

# Einführung Scala

---

# Warum hier Scala?

- Lisp, ML, Miranda, Haskell
    - rein funktional
    - eignen sich zum Erlernen funktionaler Konzepte
  - Scala, sowohl
    - imperativ
    - objektorientiert
    - funktional
    - lenkt ab vom Erlernen funktionaler Konzept
  - aber viel häufiger in Praxis eingesetzt als reine FuPs
- > Verwendung von auf funktionaler Programmierung eingeschränktes Scala

# Was ist Scala?

---

- seit 2001: von Martin Odersky entwickelt
- 2003: erstmals der Öffentlichkeit vorgestellt
- 2006: Version 2.0 ähnlich zu heutigem Scala
- ausführbar als:
  - REPL, Read-Evaluate-Print Loop (interaktive Konsole)
  - Skript
  - vollwertige App: übersetzt in JVM-Bytecode

# Quicksort in Scala

## imperativ, aber mit Rekursion

```
def sort(xs: Array[Int]) {  
    def swap(i: Int, j: Int) {  
        val t = xs(i); xs(i) = xs(j); xs(j) = t  
    }  
    def sort1(l: Int, r: Int) {  
        val pivot = xs((l + r) / 2)  
        var i = l; var j = r  
        while (i <= j) {  
            while (xs(i) < pivot) i += 1  
            while (xs(j) > pivot) j -= 1  
            if (i <= j) {  
                swap(i, j)  
                i += 1  
                j -= 1  
            }  
            if (l < j) sort1(l, j)  
            if (j < r) sort1(j, r)  
        }  
        sort1(0, xs.length - 1)  
    }  
}
```

### Aufruf:

```
var ar = Array(1,3,1,2)  
sort(ar)  
ar
```

### liefert:

```
ar: Array[Int] = Array(1, 3, 1, 2)  
res0: Unit = ()  
res1: Array[Int] = Array(1, 1, 2, 3)
```

# Quicksort in Scala

## funktional mit Higher-Order Funktion

```
def sortFun(xs: List[Int]): List[Int] = {  
    if (xs.length <= 1) xs  
    else {  
        List.concat(  
            sortFun(filter(xs, xs(1) >)),  
            filter(xs, xs(1) ==),  
            sortFun(filter(xs, xs(1) <)))  
    }  
} //pivot = 1
```

### Aufruf:

```
sortFun(1::3::2::Nil)
```

### liefert:

```
res0: List[Int] = List(1, 2, 3)
```

# Wichtiges um Scala

- **Entwicklungsumgebungen**
  - Plugins für Eclipse, IntelliJ, ....
  - direkt im Browser ausführbar: z.B. ScalaFiddle
- **Literatur:**
  - meist zu stark auf objektorientierte zugeschnitten
  - am besten Folien verwenden
  - wird nachgeliefert, wenn es um weiterführende Themen geht
  - Interessante Links:
    - [Scala by Example](#)
    - [Offizielle Dokumentation](#) (Getting Started, API, Overview, Style guide, Language Spec, ...)