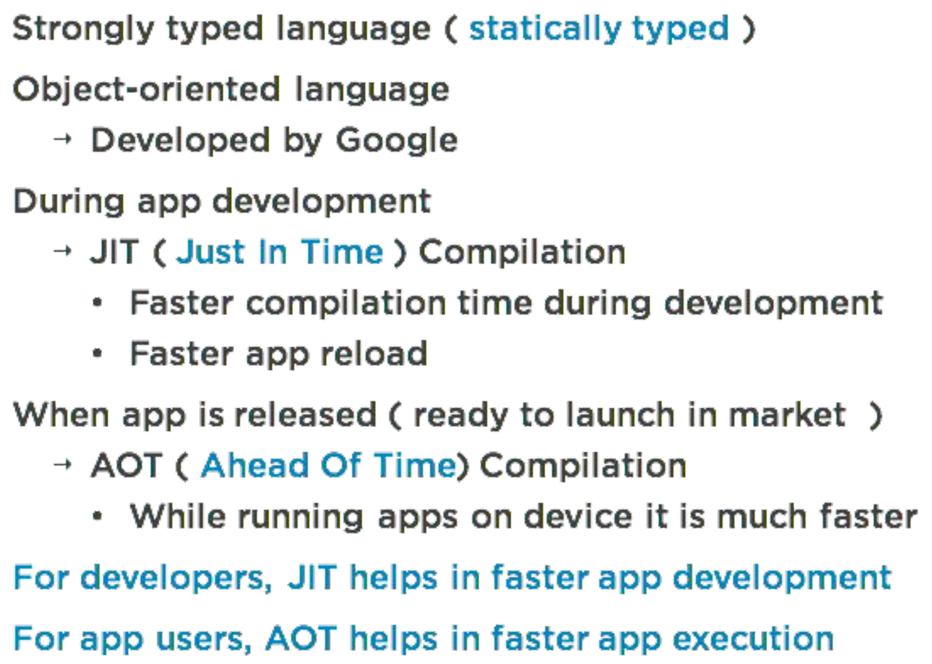
Flutter Study Jam

GDG BBSR

Women Techmakers
More women in technology

Flutter uses Dart



Strongly typed language (statically typed)

Object-oriented language

→ Developed by Google

During app development

→ JIT (Just In Time) Compilation

- Faster compilation time during development
- Faster app reload

When app is released (ready to launch in market)

→ AOT (Ahead Of Time) Compilation

- While running apps on device it is much faster

For developers, JIT helps in faster app development

For app users, AOT helps in faster app execution

Flutter is Different

Faster development and execution due to JIT and AOT

Similar user-ex compared to Native apps like Android and iOS

Single code base

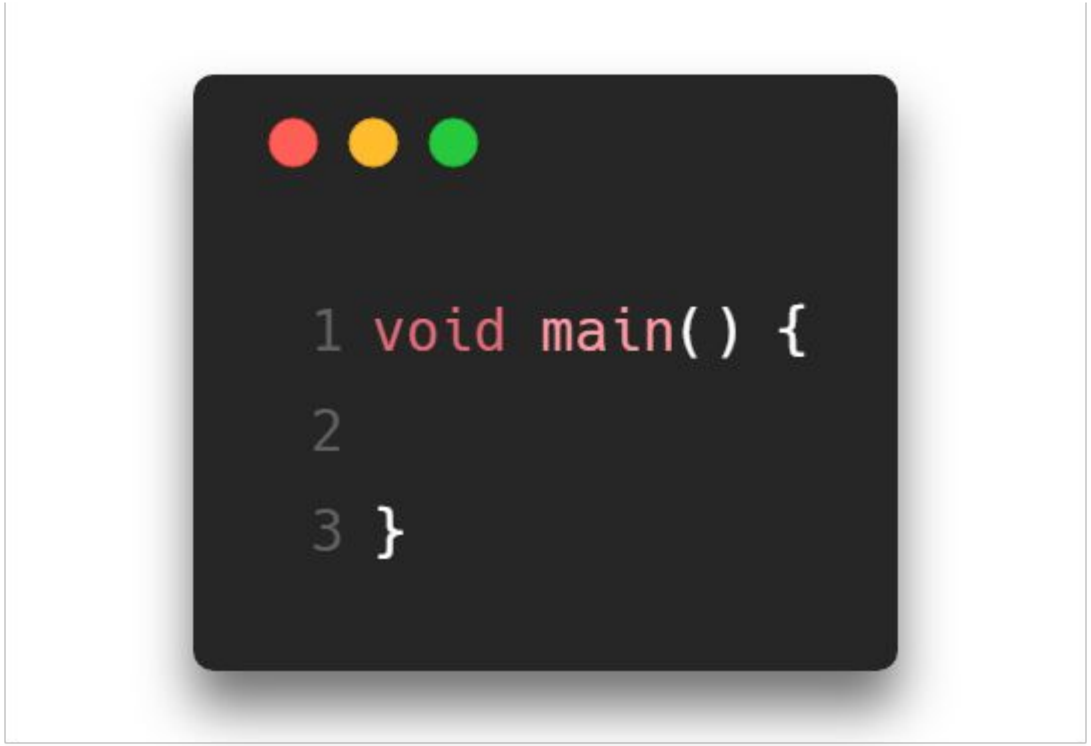
- Android
- iOS

You do not need any other declarative layout language like XML in Android, to create layouts in Flutter

Dart Codelab

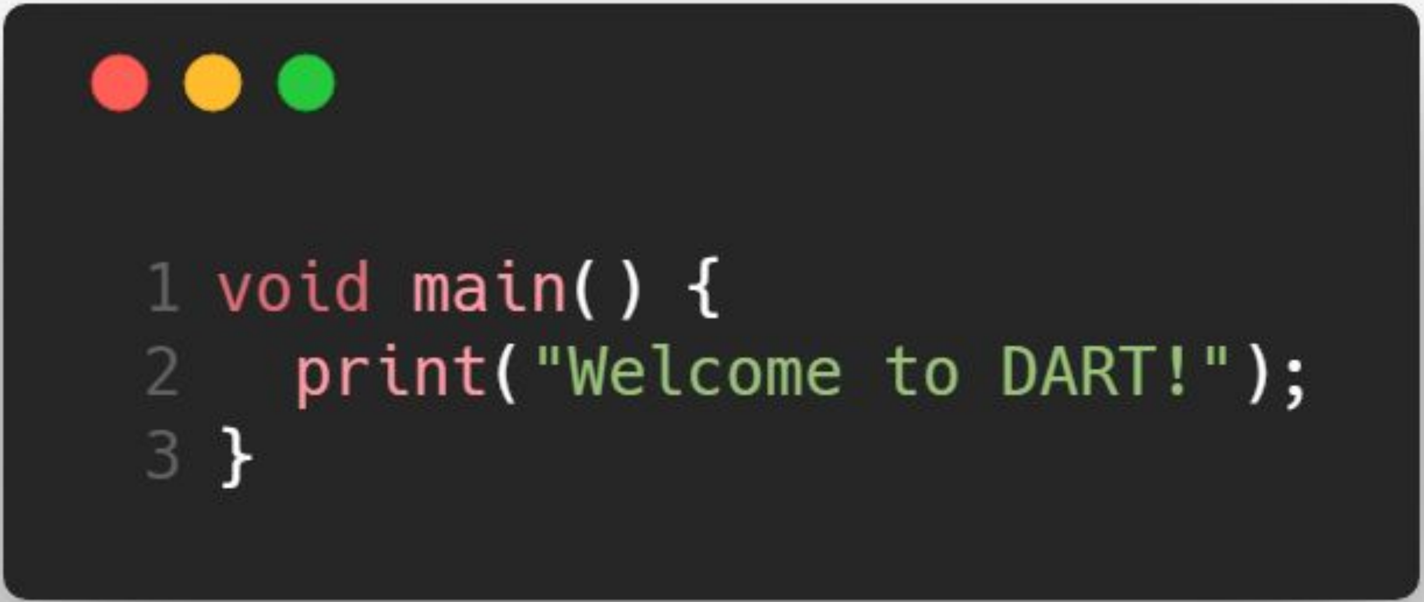
https://bit.ly/dart-lab

- Dart files should have **.dart** extensions
- Every program starts with a **main** function. Entry point to your Dart application.



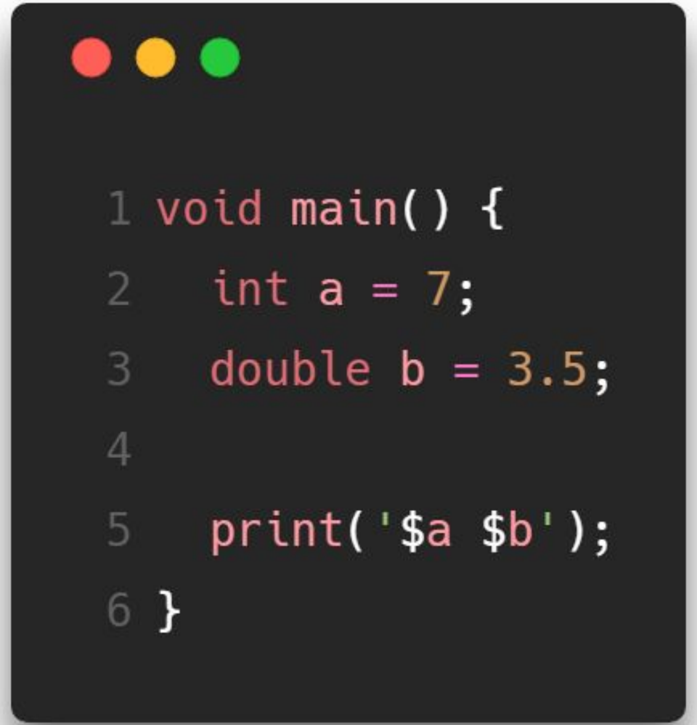
```dart
void main() {

}
```

Print to console

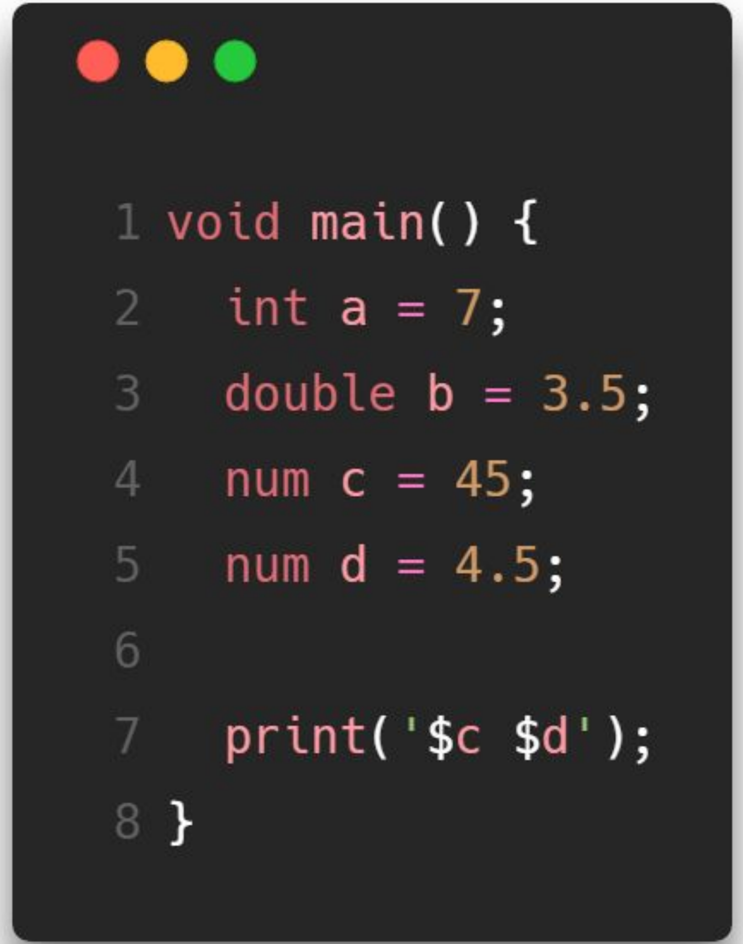


```dart
void main() {
  print("Welcome to DART!");
}
```

Data Type : Numbers

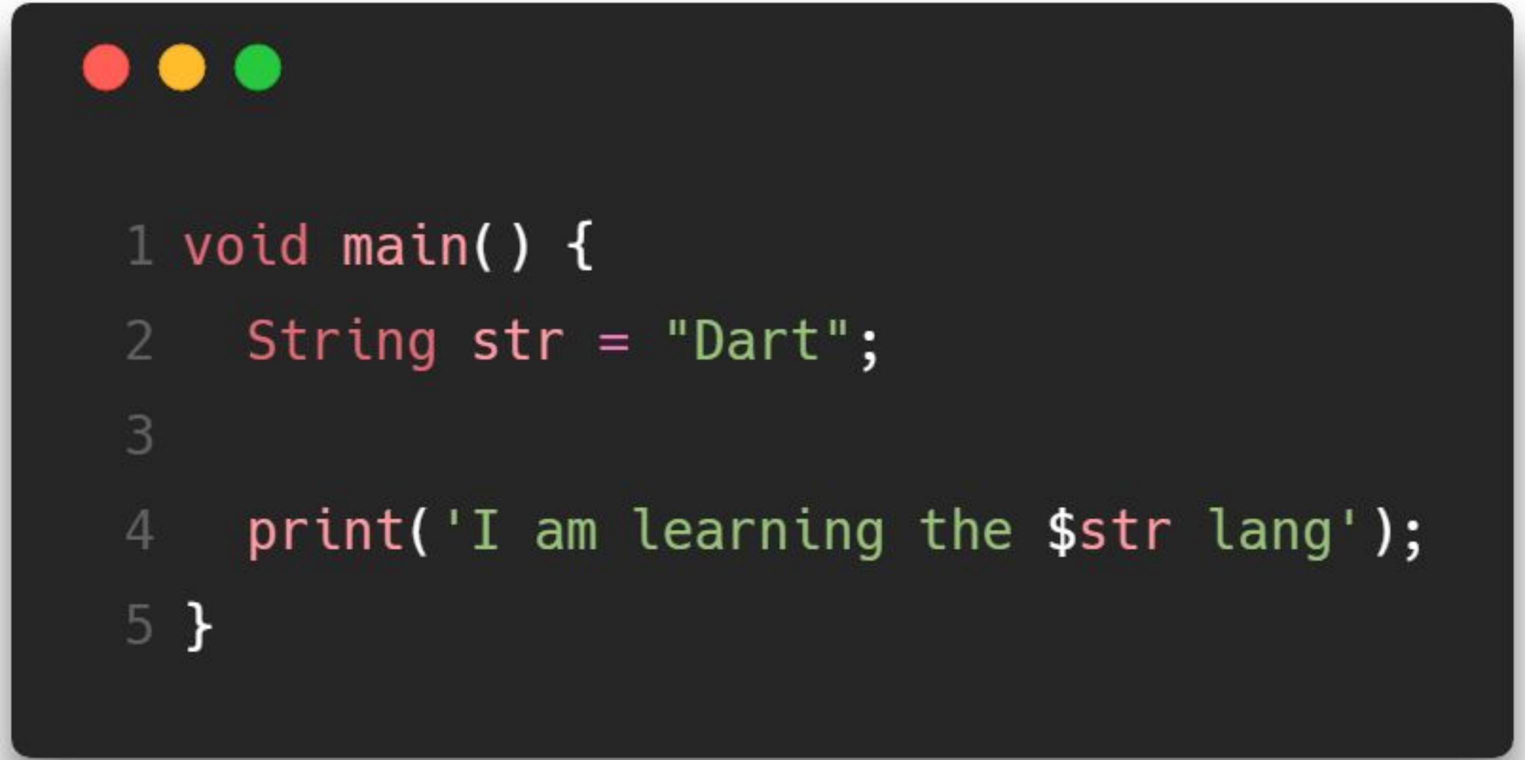


```
void main() {
  int a = 7;
  double b = 3.5;

  print('$a $b');
}
```



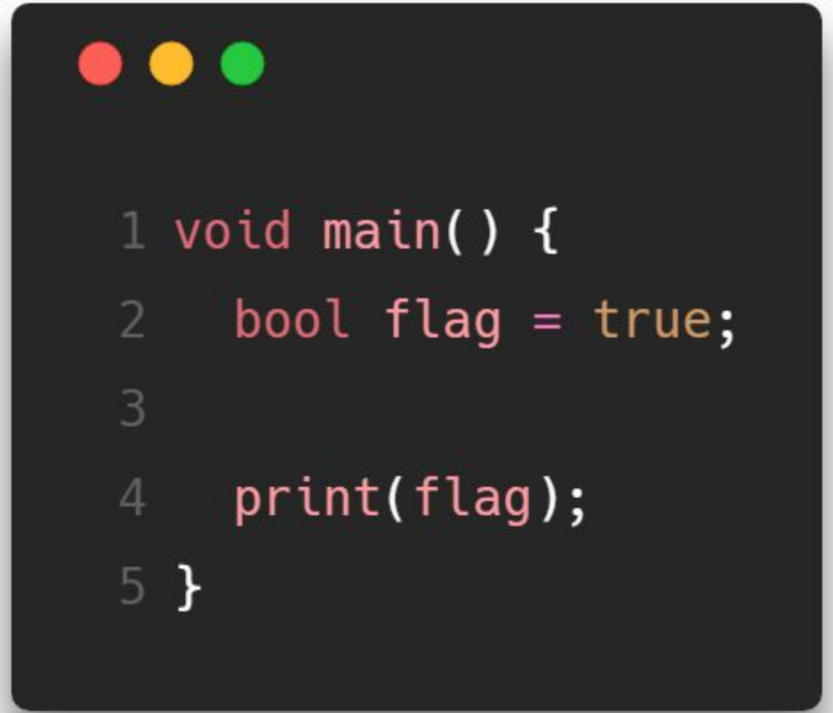
```
void main() {
  int a = 7;
  double b = 3.5;
  num c = 45;
  num d = 4.5;

  print('$c $d');
}
```

Datatype: Strings

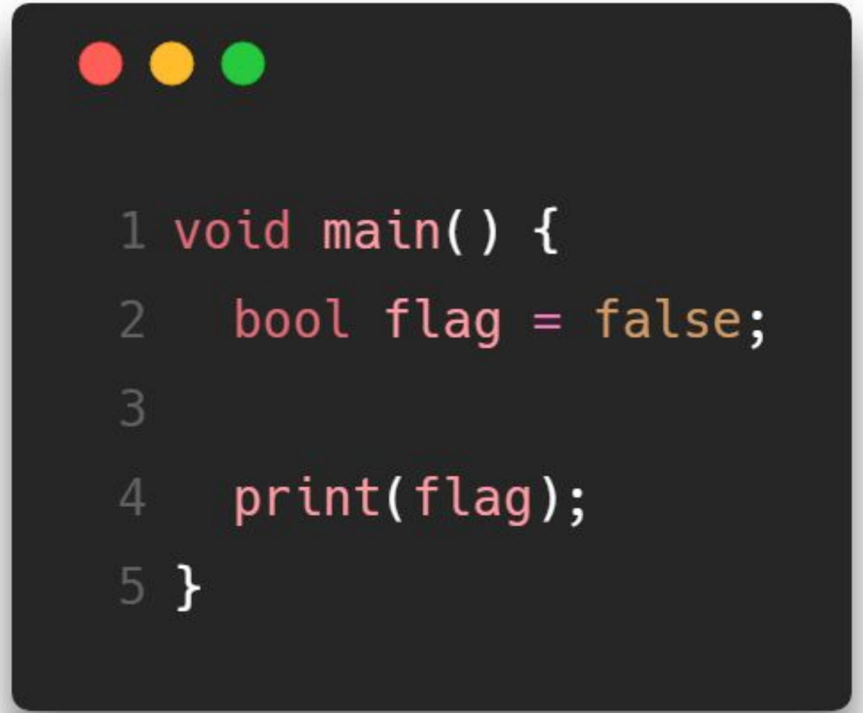


```dart
void main() {
  String str = "Dart";

  print('I am learning the $str lang');
}
```

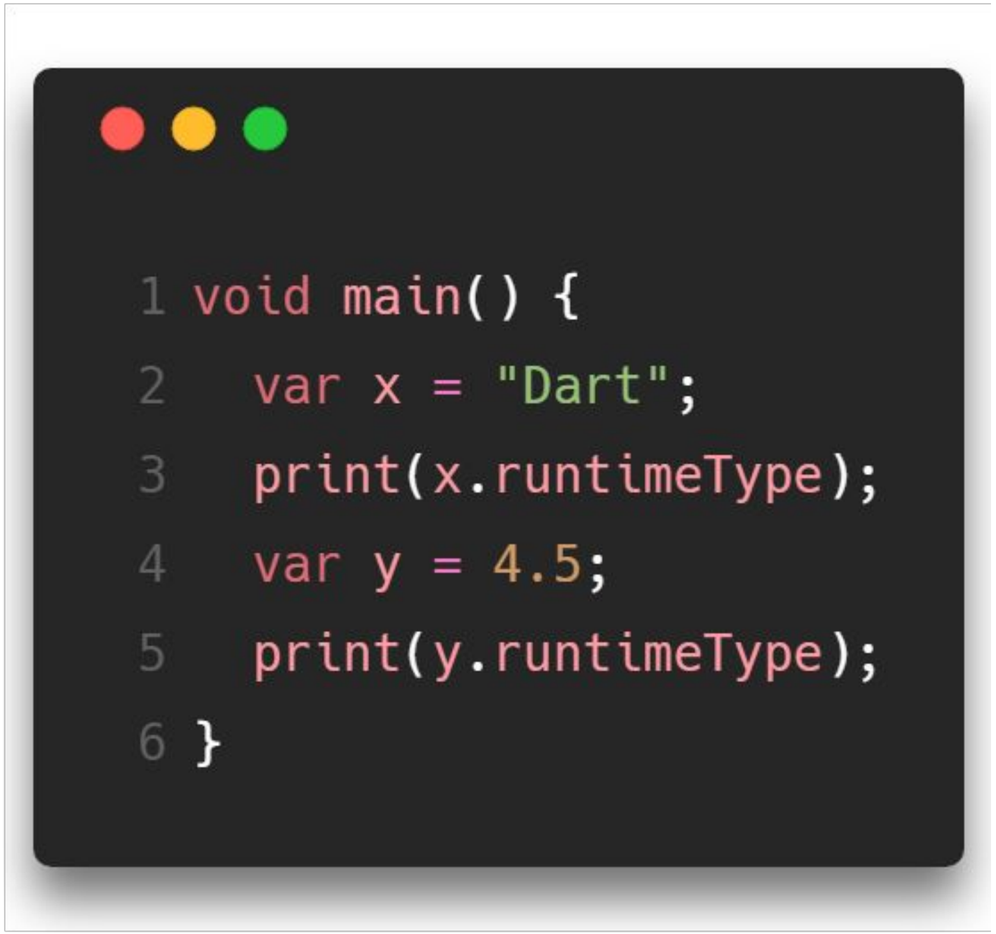
Datatype: Booleans



```
void main() {
  bool flag = true;

  print(flag);
}
```



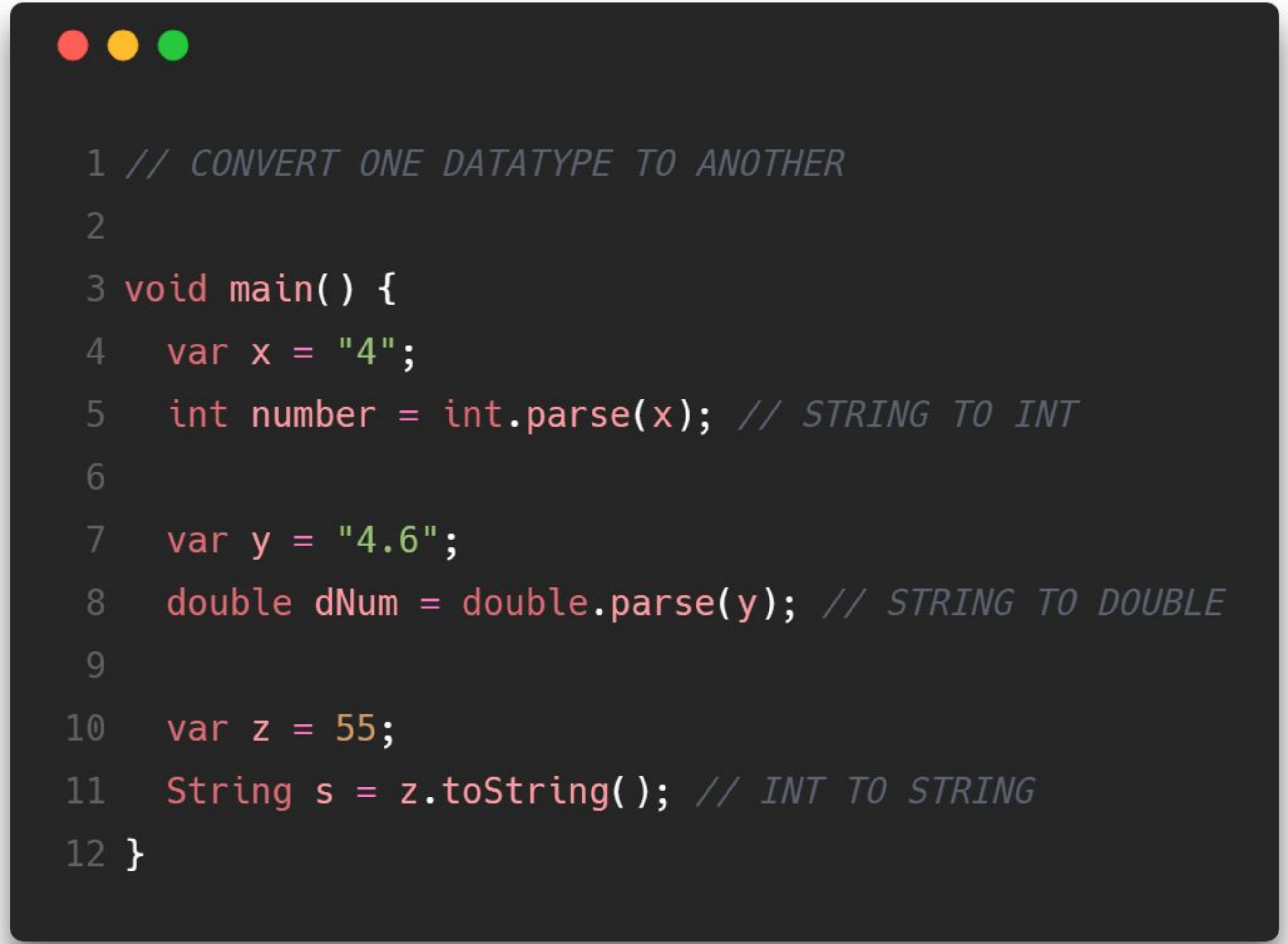
```
void main() {
  bool flag = false;

  print(flag);
}
```



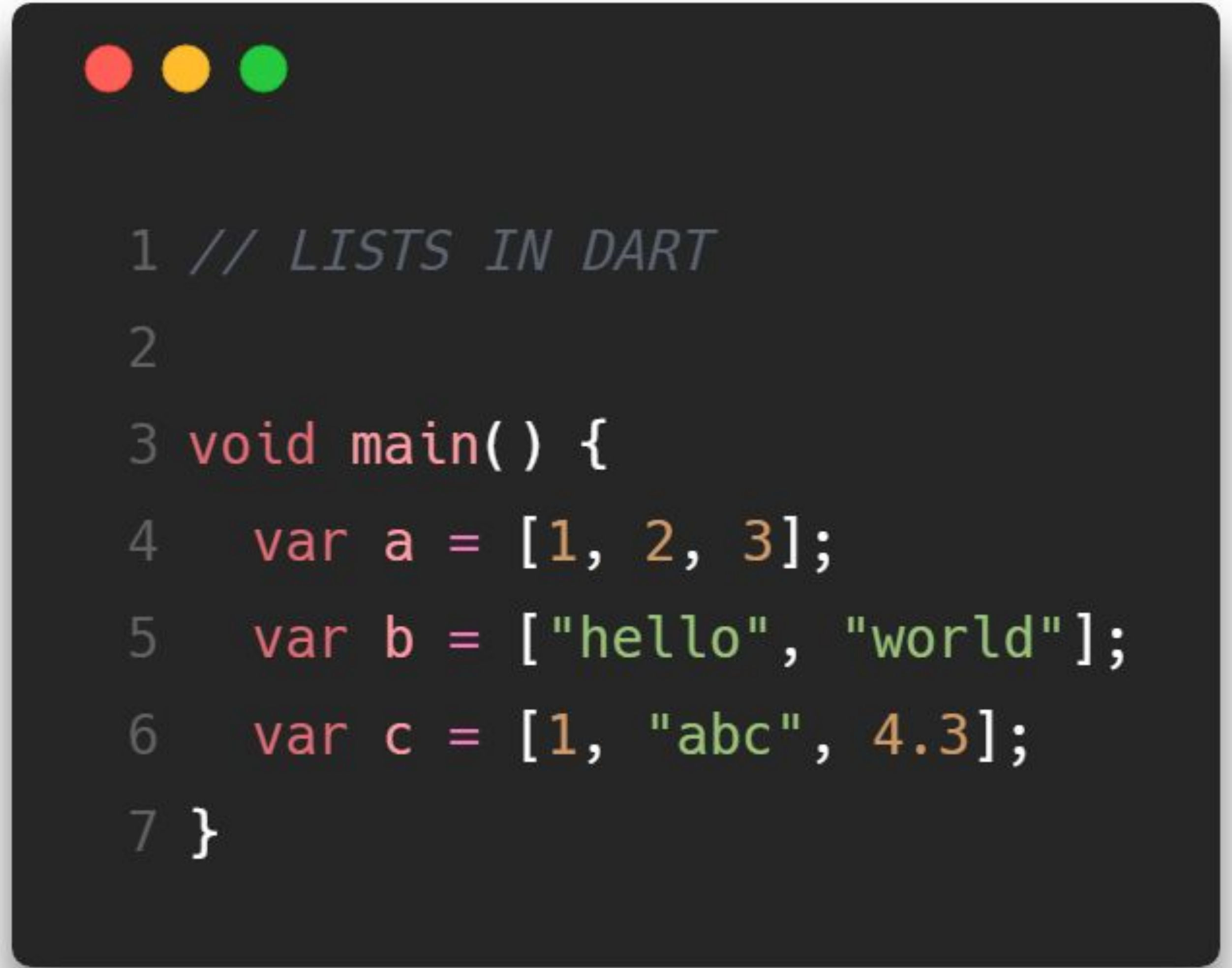
```
void main() {
  var x = "Dart";
  print(x.runtimeType);
  var y = 4.5;
  print(y.runtimeType);
}
```

var is a keyword that can declare any data type

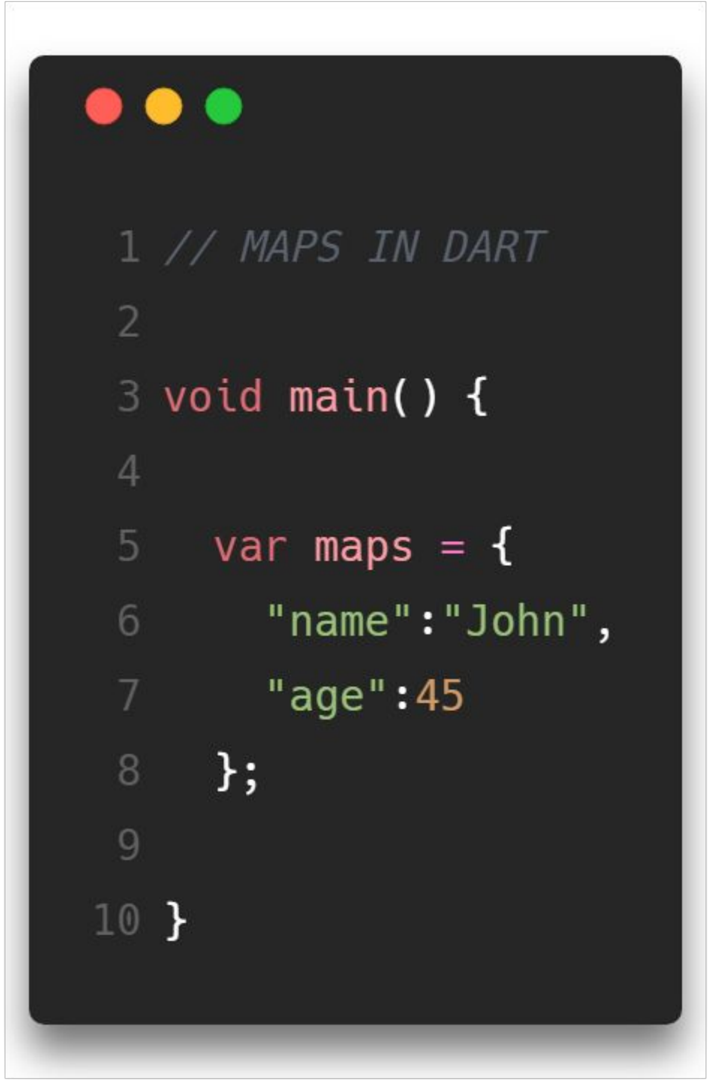
runtimeType prints the type of the variable declared



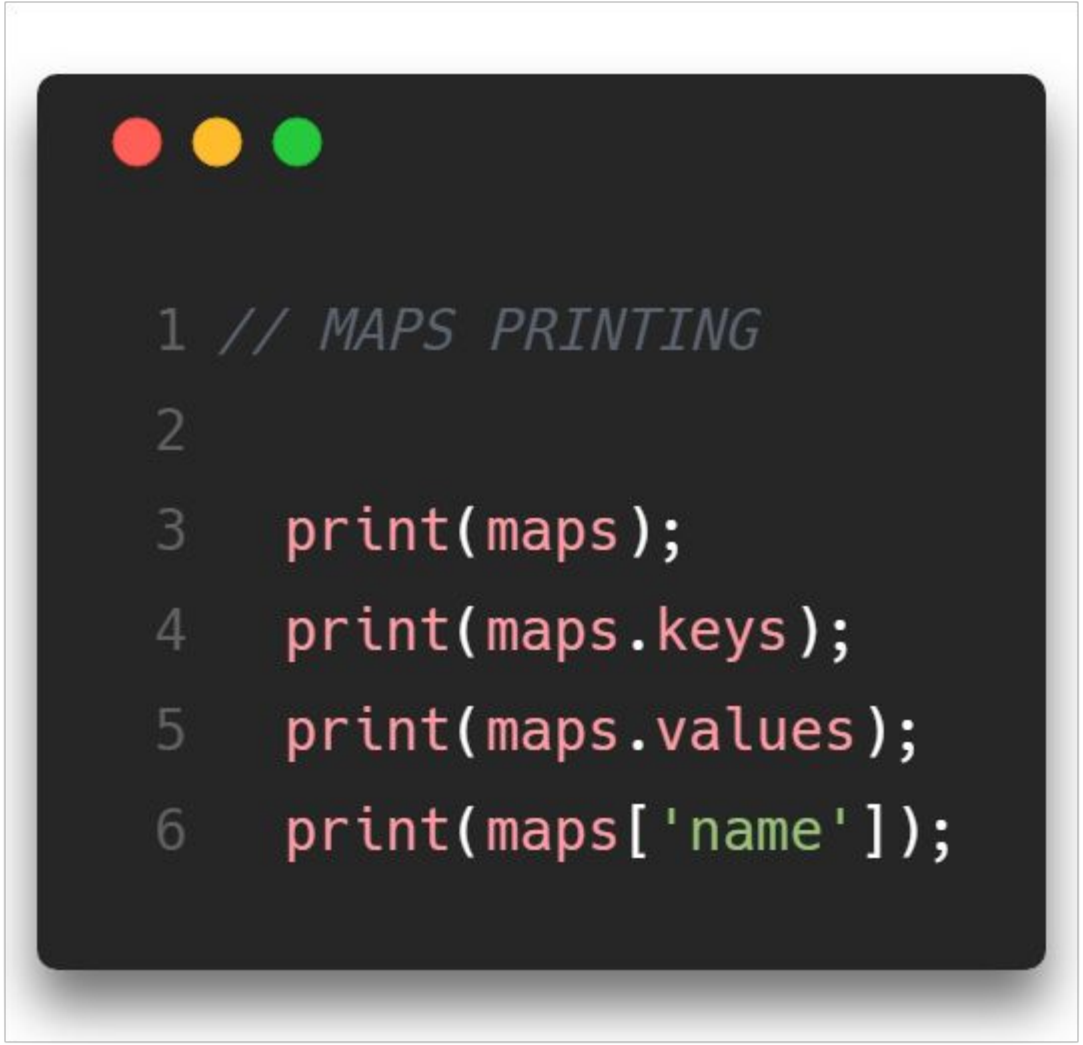
```
// CONVERT ONE DATATYPE TO ANOTHER

void main() {
  var x = "4";
  int number = int.parse(x); // STRING TO INT

  var y = "4.6";
  double dNum = double.parse(y); // STRING TO DOUBLE

  var z = 55;
  String s = z.toString(); // INT TO STRING
}
```



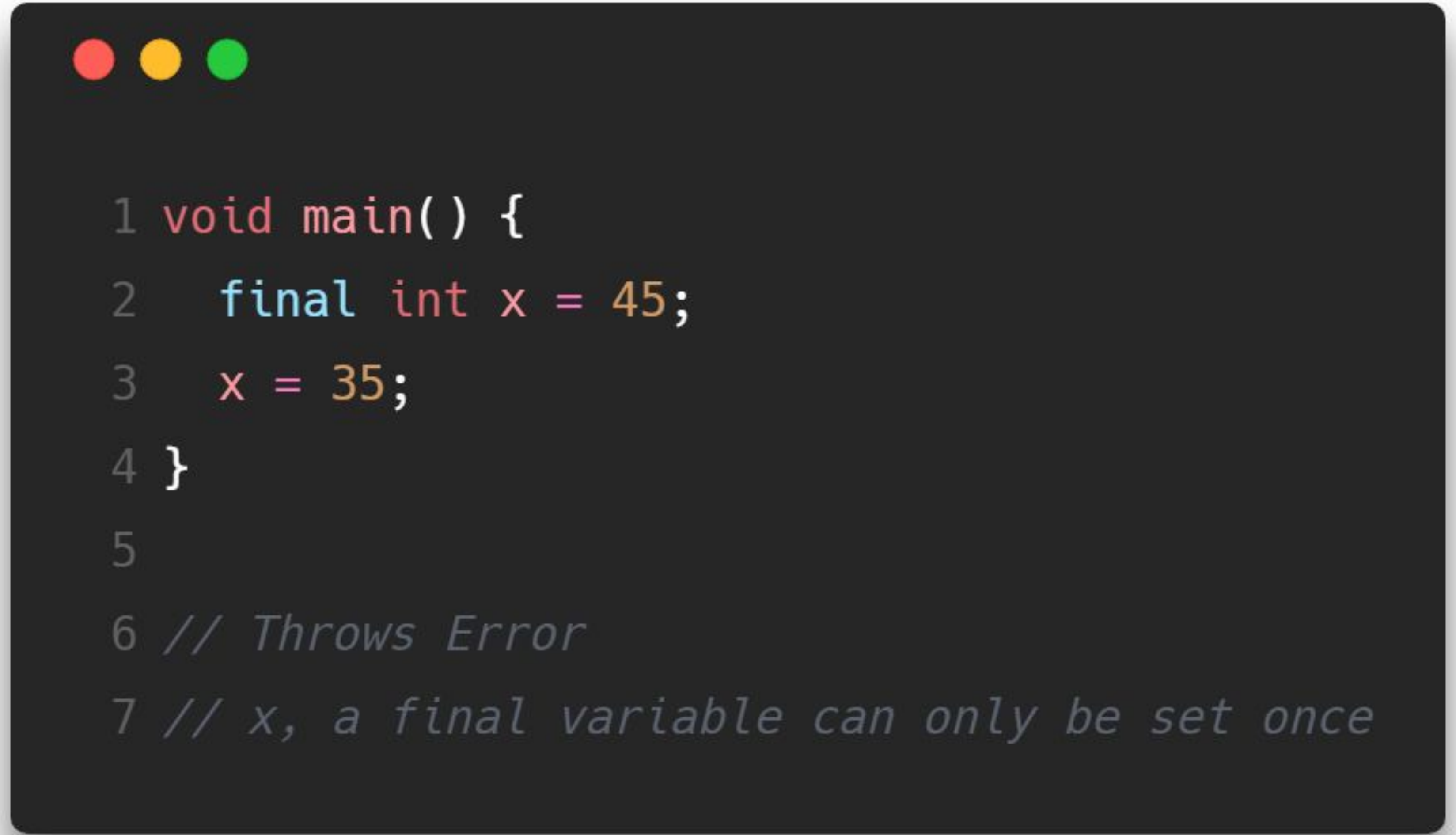
```dart
// LISTS IN DART

void main() {
  var a = [1, 2, 3];
  var b = ["hello", "world"];
  var c = [1, "abc", 4.3];
}
```



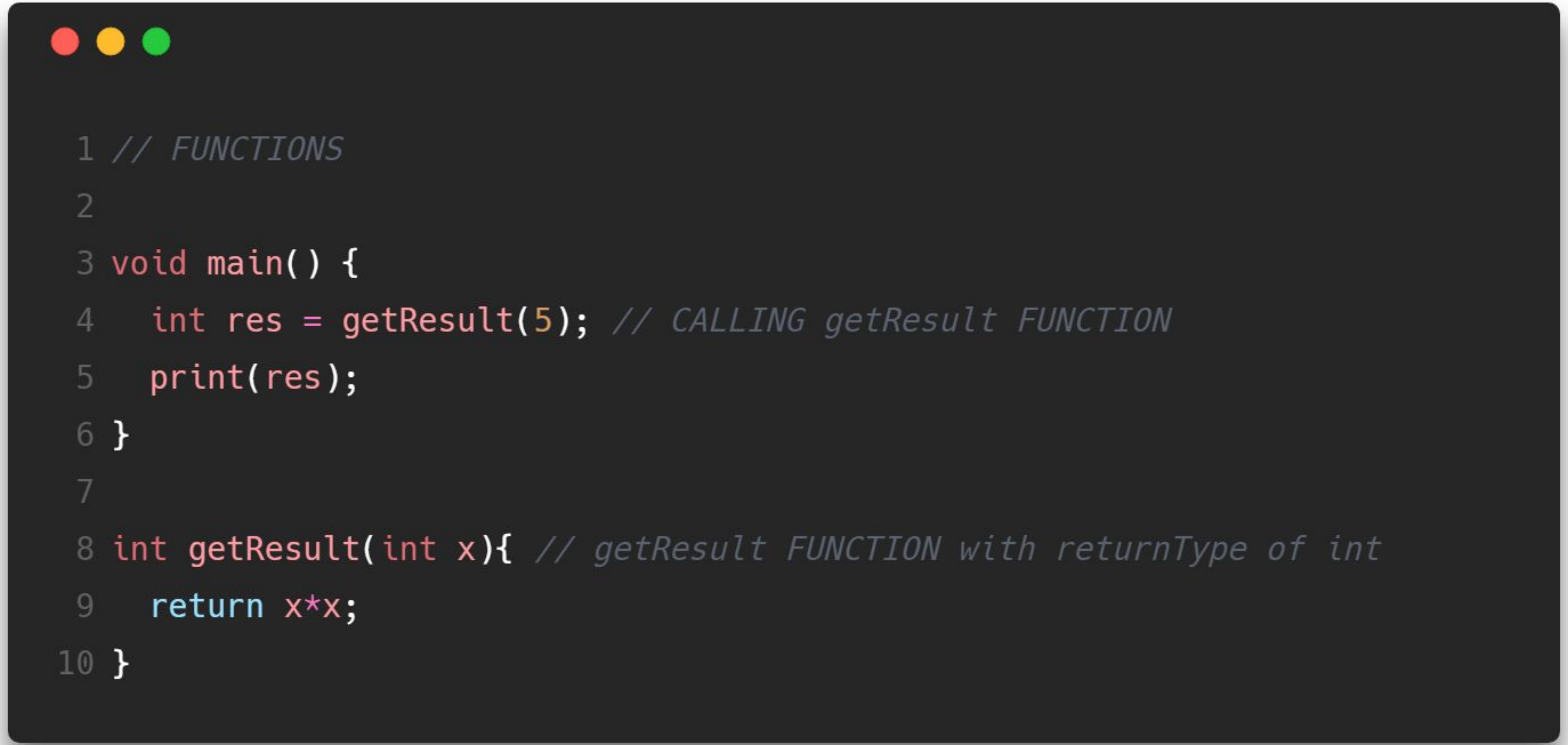
```
// MAPS IN DART

void main() {

  var maps = {
    "name":"John",
    "age":45
  };

}
```



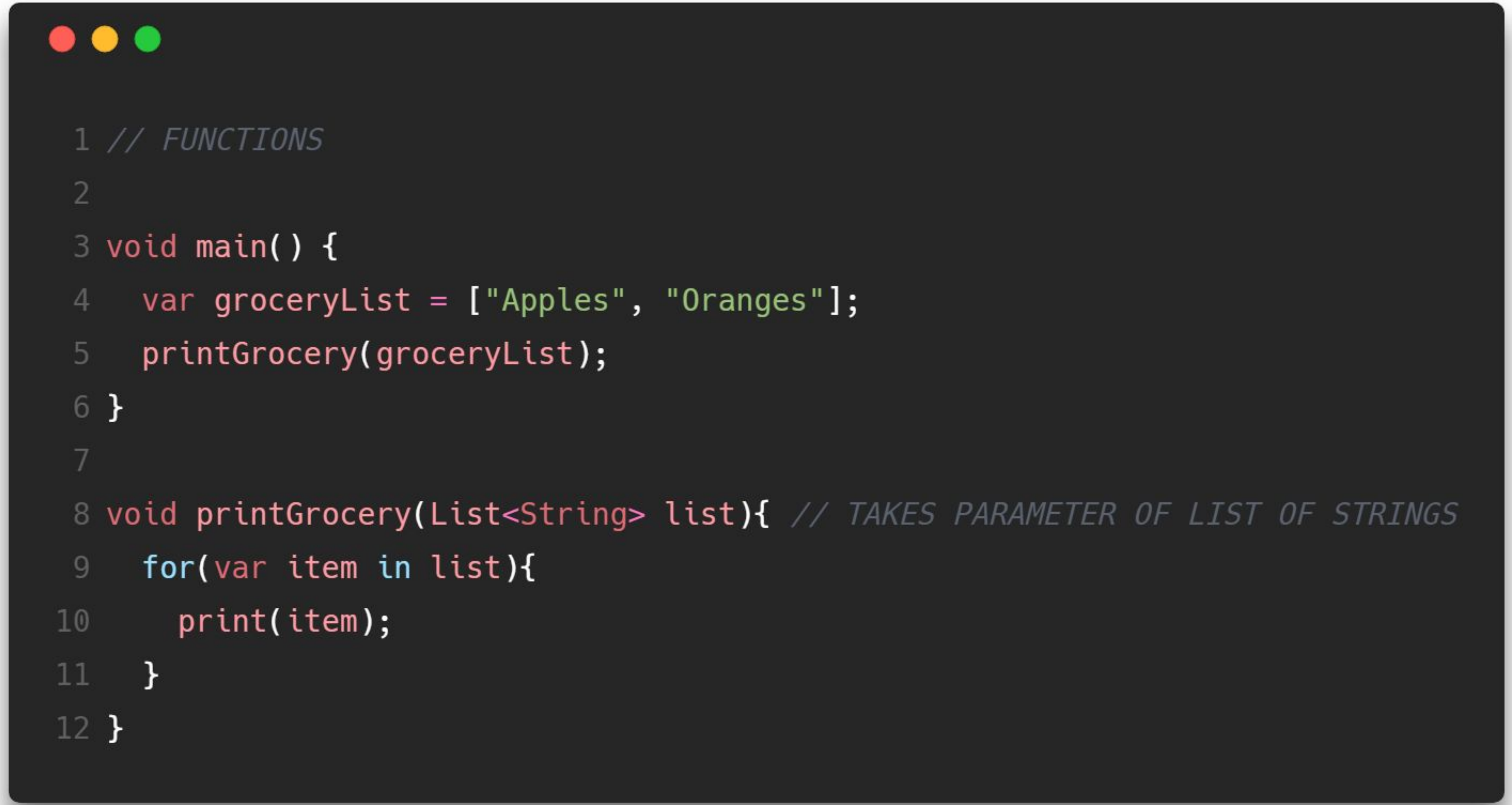
```
// MAPS PRINTING

  print(maps);
  print(maps.keys);
  print(maps.values);
  print(maps['name']);
```



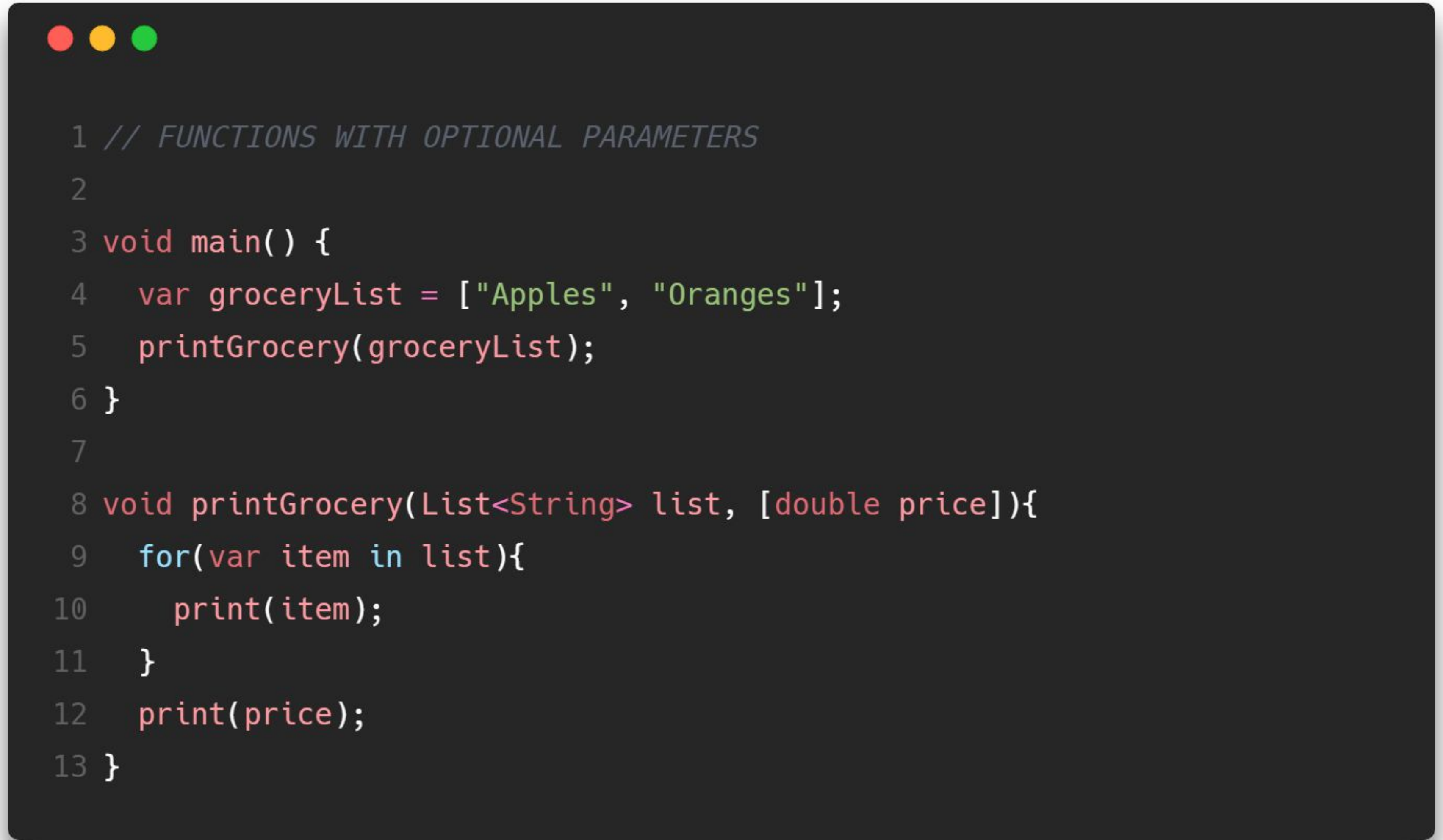
```
void main() {
  final int x = 45;
  x = 35;
}

// Throws Error
// x, a final variable can only be set once
```



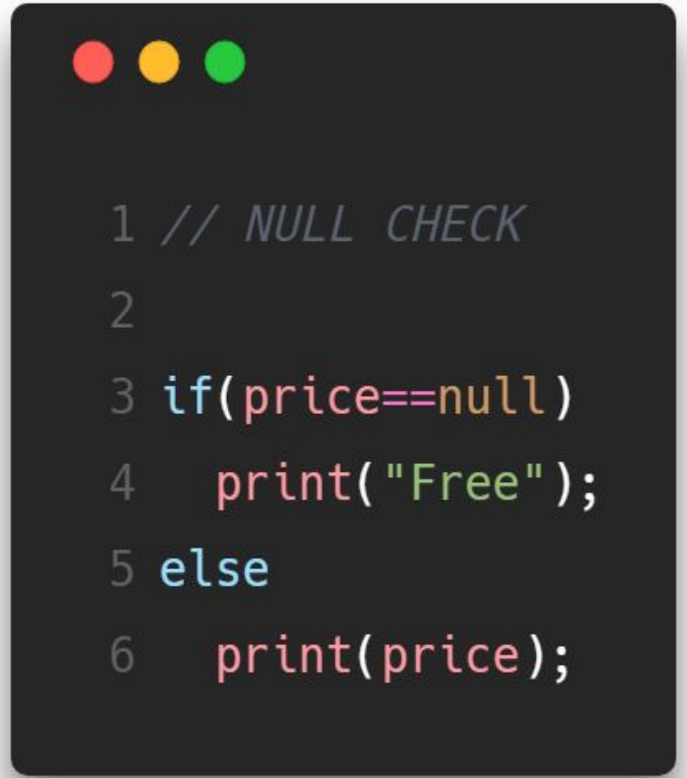
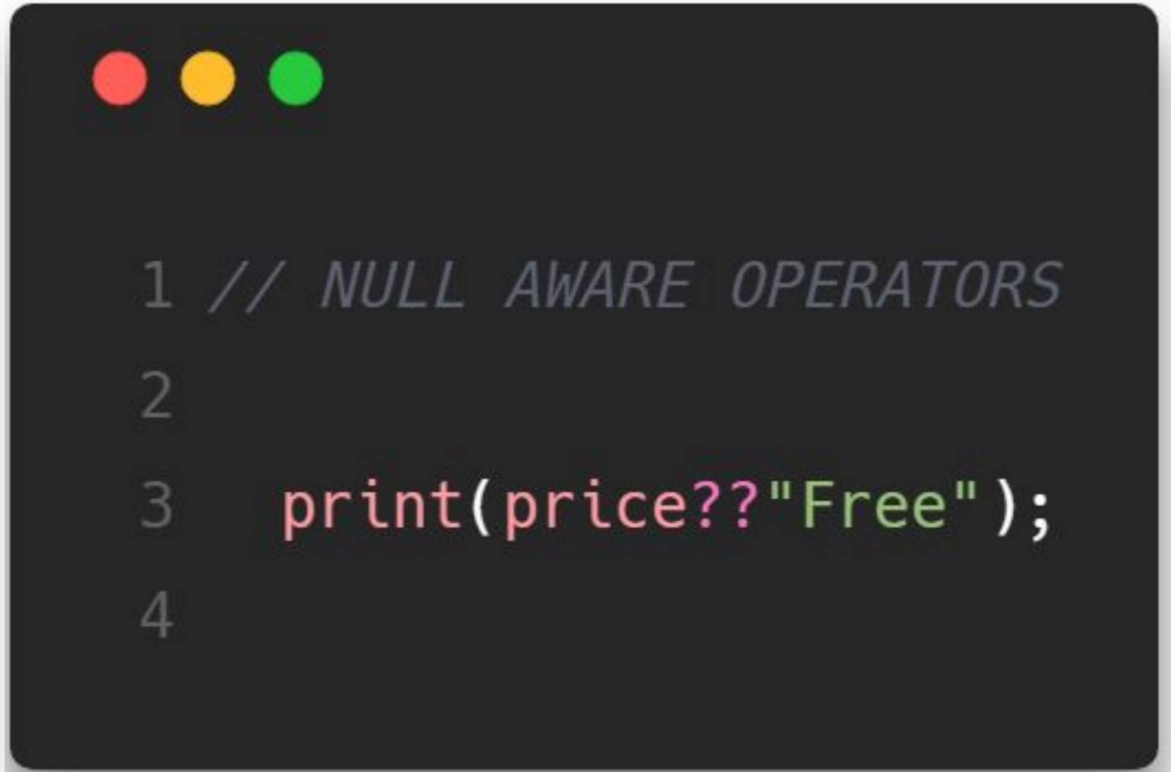
```dart
// FUNCTIONS

void main() {
  int res = getResult(5); // CALLING getResult FUNCTION
  print(res);
}

int getResult(int x){ // getResult FUNCTION with returnType of int
  return x*x;
}
```

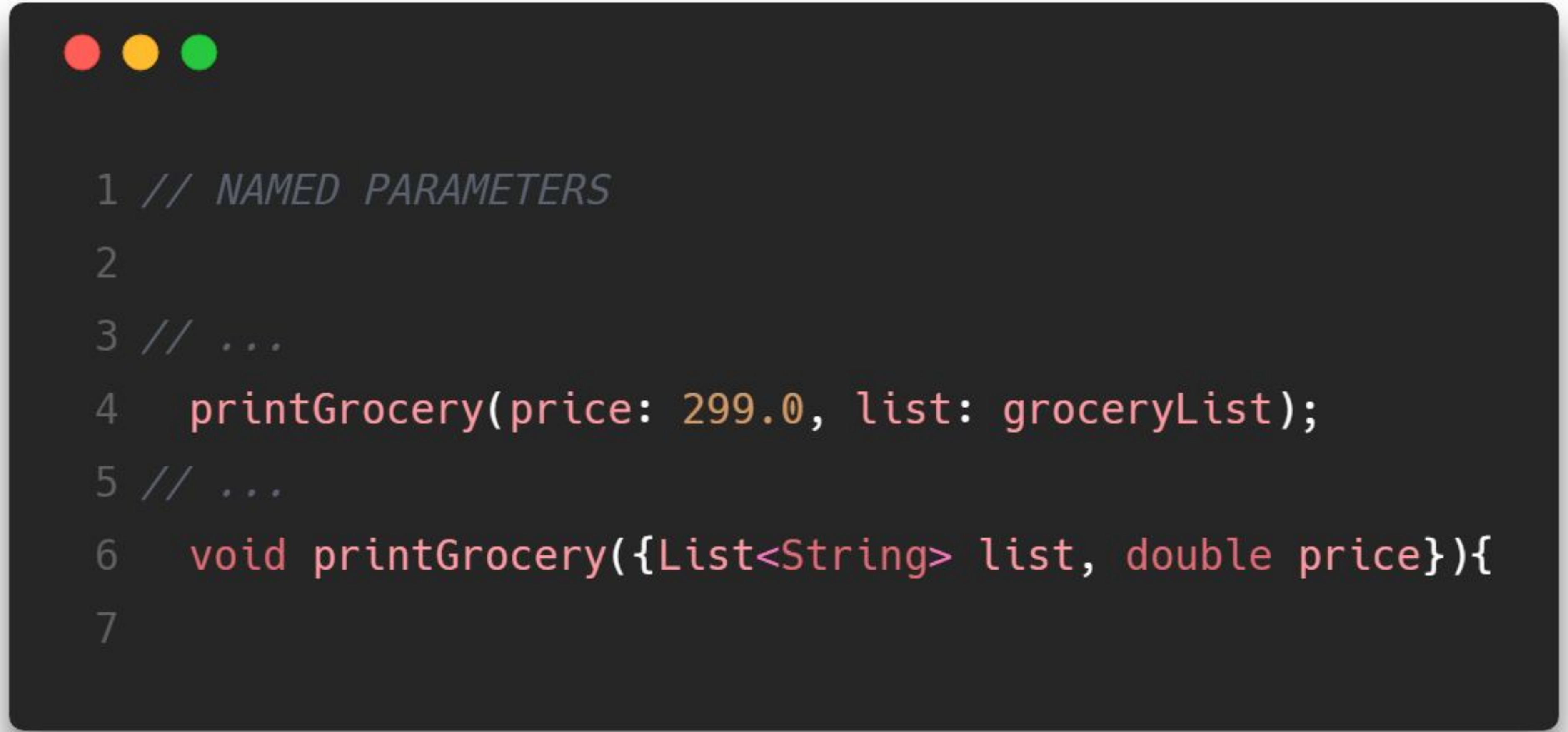


```dart
// FUNCTIONS

void main() {
  var groceryList = ["Apples", "Oranges"];
  printGrocery(groceryList);
}

void printGrocery(List<String> list){ // TAKES PARAMETER OF LIST OF STRINGS
  for(var item in list){
    print(item);
  }
}
```

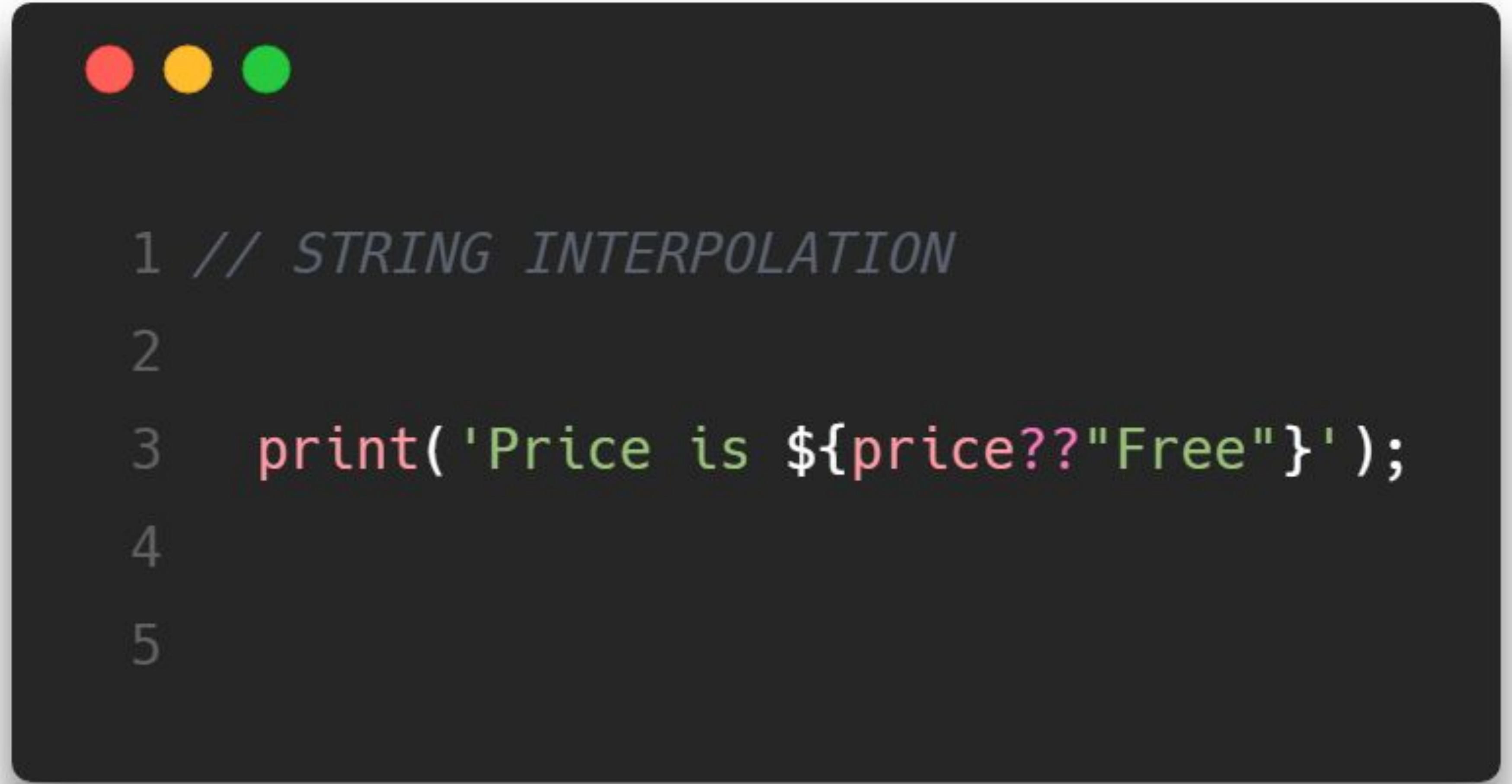


```
// FUNCTIONS WITH OPTIONAL PARAMETERS

void main() {
  var groceryList = ["Apples", "Oranges"];
  printGrocery(groceryList);
}

void printGrocery(List<String> list, [double price]){
  for(var item in list){
    print(item);
  }
  print(price);
}
```

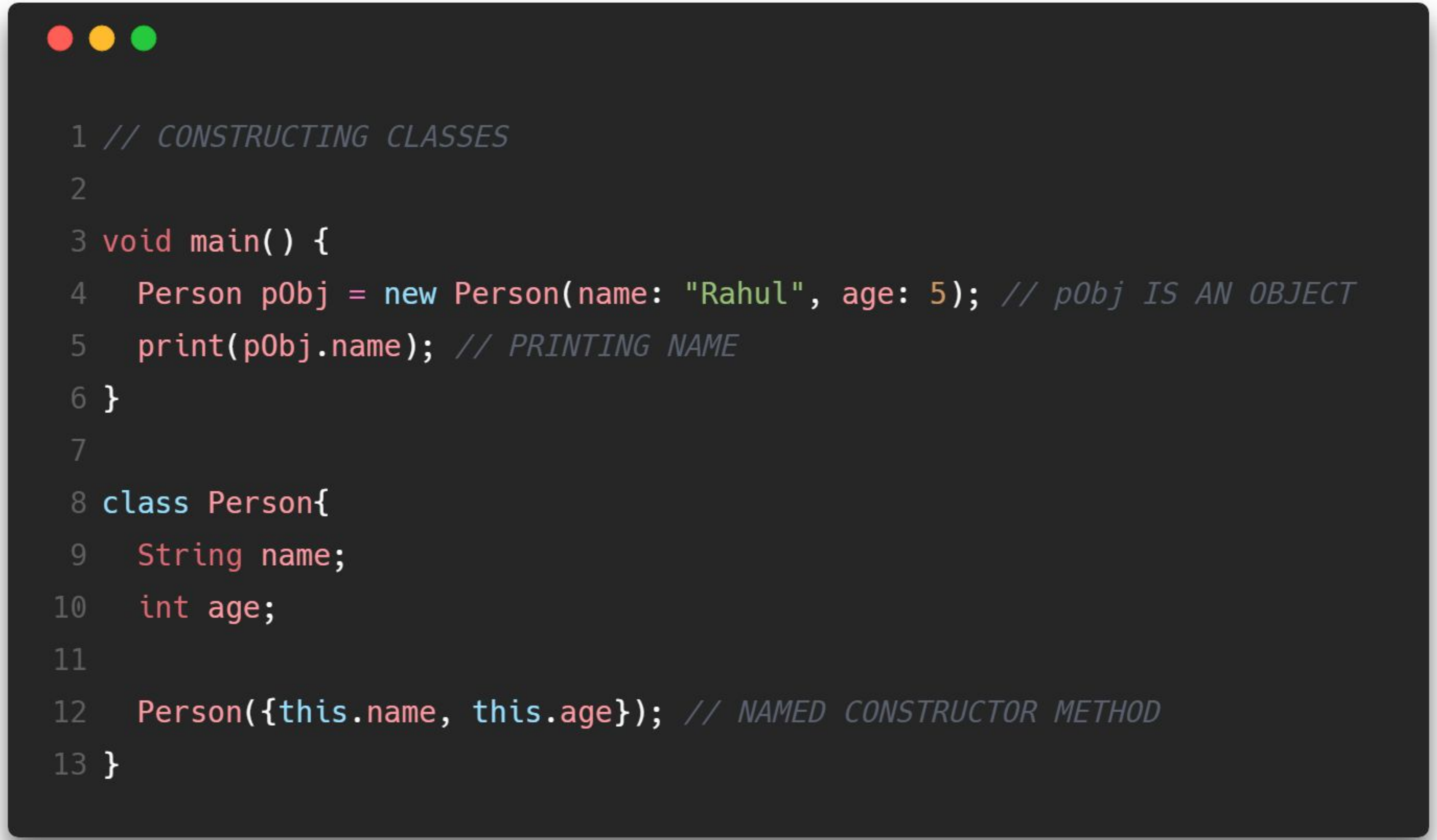

```
// NULL CHECK

if(price==null)
  print("Free");
else
  print(price);
```

```
// NULL AWARE OPERATORS

  print(price??"Free");

```



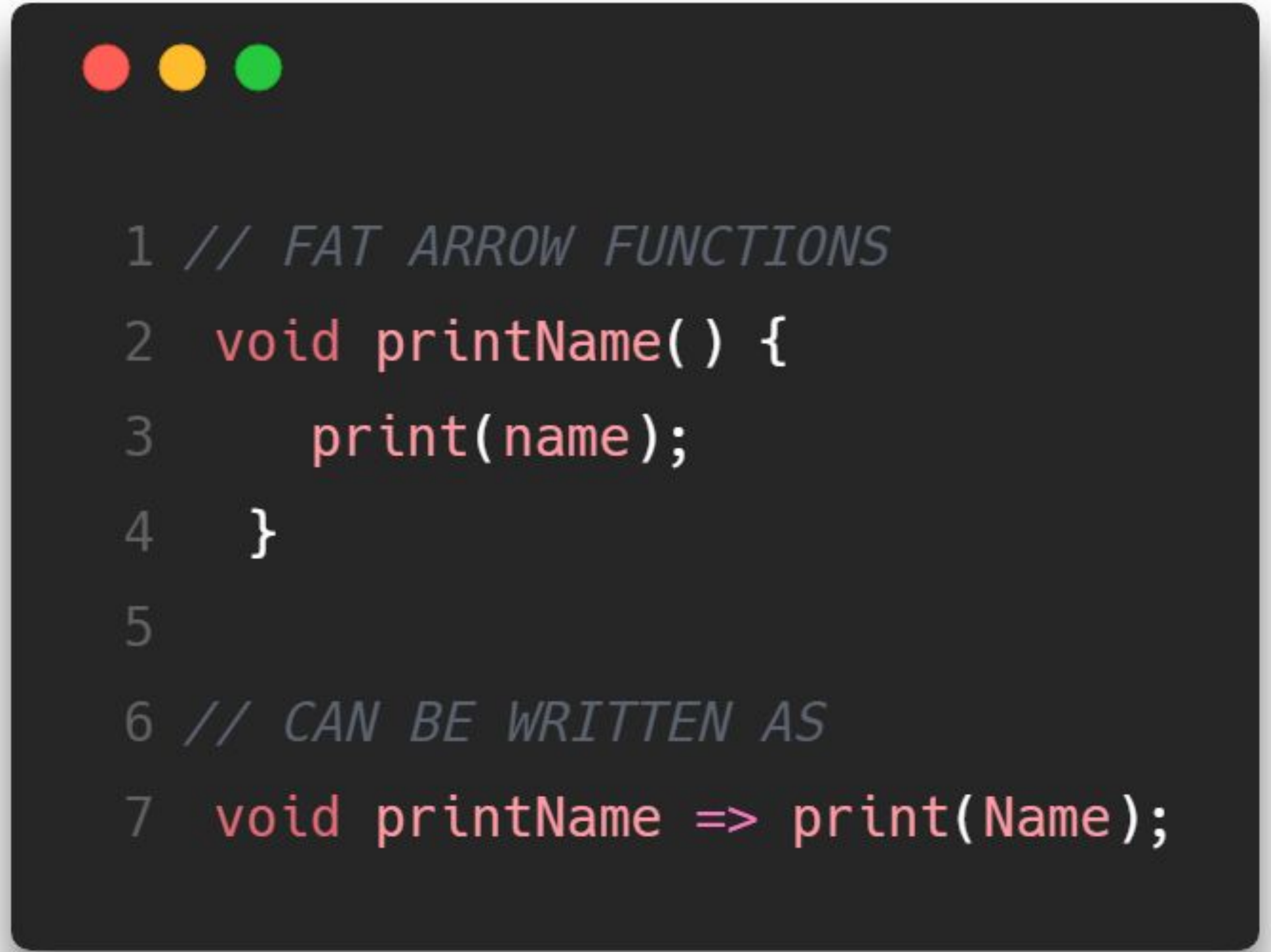
```dart
// NAMED PARAMETERS

// ...
  printGrocery(price: 299.0, list: groceryList);
// ...
  void printGrocery({List<String> list, double price}){

```



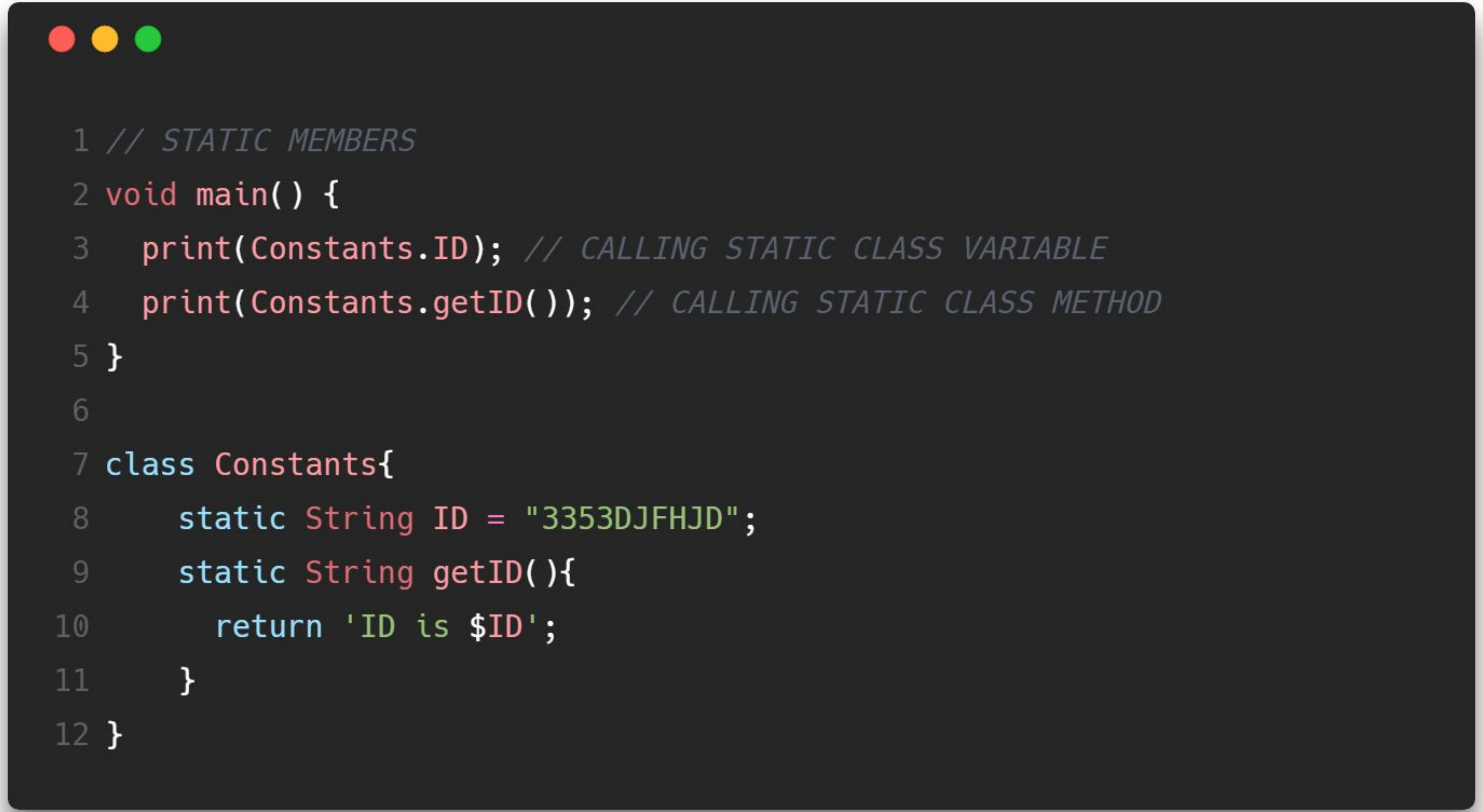
```
// STRING INTERPOLATION

  print('Price is ${price??"Free"}');


```



```dart
// CONSTRUCTING CLASSES

void main() {
  Person pObj = new Person(name: "Rahul", age: 5); // pObj IS AN OBJECT
  print(pObj.name); // PRINTING NAME
}

class Person{
  String name;
  int age;

  Person({this.name, this.age}); // NAMED CONSTRUCTOR METHOD
}
```



```
// FAT ARROW FUNCTIONS
 void printName() {
    print(name);
  }

// CAN BE WRITTEN AS
 void printName => print(Name);
```



```dart
// STATIC MEMBERS
void main() {
  print(Constants.ID); // CALLING STATIC CLASS VARIABLE
  print(Constants.getID()); // CALLING STATIC CLASS METHOD
}

class Constants{
    static String ID = "3353DJFHJD";
    static String getID(){
      return 'ID is $ID';
    }
}
```

More Dart Resources

https://www.youtube.com/watch?v=5rtujDjt50I&list=PLlxmoA0rQ-LyHW9voBdNo4gEEIh0SjG-q

https://codelabs.developers.google.com/codelabs/from-java-to-dart

https://hackr.io/tutorials/learn-dart