Constructors

In this lesson, we will learn about a special method with which you can initialize instance variables while creating an instance of a class.

We'll cover the following

Introduction

In Dart, **constructors** are special functions of a class that are responsible for initializing the instance variables of that class.

A constructor must have the same name as the class it is being declared in and since it is a function, it can be parameterized. However, unlike regular functions, constructors do not have a return value, hence cannot have a return type.

Dart offers multiple types of constructors. In this course, we will be learning about two of them.

1. Generative Constructor
2. Named Constructor

Generative Constructor

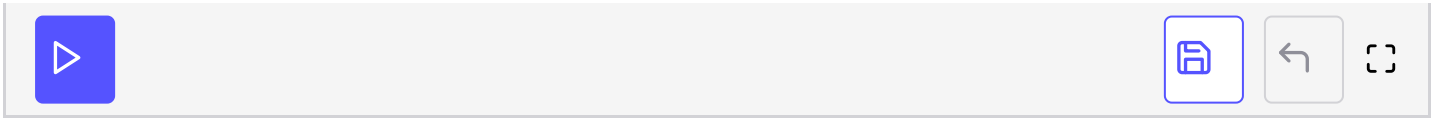
The most common form of a constructor, the generative constructor, creates a new instance of a class.

```
class Person{
  String name;
  String gender;
  int age;
```

```
  // Generative Constructor
  Person(String nameC, String genderC, int ageC){
    this.name = nameC;
    this.gender = genderC;
    this.age = ageC;
  }

  walking() => print('$name is walking');
  talking() => print('$name is talking');
}

int main() {
  var firstPerson = Person("Sarah","Female",25);
  print(firstPerson.name);
  print(firstPerson.gender);
  print(firstPerson.age);
}
```



As discussed in the previous lesson, we can create multiple instances of a single class. The `this` keyword refers to the current instance. On **line 8** (`this.name = nameC;`), we are assigning the value `nameC` to the instance variable `name` of the current instance.

On **line 18**, we are creating an instance of the `Person` class using the generative constructor. Now instead of individually assigning values to our instance variables, all we have to do is pass them to the constructor that will do the rest.

Syntactic Sugar

A generative constructor doesn't require any function body. We can assign `this.name` as a parameter. This allows us to write concise code.

```
class Person{
  String name;
  String gender;
  int age;

  // Generative Constructor
  Person(this.name, this.gender, this.age);

  walking() => print('$name is walking');
  talking() => print('$name is talking');
}

int main() {
  var firstPerson = Person("Sarah","Female",25);
  print(firstPerson.name);
  print(firstPerson.gender);
  print(firstPerson.age);
}
```

Named Constructor

We can create multiple constructors in Dart based on different scenarios. In such cases, it is better to name the constructors for better clarity.

The general syntax is as follows:

```
ClassName.contructorName{
   Constuctor Body
}
```

```
class Person{
  String name;
  String gender;
  int age;

  // Generative Constructor
  Person(this.name, this.gender, this.age);

  // Named Constructor
  Person.newBorn(){
    this.age = 0;
  }

  walking() => print('$name is walking');
  talking() => print('$name is talking');
}

int main() {
  var firstPerson = Person("Sarah","Female",25);
  var secondPerson = Person.newBorn();
  print(secondPerson.age);
}
```

Default Constructor

In the previous lesson, we created an instance of a class without using a constructor, or so we thought. If you don't declare a constructor, a default constructor is provided for you. A default constructor has zero arguments and creates an instance of a class without initializing its instance variables.

Let’s move on to getters and setters in the next lesson.