

Objects



Lesson Objectives

- After completing this lesson, you should be able to:
 - Create Singleton objects in Scala
 - Describe the difference between a class and an object in Scala
 - Outline usages for objects in Scala applications
 - Start a Scala application

What is an Object?

- The Singleton Pattern
 - Defines a single instance of a class that cannot be recreated within a single JVM instance
 - Can be directly accessed via its name

A Simple Scala Object

```
object Hello {  
  def message = "Hello!"  
}
```

```
scala> Hello.message  
res0: java.lang.String = "Hello!"
```

Why is this Useful?

- Many languages permit the definition of “static” fields and methods
- These are globally available within the runtime, such as a JVM
- They are not related to specific instances of a class

When are Objects Used?

- Class Factories
- Utility methods
- Constant definitions

A Simple Object

```
object Hello {  
  val oneHourInMinutes: Int = 60  
  
  def createTimeFromMinutes(minutes: Int) =  
    new Time(  
      minutes / oneHourInMinutes,  
      minutes % oneHourInMinutes)  
}
```

Starting a Scala Application

```
object Hello {  
  def main(args: Array[String]): Unit =  
    println("Hello!")  
}
```


Starting a Scala Application

```
$ scala -cp target/scala-2.11/classes/ Hello  
Hello
```

```
> run  
[info] Running Hello  
Hello  
[success] Total time: 0 s, completed Jul 20, 2012 6:00:20 PM
```

Lesson Summary

- Having completing this lesson, you should be able to:
 - Create Singleton objects in Scala
 - Describe the difference between a class and an object in Scala
 - Outline usages for objects in Scala applications
 - Start a Scala application