NAMED CONSTRUCTOR IN DART

Named Constructor In Dart

In most programming languages like java, c++, c#, etc., we can create multiple constructors with the same name. But in Dart, this is not possible. Well, there is a way. We can create multiple constructors with the same name using **named constructors**.

> **Info**
>
> **Note**: Named constructors improves code readability. It is useful when you want to create multiple constructors with the same name.

Example 1: Named Constructor In Dart

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

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

  // Default Constructor
  Student() {
    print("This is a default constructor");
  }

  // Named Constructor
  Student.namedConstructor(String name, int age, int rollNumber) {
    this.name = name;
    this.age = age;
    this.rollNumber = rollNumber;
```

```
  }
}

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

Save this Snippet

Show Output

Run Online

Example 2: Named Constructor In Dart

In this example below, there is class **Mobile** with three properties **name**, **color**, and **price**. The class has one method **display** which prints out the values of the three properties. We also have an object of the class **Mobile** called **mobile**. There is also constructor **Mobile** which takes all the three properties as parameters. Named constructor **Mobile.namedConstructor** is used to create an object of the class **Mobile** with name, color and optional price. The default value of the price is 0. If the price is not passed, then the default value is used.

```
class Mobile {
  String? name;
  String? color;
  int? price;

  Mobile(this.name, this.color, this.price);
  // here Mobile() is a named constructor
  Mobile.namedConstructor(this.name, this.color, [this.price = 0]);

  void displayMobileDetails() {
    print("Mobile name: $name.");
    print("Mobile color: $color.");
    print("Mobile price: $price");
  }
}

void main() {
  var mobile1 = Mobile("Samsung", "Black", 20000);
  mobile1.displayMobileDetails();
  var mobile2 = Mobile.namedConstructor("Apple", "White");
```

```
  mobile2.displayMobileDetails();
}
```

Save this Snippet

Show Output

Run Online

Example 3: Named Constructor In Dart

In this example below, there is a class **Animal** with two properties **name** and **age**. The class has three constructors. The first constructor is a default constructor. The second and third constructors are named constructors. The second constructor is used to initialize the values of name and age, and the third constructor is used to initialize the value of name only. We also have an object of the class **Animal** called **animal**.

```
class Animal {
  String? name;
  int? age;

  // Default Constructor
  Animal() {
    print("This is a default constructor");
  }

  // Named Constructor
  Animal.namedConstructor(String name, int age) {
    this.name = name;
    this.age = age;
  }

  // Named Constructor
  Animal.namedConstructor2(String name) {
    this.name = name;
  }
}
void main(){
  // Here animal is object of class Animal.
  Animal animal = Animal.namedConstructor("Dog", 5);
  print("Name: ${animal.name}");
  print("Age: ${animal.age}");

  Animal animal2 = Animal.namedConstructor2("Cat");
  print("Name: ${animal2.name}");
}
```

Show Output

Run Online

Example 4: Real Life Example Of Named Constructor In Dart

In this example below, there is a class **Person** with two properties **name** and **age**. The class has three constructors. The first is a parameterized constructor which takes two parameters **name** and **age**. The second and third constructors are named constructors. Second constructor fromJson is used to create an object of the class **Person** from a JSON. The third fromJsonString is used to create an object of the class **Person** from a JSON string. We also have an object of the class **Person** called **person**.

```
import 'dart:convert';

class Person {
  String? name;
  int? age;

  Person(this.name, this.age);

  Person.fromJson(Map<String, dynamic> json) {
    name = json['name'];
    age = json['age'];
  }

  Person.fromJsonString(String jsonString) {
    Map<String, dynamic> json = jsonDecode(jsonString);
    name = json['name'];
    age = json['age'];
  }
}

void main() {
// Here person is object of class Person.
  String jsonString1 = '{"name": "Bishworaj", "age": 25}';
  String jsonString2 = '{"name": "John", "age": 30}';

  Person p1 = Person.fromJsonString(jsonString1);
  print("Person 1 name: ${p1.name}");
  print("Person 1 age: ${p1.age}");

  Person p2 = Person.fromJsonString(jsonString2);
  print("Person 2 name: ${p2.name}");
```

```
  print("Person 2 age: ${p2.age}");
}
```

Save this Snippet

Show Output

Run Online

Challenge

Try to create a class **Car** with three properties **name**, **color**, and **price** and one method **display** which prints out the values of the three properties. Create a constructor, which takes all 3 parameters. Create a named constructor which takes two parameters **name** and **color**. Create an object of the class from both the constructors and call the method **display**.

Video

Watch our video on default constructor in Dart.