



Three languages in thirty minutes or so...



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Three languages

Who are we?

- Fabio Marinelli
 - Project manager @ Poste Italiane
 - twitter @fmarinelli
- Sergio Bossa
 - Senior Software Engineer @ Bwin Italy
 - (Micro-)Blogger @sbtourist
- Massimiliano Dessí
 - Architect/Developer @ Pronetics/Sourcesense
 - twitter @desmax74



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessí - Marinelli



Three languages

Why this challenge?

More languages means:

- Right tool for your problems
- Different mindsets improve your problem solving skills
- ... and it's cool!



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Three languages

How we do this? Let us talk about ...

Data structures

Variables

Objects and Functions

Flow control

Concurrency



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Javascript

Fabio Marinelli



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Javascript - Variables

- Loosely typed language
- Types: number, string, boolean, function and object
- Everything is an Object
- Function scoped!!



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Javascript - Variables

Declare and initialize:

```
var a;  
  
var a = 1; //number  
  
var a = "string";  
  
var a = 'string';  
  
var a = true || false;  
  
var a = {name:{first:"Fabio", last:"M."}, age:33};  
  
var a = function (text) { alert(text);};
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Javascript - Variables

False values:

- false
- null
- undefined
- empty string
- 0
- NaN



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Javascript - Flow control

Is this C?

```
if (expression) { ... } else { ... }

while (expression) { ... }

do { ... } while (expression);

for (initialization; condition; increment) { ... }

for (name in object) { ... }

try { ... } catch (variable) { ... }
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>



Javascript - Data structures

Javascript object is a mutable key-value map.

```
var object = {  
    "name": value,  
    "other": function (text) {  
        alert(text);  
    }  
};
```

Values can be also a function. Everything is an object.



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*



Javascript - Objects and functions

Functions are first class objects

```
var numbers = [1,5,2,12,7,9];
numbers.sort(function(a, b) {
    return a-b;
});
//Output a sorted array
//1,2,5,7,9,12
numbers.sort(function(a, b) {
    return b-a;
});
//Output a sorted array
//12,9,7,5,2,1
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>



Javascript - Objects and functions

Functions has access to context in which they are created

```
var counter = function (value) {  
    return {  
        getAndInc : function () { return value++; }  
    };  
};  
  
var myCounter = counter(1);  
alert(myCounter.getAndInc()); // Output 1  
alert(myCounter.getAndInc()); // Output 2
```

This is called closure



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>



Javascript - Objects and functions

Let's use the closure.

```
var Foo = function() {  
    var value = 1; //private to the function  
    this.inc = function() { ++value; };  
    this.print = function() { alert(value); };  
    return this;  
};  
  
var b = new Foo();  
b.print(); //Output 1  
b.inc();  
b.print(); //Output 2  
  
alert(b.value); //undefined
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>



Javascript is a prototypal language

```
var Foo = function() {  
    this.value = 1; //public value  
    return this;  
};  
Foo.prototype.inc = function() {  
    ++this.value;  
};  
Foo.prototype.print = function() {  
    alert(this.value);  
};  
var b = new Foo();  
b.print(); //Output 1  
b.inc();  
b.print(); //Output 2
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>



Javascript - Objects and functions

Developer can add responsibilities to any object

```
var b = 1;  
Number.prototype.inc = function() {  
    return new Number(this+1);  
};  
Number.prototype.print = function() {  
    alert(this);  
};  
b.print();  
//Output 1  
b.inc().print();  
//Output 2
```

Again, everything is an object.



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>



Javascript - Concurrency

- Specification says nothing about concurrency
- Browser engine is single threaded
- Node.js offers event model concurrency



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>



Node.js

```
var sys = require("sys");
setTimeout(function () {
    sys.puts("world");
}, 1000);
sys.puts("hello");
```

Evented model with non blocking I/O



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>



Javascript - Concurrency

...output

hello

after 1 second

world



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*



Clojure

Sergio Bossa



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Clojure - Three Words

Homoiconic

Functional

Dynamic



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Clojure - Basics

From “Programming Clojure”, by Stuart Halloway:

Form	Example(s)
Boolean	true, false
Character	\a
Keyword	:tag, :doc
List	(1 2 3), (println “foo”)
Map	(:name “Bill”, :age 42)
Nil	nil
Number	1, 4.2
Set	#(:snap :crackle :pop)
String	“hello”
Symbol	user/foo, java.lang.String
Vector	[1 2 3]



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Clojure - Functions (Definition)

Standard

```
(defn my-fun [name] (println "Hello, " name "!"))
```

Anonymous I

```
(fn [name] (println "Hello, " name "!"))
```

Anonymous II

```
#(println "Hello, " %1 "!")
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Clojure - Functions (Invocation)

Standard

(my-fun “Sergio”)

Anonymous I

((fn [name] (println “Hello, ” name “!”)) “Sergio”)

Anonymous II

(#(println “Hello, ” %1 “!”) “Sergio”)



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Clojure - Variables (Basics)

Namespace-scoped

(**def my-variable “value”**)

Locally-scoped

(**let [my-variable “value”]**)

Use

(**println my-variable**)



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Clojure - Variables (Destructuring)

Definition

```
(defn my-fun [{name:name}] (println "Hello, " name "!"))
```

Invocation

```
(my-fun {:name "Sergio" :city "Rome"})
```

Result

=> Hello, Sergio!



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Clojure - Control Structures (Conditionals)

Single choice

(when (= n 0) “zero!”)

Double choice

(if (= n 0) “zero!” “not zero!”)

Multiple choice

(cond (< n 0) ‘less than zero!’ (= n 0) ‘zero!’ :else ‘greater than zero!’)



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Clojure - Control Structures (Loops)

**NO FOR LOOPS
(almost)**

Explicit recursion

```
(defn sum-a [array] (loop [a array s 0] (if (empty? a) s (recur (rest a) (+ s (first a))))))
```



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Clojure - Data Structures (Concepts)

Everything is (treated as) a Sequence

Immutable

(Possibly) Lazy



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Clojure - Data structures (Basics)

Essentials

first, rest, cons, conj, into

Comprehension

(for [m [1 2 3] n [4 5 6]] (+ m n))

Map/Reduce

(reduce #(merge-with + %1 %2) (map #(hash-map %1 1) ['pippo' 'pluto' 'pluto']))



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Clojure - Concurrency (Concepts)

Explicit support

Coordinated

Uncoordinated

Asynchronous



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Clojure - Concurrency (Coordinated)

References (aka STM)

(def person (ref {:name 'Sergio Bossa' :age 28}))

Get

@person

Update

(dosync (alter person conj {:age 29}))



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Clojure - Concurrency (Uncoordinated)

Atoms

```
(def person (atom {:name 'Sergio Bossa' :age 28}))
```

Get

```
@person
```

Update

```
(swap! person conj {:age 29})
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Clojure - Concurrency (Asynchronous)

Agents

```
(def person (agent {:name 'Sergio Bossa' :age 28}))
```

Get

```
@person
```

Update

```
(send person conj {:age 29})
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Clojure - Final Words

KISS

Powerful

Concurrent



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Scala

Massimiliano Dessì



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Scala variables

Scala is a hybrid functional/object-oriented language.

Scala runs on the JVM and interoperates with all Java libraries.

Scala is statically typed with inference

Powerful language for designing internal DSLs (DSL In Action)

Good for Event Driven Architecture/CQRS (Akka actors)



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Scala variables

In functional programming, variables' values are immutable, as consequence of the mathematical orientation

Immutability = no side effect

No side effect = no shared state, better concurrency

Scala allows you to decide whether a variable is immutable or not.



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Variable declaration

An immutable “variable” it’s declared with `val` keyword (Value Object),
the reference can not be changed, but the object itself can be
modified.

```
val myArray[String]:Array= new Array(3)  
array: Array[String] = Array(null, null, null)
```

A mutable “variable” it’s declared with `var` keyword, You can assign a
new value to a var as often as you want

```
var price: Double = 100  
price = 10
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Control structures

Function and `if`, `while`, `for`, `try`, `match` in the base syntax, higher-level control structure other than these in the libraries

• Function

Functional programming is based on the behavior of functions in the mathematical sense, with all the implications that starting point implies.

`(x: Int) => x + 1`

functions have no side effects

When a function takes other functions as arguments or returns a function, it is called a higher-order function.

`List(1, 2, 3, 4, 5).reduceLeft { _ * _ }`



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Control structures

- Closure

```
var factor = 3  
val multiplier = (i:Int) => i * factor
```

- For Comprehension

```
val dogs = List("Lassie", "Rex", "Rin tin tin")  
for (dog <- dogs) println(dog)
```

- Currying

```
def add(x:Int) = (y:Int) => x + y  
println(add(2)(3))  
val addCurried = curried(add _)
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Control structures

• Composition

```
val f = (x: Int) => x + 1
val g = (x: Int) => x * 3.3
val h = (x: Double) => x + 0.1
val result = h compose g compose f
```

• Pattern matching

```
def factorial(i: BigInt): BigInt = i match {
    case _ if i == 1 => i
    case _ => i * factorial(i - 1)
}
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Control structures

- Precondition

```
class Rational(n: Int, d: Int) {  
    require(d != 0) ....  
}
```

- Filtering

```
for (file <- filesHere if file.getName.endsWith(".scala"))
```

- Folding and reducing

```
List(1,2,3,4,5,6).reduceLeft(_ + _)  
List(1,2,3,4,5,6).foldLeft(10)(_ * _)
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Specification

```
class ElementSpec extends FlatSpec with ShouldMatchers {  
  "A UniformElement" should  
    "have a width equal to the passed value" in {  
      val ele = elem('x', 2, 3)  
      ele.width should be (2)  
    }  
    it should "have a height equal to the passed value" in {  
      val ele = elem('x', 2, 3)  
      ele.height should be (3)  
    }  
    it should "throw an IAE if passed a negative width" in {  
      evaluating {  
        elem('x', -2, 3) } should produce [IllegalArgumentException]  
    }  
}
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Collection

- Sequences (List, Array, ListBuffer) Immutable and mutable
- Set and Maps Immutable and mutable
- Tuples, a tuple combines a fixed number of items together

(1, "hello", Console)



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Concurrency

• Actor and messages

```
class McQueen extends Actor {  
    loop {  
        react {  
            case "how many?" => {  
                println("I've got " + mailboxSize.toString  
                    + " messages in my mailbox.")  
            }  
        }  
    }  
}
```

Every Actor has a mailbox in which messages sent to that Actor are queued



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Concurrency

Actors can receive any sort of object as a message

```
val fussyActor = actor {  
    loop {  
        receive {  
            case s: String => println("I got a String: " + s)  
            case i: Int => println("I got an Int: " + i.toString)  
            case _ => println("I have no idea what I just got.")  
        }  
    }  
  
    fussyActor ! "hi there"  
    fussyActor ! 23  
    fussyActor ! 3.33
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Concurrency

A strategy to keep your Actor-based code asynchronous is the use of futures

A future is a placeholder object for a value that hasn't yet been returned from an asynchronous process

the double-bang (!!) on Actor causes operation to return a future

```
/**
```

```
* Asynchronous message send. Send-and-receive eventually. Waits  
on a Future for the reply message.
```

```
* If received within the Actor default timeout interval then it  
returns Some(result) and
```

```
* if a timeout
```

```
* has occurred None.
```

```
*/
```

```
def !!(message: T): Box[R]
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Trait

Traits are like interfaces in Java, but they can also have method implementations and even fields.

Objects are constructed by mixin composition

```
class Animal trait HasLegs  
class Frog extends Animal with Philosophical with HasLegs {  
    override def toString = "green"  
}
```



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Now ... what?!?!

Quick (really!) start



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Quickstart - Javascript

- Download and install Nodejs.
- Write your first example.js

```
var sys = require("sys") ;  
setTimeOut(function () { sys.puts("world") ; } , 1000) ;  
sys.puts("hello") ;
```

- Run example

```
% node example.js
```

- Or easily start node console by running

```
% node-repl
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Quickstart - Clojure

Download JDK >= 5.x and Clojure jars

<http://www.sun.com> <http://www.oracle.com> <http://clojure.org/downloads>

Download and install Leiningen build tool

<http://github.com/technomancy/lein>

\$> lein self-install

Start new project

\$> lein new my-project

Start project REPL

\$> lein repl



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



Quickstart - Scala

The SBT can be used for Scala projects as Maven for Java Projects. SBT enable Dependency management. It support multiple project/subproject, and mixed project Scala/Java. It also has built-in support for generating Scaladocs and for testing with ScalaTest, Specs, and ScalaCheck.

```
class Project(info: ProjectInfo) extends DefaultWebProject(info) with IdeaProject {  
  
    lazy val scalatest = "org.scalatest" % "scalatest" % "1.3" % "test"  
  
    lazy val scalateCore = "org.fusesource.scalate" % "scalate-core" % "1.4.0"  
  
    lazy val casbah = "com.mongodb.casbah" %% "casbah" % "2.0.3"  
  
    lazy val salat = "com.novus" %% "salat" % "0.0.5"  
  
    lazy val junit = "junit" % "junit" % "4.8.2" % "test"  
  
    lazy val slf4japi = "org.slf4j" % "slf4j-api" % "1.6.1" % "runtime"  
  
    lazy val logback = "ch.qos.logback" % "logback-classic" % "0.9.27"  
  
    val scalaTools = "Scala Tools Repository snapshot" at "http://nexus.scala-tools.org/content/repositories/public/"  
  
}
```



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



Quickstart - Scala

1) Download JDK >= 5.x (and set the env vars) <http://www.oracle.com>

2) Download Scala installer jar <http://www.scala-lang.org/downloads>

and set the environment \$SCALA_HOME (ie: /usr/local/share/scala) and

\$PATH:\$SCALA_HOME/bin

3) Use scala with

-shell

-IntelliJ idea community with scala plugin

-eclipse with scala plugin



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli



References



*Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>*

Bossa - Dessì - Marinelli



References

• Javascript

- Douglas Crockford (Javascript the good parts) <http://www.crockford.com/>
- Nodejs <http://nodejs.org/>
- John Resig (JQuery, Pro Javascript and Javascript secrets books) <http://ejohn.org/>

• Clojure

- Clojure Web Site: <http://clojure.org>
- Clojure Online Docs: <http://clojuredocs.org/>
- Programming Clojure Book: <http://pragprog.com/titles/shcloj/programming-clojure>

• Scala

- Scala site : <http://www.scala-lang.org>
- Scala docs: <http://www.scala-lang.org/node/197>
- Scala Books : <http://www.scala-lang.org/node/959>



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Bossa - Dessì - Marinelli