

Dart Class

Dart Class

Dart is an Object Oriented Programming language and Dart supports Classes.

In this tutorial, we will learn about Dart Classes, constructors and named constructors in a class.

Syntax of a Class

A class can contain variables (properties), constructors and methods (behavior). `class` keyword is used to define a class in Dart.

Following is the syntax of a Dart Class.

```
class ClassName {  
    //variables  
    //constructor  
    //named constructors  
    //methods  
}
```

where ClassName is the name by which this class is referenced.

Example – Dart Class with variables

Let us define a simple Dart Class with just variables.

```
class Car{  
    String name;  
    int miles;  
}
```

This class has two variables: one is of type String and the other is of type integer.

Example – Dart Class with variables and methods

Let us define a simple Dart Class with variables and methods.

```
class Car{
  String name;
  int miles;

  void printDetails() {
    print(name+' has gone '+miles.toString()+ ' miles. ');
  }
}
```

This class has two variables and one method. The method is `printDetails()`.

Create an object of this Class type

We can create as many number of objects of a Class type as required. And you can access variables and call methods using the dot operator.

Let us the Car class we defined in the previous example and create an object, access the variables and call methods.

example.dart

```
class Car{
  String name;
  int miles;

  void printDetails() {
    print(name+' has gone '+miles.toString()+ ' miles. ');
  }
}

void main(){
  //create obejct
  Car car = Car();

  //set variables
  car.name = 'Ford Mustang';
  car.miles = 22000;

  //call method
  car.printDetails();
}
```

Output

```
Ford Mustang has gone 22000 miles.
```

Constructor

A constructor shares the syntax of a method, with same name as that of class and without any return type.

When you create an object of a class type, you can provide the values for parameters given in a constructor.

```
class Car{
  String name;
  int miles;

  Car(name, miles) {
    this.name = name;
    this.miles = miles;
  }
}
```

The number of parameters you provide for a constructor are your choice.

Let us see an example of class using constructor.

Dart Program – example.dart

```
class Car{
  //variables
  String name;
  int miles;

  //constructor
  Car(name, miles) {
    this.name = name;
    this.miles = miles;
  }

  //method
  void printDetails() {
    print(name+' has gone '+miles.toString()+' miles.');
```

```
  }
}

void main(){
  //create obejct
  Car car = Car('Ford Mustang', 22320);

  //call method
  car.printDetails();
}
```

Output

```
Ford Mustang has gone 22320 miles.
```

Named Constructors

You can define special type of constructors called named constructors in a class. The name of this named constructor is followed after the class name with a dot in the definition.

An example is shown below, where in we defined a named constructor, along with a standard constructor for class `Car`.

Dart Program – example.dart

```
class Car{
  //variables
  String name;
  int miles;

  //constructor
  Car(name, miles) {
    this.name = name;
    this.miles = miles;
  }

  //named constructor
  Car.fromName(name){
    this.name = name;
    this.miles = 10000;
  }

  //method
  void printDetails() {
    print(name+' has gone '+miles.toString()+ ' miles.');
```

```
  }
}

void main(){
  //create object
  Car car = Car.fromName('Ford Mustang');
```

```
  //call method
  car.printDetails();
}
```

Output

```
Ford Mustang has gone 10000 miles.
```

Conclusion

In this [Dart Tutorial](#), we have learned how to define a class in Dart, structure of a Dart Class, how to create objects of a class type, constructors and named constructors for a class.

Dart Programming

- ◆ [Dart Tutorial](#)
- ◆ [Install Dart on Windows](#)
- ◆ [Dart - Hello World](#)
- ◆ [Dart - Variables](#)
- ◆ [Dart - Comments](#)
- ◆ [Dart - If Else](#)
- ◆ [Dart - For Loop](#)

Dart String Operations

- ◆ [Dart - Concatenate Strings](#)
- ◆ [Dart - Split String](#)
- ◆ [Dart - Replace Substring in String](#)
- ◆ [Dart - Find Substring of String](#)
- ◆ [Dart - String Length](#)
- ◆ [Dart - Trim String](#)

Dart Exception Handling

- ◆ [Dart - Try Catch](#)

Dart List Operations

- ◆ [Dart - List](#)
- ◆ [Dart List - Iterate](#)
- ◆ [Dart - Check if List is Empty](#)
- ◆ [Dart - Check if List Contains Element](#)
- ◆ [Dart Reverse List](#)
- ◆ [Dart Join Lists](#)
- ◆ [Dart - Check Equality of Two Lists](#)