Scala

My title here

Carlos Tomé Cortiñas, Renate Eilers and Matthew Swart

Universiteit Utrecht

Introduction

- ▶ Background information
- ▶ Scala
- ▶ Conclusion

Universiteit Utrecht

History

- ▶ 2001
- ▶ École Polytechnique Fédérale de Lausanne (EPFL) by Martin Odersky

Universiteit Utrecht

Companies (Not sure)

- ▶ Twitter
- ▶ Linkedin
- ▶ The Guardian
- ▶ FourSquare
- ▶ Sony
- ▶ etc.

Projects

- PlayFramework
- Akka

Scala (Still need to add a picture)

- ▶ Object oriented
- ▶ Functional
- ▶

Universiteit Utrecht

REPL

Figure 1

Universiteit Utrecht

[Faculty of Science
Information and Computing
Sciences]

Mutable and immutable

```
Welcome to Scala 2.12.0 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_111).
Type in expressions for evaluation. Or try :help.

scala> var mutableNumber = 5
mutableNumber: Int = 5

scala> mutableNumber = 10
mutableNumber: Int = 10

scala> val immutableNumber = 5
immutableNumber: Int = 5

scala> immutableNumber = 10
<console>:12: error: reassignment to val
       immutableNumber = 10
```

Figure 2

For loops

```scala
object ForLoops {
  def forEach(values : List[String]) : Unit = {
    for(value <- values){
      println(value)
    }
  }
  def rangeLoop(): Unit ={
    for( i <- 0 to 10){
      print(i)
    }
  }
  def rangeUntilLoop(): Unit ={
    for( i <- 0 until 10){
      print(i)
    }
  }
}
```

[Faculty of Science
Information and Computing
Sciences]

For loops

```scala
object ForLoops {
  def filterLoop(values : List[Int]): Unit ={
    for( value <- values if (value % 2 == 0)){
      print(value)
    }
  }
  def concatList(values : List[List[Any]]) : List[Any]
    for(value <- values; n <- value) yield n
  }

}
```

Universiteit Utrecht

[Faculty of Science
Information and Computing
Sciences]

Currying

Universiteit Utrecht

Function types

A => B is an abbrevication for the class scala.Function1[A,B]
package scala

```scala
trait Function1[A,B]{
    def apply(x : A) : B
}
```

scala traits Function2...Function22

Anonymous function

```scala
scala> values.map(x => x + 1)
res3: List[Int] = List(2, 3, 4, 5, 6, 7, 8, 9, 10, 11)
```

Universiteit Utrecht

[Faculty of Science
Information and Computing
Sciences]

Anonymous function

Anonymous function (x : Int) => x + 1

This will be expanded to (Same syntax as Java)

```
new Function1[Int,Int]{
    def apply(x : Int) : Int = x + 1
}
```
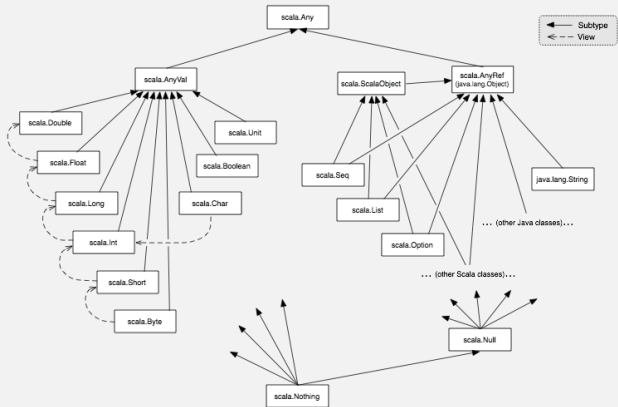
Figure 3

Universiteit Utrecht

Apply method

Universiteit Utrecht

Classes

```scala
class Dog(name : String){
  def sound : Unit = println("Woof!")
  def movement : String = "Walk"
}
```

```java
class Dog{
    private String name;
    public Dog(String name){
        this.name = name;
    }
    public void sound(){
        System.out.println("Woof!")
    }
    public String movement(){
        return("Walk")
    }
}
```

Abstract classes

Examples

Universiteit Utrecht

Anonymous class

Case Classes

```scala
case class Person(name : String) extends Animal{
  def noise = "I am a person"
}
case class Tiger() extends Animal{
  override def noise : String = "Grr"
}

case class Frog() extends Animal{
  def noise : String = "CROAK"
}
```

Pattern Match

```scala
object AnimalNoise{
  def mkSound(animal : Animal) : Unit =
    animal match {
      case Frog() => println(Frog().noise)
      case Person(name) => println(name)
      case x => println(x.noise)
    }
}
```

Pattern Match

```scala
object pattern{
  def all(allTypes : Any) : Unit = {
    allTypes match{
      case (x,1) => println("(x,1)")
      case (x,y) => println("(x,y)")
      case x : String => println(x)
      case Tiger() => println(Tiger().noise)
      case 1 => println("One")
      case true => print("True")
    }
  }
}
```

Universiteit Utrecht

[Faculty of Science
Information and Computing
Sciences]

Examples.

Traits.

Traits

example

Universiteit Utrecht

Mixin - Example

example

Universiteit Utrecht

Sealed

example

Universiteit Utrecht

Classes

	Class	Abstract class	object	object
Inherentence				
Cmposition				
Parameters				

Universiteit Utrecht

Tuples

Universiteit Utrecht

Evaluation

```scala
class LazyMethod(values : () => List[List[String]]) {
    def heavyComputation : List[String] = {
      for(value <- values; n <- value)
        for(value1 <- values; n1 <- value)
          for(value2 <- values; n2 <- value)
            for(value3 <- values; n3 <- value)
              for(value4 <- values; n4 <- value)
    }
}

...
```

[Faculty of Science
Information and Computing
Sciences]

Currying

Currying - Anonymous function

```scala
var f = (x : Int) => (y : Int) => x + 1
f: Int => (Int => Int) = $$Lambda$1113/551797833@2c58d

scala> f(1)
res0: Int => Int = $$Lambda$1130/1813375175@56380231

scala> f(1)(2)
res1: Int = 2
```

Universiteit Utrecht

[Faculty of Science
Information and Computing
Sciences]

Typeclasses

example

Typeclasses Scala

example

Generics

example

Generics

```
class GParent
class Parent extends GParent
class Child extends Parent
class ChildOfChild extends Child
case class Children[A <: Child](value : A)
case class Children1(Value : Child)
case class Ancestors[A >: Child](value : A)
case class Family[P >: Parent, C <: Child]
        (ancestors : (P,P), children : List[C])
```

...

Covariance

example

Questions

Eend

Universiteit Utrecht