



<http://www.scala-lang.org>

David Vilaseca
Sergi Mas

Índex

- Curiositats
- Client (codi)
- Particularitats
- Opinió
- Preguntes

Curiositats

- École Polytechnique Fédérale de Lausanne, Suïssa
- Martin Odersky → 2001
- Llenguatge modern
- Elegant
- Online Learning (web)
- Comunitat

- Orientat a objecte
- Bytecodes per a la JVM
- Successor de JAVA
- Compatibilitat amb Android i JavaScript

```

1 import akka.actor._
2 import java.net._
3 import java.io._
4 import scala.io._
5
6
7 ▼ object ProvaClient extends App{
8     »
9         val client = ActorSystem("Client")
10        val socket = new Socket(args(0), args(1).toInt)
11        val write = client.actorOf(Props(new ProvaClient(socket)))
12        val read = client.actorOf(Props(new ProvaClient(socket)))
13        write ! "write"
14        read ! "read"
15    }
16
17 ▼ class ProvaClient(socket: Socket) extends Actor {
18     var line = new String()
19     val in = new BufferedReader(new InputStreamReader(socket.getInputStream()))
20     val out = new PrintWriter(socket.getOutputStream(), true)
21     ▼ def receive = {
22         case "write" =>
23             while({line = StdIn.readLine(); line != null})
24                 out.println(line)
25         case "read" =>
26             while({line = in.readLine(); line != null})
27                 println(line)
28     }
29 }

```

Particularitats

```
7 ▼ object ProvaClient extends App{  
8   »  
9     val client = ActorSystem("Client")  
10    val socket = new Socket(args(0), args(1).toInt)  
11    val write = client.actorOf(Props(new ProvaClient(socket)))  
12    val read = client.actorOf(Props(new ProvaClient(socket)))  
13    write ! "write"  
14    read ! "read"  
15  }
```

- App
- ActorSystem
- Props
- ! (receive)

```
17 ▼ class ProvaClient(socket: Socket) extends Actor {  
18     var line = new String()  
19     val in = new BufferedReader(new InputStreamReader(socket.getInputStream()))  
20     val out = new PrintWriter(socket.getOutputStream(), true)  
21 ▼     def receive = {  
22         case "write" =>  
23             while({line = StdIn.readLine(); line != null})  
24                 out.println(line)  
25         case "read" =>  
26             while({line = in.readLine(); line != null})  
27                 println(line)  
28     }  
29 }
```

- receive
- Unit → while


```

1 import akka.actor._
2 import java.net._
3 import java.io._
4 import scala.io._
5
6 ▼ object ProvaClient extends App{
7     val client = ActorSystem("Client")
8     val socket = new Socket("localhost", args(0).toInt)
9     val write = client.actorOf(Props(new ProvaClient(socket, client)))
10    val read = client.actorOf(Props(new ProvaClient(socket, client)))
11    write ! "write"
12    read ! "read"
13    client.awaitTermination()
14 }
15
16 ▼ class ProvaClient(socket: Socket, client: ActorSystem) extends Actor {
17     var line = new String()
18     line = null
19     val in = new BufferedReader(new InputStreamReader(socket.getInputStream()))
20     val out = new PrintWriter(socket.getOutputStream(), true)
21     ▼ def receive = {
22         case "write" =>
23             while({line = StdIn.readLine(); line != null})
24                 out.println(line)
25                 out.println("\u0000")
26                 self ! PoisonPill
27         case "read" =>
28             while({line = in.readLine(); line != "\u0000"})
29                 println(line)
30                 socket.close()
31                 println("Server-->Bye!")
32                 self ! PoisonPill
33     }
34     client.shutdown()
35 }

```


Opinió

Preguntes?

Gràcies