



TOUR OF SCALA

CASE CLASSES

Case classes are like regular classes with a few key differences which we will go over. Case classes are good for modeling immutable data. In the next step of the tour, we'll see how they are useful in [pattern matching](#).

Defining a case class

A minimal case class requires the keywords `case class`, an identifier, and a parameter list (which may be empty):

```
case class Book(isbn: String)

val frankenstein = Book("978-0486282114")
```

Notice how the keyword `new` was not used to instantiate the `Book` case class. This is because case classes have an `apply` method by default which takes care of object construction.

When you create a case class with parameters, the parameters are public `val`s.

```
case class Message(sender: String, recipient: String, body: String)
val message1 = Message("guillaume@quebec.ca", "jorge@catalonia.es", "Ça va ?")

println(message1.sender) // prints guillaume@quebec.ca
message1.sender = "travis@washington.us" // this line does not compile
```

You can't reassign `message1.sender` because it is a `val` (i.e. immutable). It is possible to use `var`s in case classes but this is discouraged.

Comparison

Instances of case classes are compared by structure and not by reference:

```
case class Message(sender: String, recipient: String, body: String)

val message2 = Message("jorge@catalonia.es", "guillaume@quebec.ca", "Com va?")
val message3 = Message("jorge@catalonia.es", "guillaume@quebec.ca", "Com va?")
val messagesAreTheSame = message2 == message3 // true
```

Even though `message2` and `message3` refer to different objects, the value of each object is equal.

Copying

You can create a (shallow) copy of an instance of a case class simply by using the `copy` method. You can optionally change the constructor arguments.

```
case class Message(sender: String, recipient: String, body: String)
val message4 = Message("julien@hpotage.fr", "travis@washington.us", "Me zez a komz gent ma amezeg")
```

```
val message4 = message( julien@breizhne.fr , trAVIS@wASHINGTON.US , Me zo o komz gant ma amezeg )
val message5 = message4.copy(sender = message4.recipient, recipient = "claire@bourgogne.fr")
message5.sender // trAVIS@washington.us
message5.recipient // claire@bourgogne.fr
message5.body // "Me zo o komz gant ma amezeg"
```

The recipient of `message4` is used as the sender of `message5` but the `body` of `message4` was copied directly.

More resources

- Learn more about case classes in the [Scala Book](#)

[← previous](#)

[next →](#)

Contributors to this page:

 ckipp01

 mlachkar

 int-index

 ashawley

 fineconstant

 sujithjay

 EricJLarson

 heathermiller

DOCUMENTATION

- Getting Started
- API
- Overviews/Guides
- Language Specification

DOWNLOAD

- Current Version
- All versions

COMMUNITY

- Community
- Mailing Lists
- Chat Rooms & More
- Libraries and Tools
- The Scala Center

CONTRIBUTE

- How to help
- Report an Issue

SCALA

- Blog
- Code of Conduct
- License

SOCIAL

- GitHub
- Twitter

