CONSTANT CONSTRUCTOR IN DART

Constant Constructor In Dart

Constant constructor is a constructor that creates a constant object. A constant object is an object whose value cannot be changed. A constant constructor is declared using the keyword **const**.

> Info
>
> **Note**: **Constant Constructor** is used to create a object whose value cannot be changed. It Improves the performance of the program.

Rule For Declaring Constant Constructor In Dart

- All properties of the class must be final.
- It does not have any body.
- Only class containing **const** constructor is initialized using the **const** keyword.

Example 1: Constant Constructor In Dart

In this example below, there is a class **Point** with two final properties: **x** and **y**. The class also has a constant constructor that initializes the two properties. The class also has a method called **display**, which prints out the values of the two properties.

```
class Point {
  final int x;
  final int y;

  const Point(this.x, this.y);
}

void main() {
```

```
  // p1 and p2 has the same hash code.
  Point p1 = const Point(1, 2);
  print("The p1 hash code is: ${p1.hashCode}");

  Point p2 = const Point(1, 2);
  print("The p2 hash code is: ${p2.hashCode}");
  // without using const
  // this has different hash code.
  Point p3 = Point(2, 2);
  print("The p3 hash code is: ${p3.hashCode}");

  Point p4 = Point(2, 2);
  print("The p4 hash code is: ${p4.hashCode}");
}
```

Save this Snippet

› Show Output

Run Online

Info

Note: Here p1 and p2 has the same hash code. This is because p1 and p2 are constant objects. The hash code of a constant object is the same. This is because the hash code of a constant object is computed at compile time. The hash code of a non-constant object is computed at run time. This is why p3 and p4 have different hash code.

Example 2: Constant Constructor In Dart

In this example below, there is a class **Student** with three properties: **name**, **age**, and **rollNumber**. The class has one constant constructor. The constructor is used to initialize the values of the three properties. We also have an object of the class **Student** called **student**.

```
class Student {
  final String? name;
  final int? age;
  final int? rollNumber;

  // Constant Constructor
  const Student({this.name, this.age, this.rollNumber});
}

void main() {
```

```
  // Here student is object of Student.
  const Student student = Student(name: "John", age: 20, rollNumber: 1);
  print("Name: ${student.name}");
  print("Age: ${student.age}");
  print("Roll Number: ${student.rollNumber}");
}
```

Save this Snippet

Show Output

Run Online

Example 3: Constant Constructor With Named Parameters In Dart

In this example below, there is a class **Car** with three properties: **name**, **model**, and **price**. The class has one constructor. The constructor is used to initialize the values of the three properties. We also have an object of the class **Car** called **car**.

```
class Car {
  final String? name;
  final String? model;
  final int? price;

  // Constant Constructor
  const Car({this.name, this.model, this.price});
}

void main() {
  // Here car is object of class Car.
  const Car car = Car(name: "BMW", model: "X5", price: 50000);
  print("Name: ${car.name}");
  print("Model: ${car.model}");
  print("Price: ${car.price}");
}
```

Save this Snippet

Show Output

Run Online

Benefits Of Constant Constructor In Dart

- Improves the performance of the program.

Challenge

Create a class **Customer** with three properties: **name**, **age**, and **phone**. The class should have one constant constructor. The constructor should initialize the values of the three properties. Create an object of the class **Customer** and print the values of the three properties.

Video

Watch our video on constant constructor in Dart.