Syntax Institut

Modul 3 – Android App Entwicklung mit Kotlin

Kotlin Basics

(Datentypen und Listen)

Gliederung

- Was sind Daten Typen
- Boolean, Integer, Double, String
- Listen

Was sind Datentypen?

Quelle: https://www.nabu.de/umwelt-und-ressourcen/ressourcenschonung/einzelhandel-und-umwelt/nachhaltigkeit/30684.html

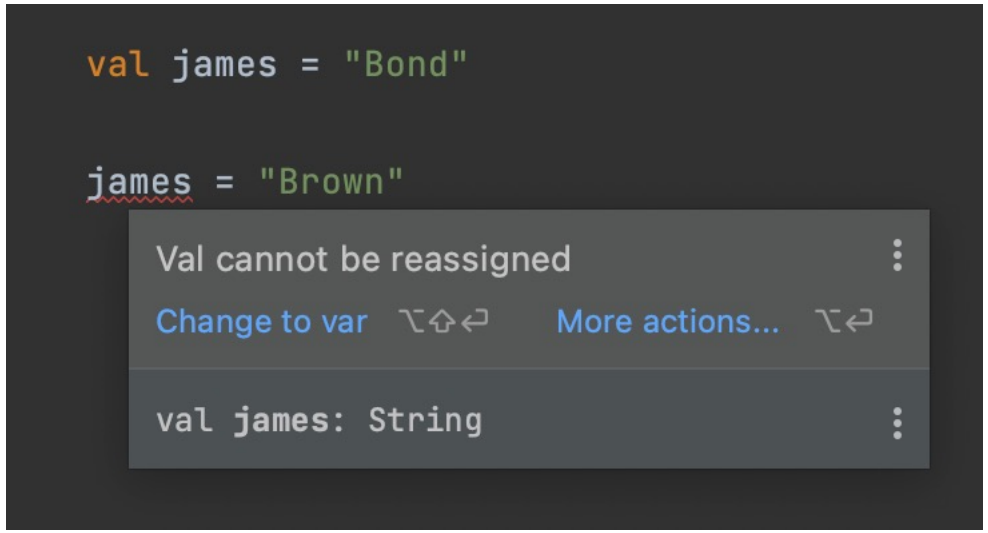
Kotlin

```
val x = 0
val y = "Null"
var z: Int? = 7
var james: String = "Bond"
val license = 287.781f
```

- von Jetbrains
- seit 2016 stabil
- strongly typed
- null safety

- Android (seit 2019 bevorzugt)
- Backend

var oder val



val

Wert („value“)
wird am häufigsten verwendet
kann nur einmal befüllt werden

var

veränderlich („variable“)
Nur verwenden wenn wirklich neu befüllt wird
Kann beliebig oft mit gleichem Datentyp befüllt werden

Boolean

```
val george: Boolean = true

val boole = false
```

Boolean

```
val a = true
val b = false

(a && b) //false

(a || b) //true

(!a) //false
```

Konjunktion (UND)

∧	**0**	**1**
0	0	0
1	0	1

Disjunktion (ODER)

∨	**0**	**1**
0	0	1
1	1	1

Negation (NICHT)

	¬
0	1
1	0

Quelle: https://de.wikipedia.org/wiki/Boolesche_Algebra

Integer

Ganze Zahl

-2 147 483 648 bis 2 147 483 647

(Long für noch größere Zahlen)

```
val x = 7

val y: Int = 7
```

Double

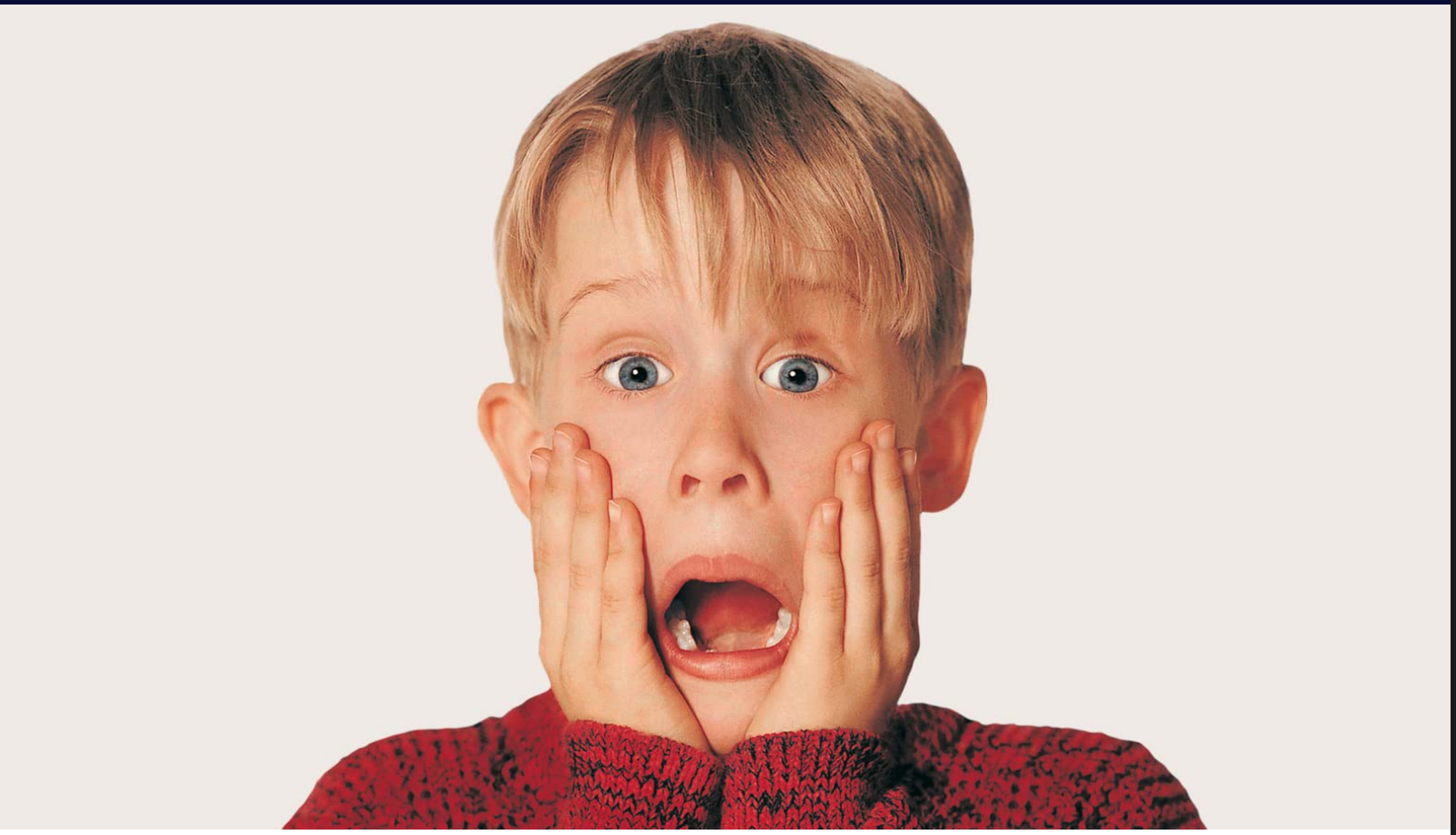
Gleitkommazahl

15 - 16 Dezimalstellen

als Float nur 6-7

```
val pi: Double = 3.14159265359
val euro = 13.7603
var desk = 89.74f
```

Konvertieren



Quelle: https://m.media-amazon.com/images/

```
toByte(): Byte

toShort(): Short

toInt(): Int

toLong(): Long

toFloat(): Float

toDouble(): Double

toChar(): Char
```

Char

Einzelnes Zeichen

```
val moveForward: Char = 'w'

val useItem = 'f'
```

String

Zeichenkette

```
val intro: String = "A long time ago in a galaxy far, far away...."
```

List

Ansammlung gleicher DatenTypen

wird grundsätzlich bevorzugt weil dynamische Größe

List<T> unveränderbare Liste (vergleichbar mit val)

MutableList<T> veränderbare Liste (vergleichbar mit var)

```
val books: MutableList<Book> = mutableListOf(bookOne, bookTwo)

val favouriteNumbers = listOf<Int>(1, 3, 24, 6)
```

List - Funktionen

Unzählige praktische Funktionen

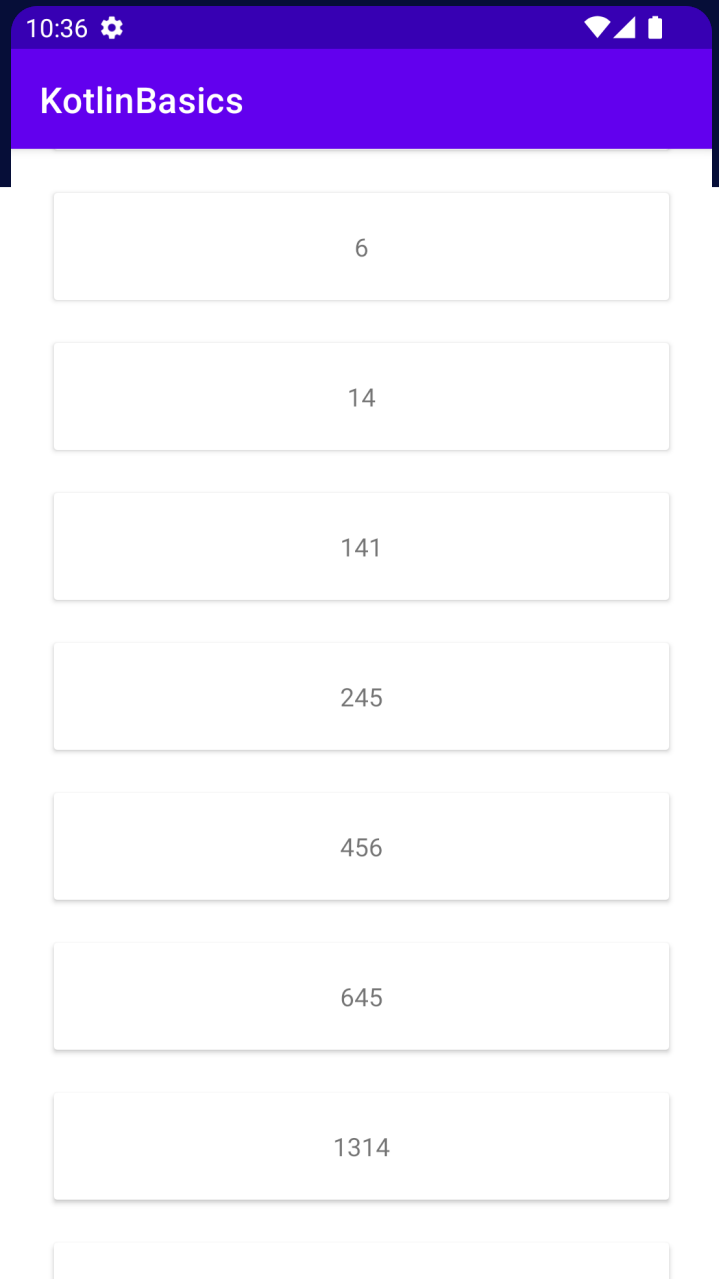
https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/

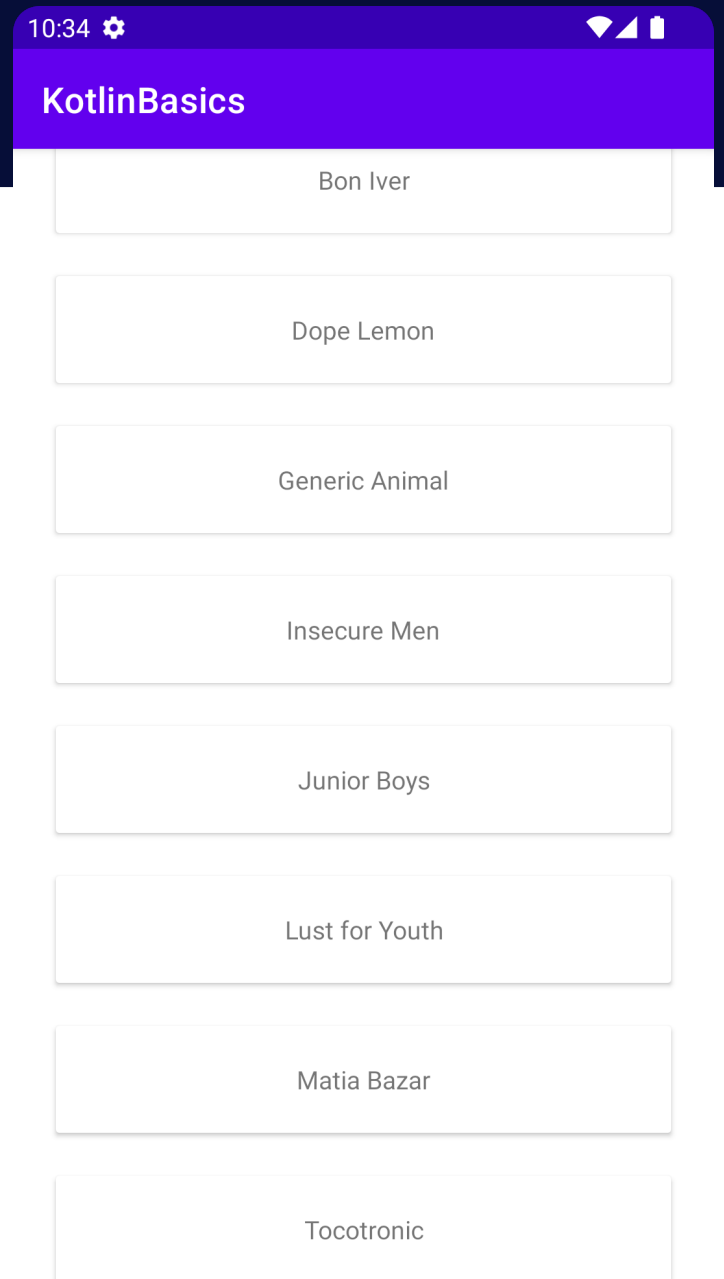
```
val numbers = mutableListOf<Int>(2424, 0, 534, 12)

numbers.add(3)
numbers.remove( element: 12)
numbers[2]
numbers.sort()
numbers.binarySearch( element: 0)
```

Live Beispiel

Verschiedenste Listen





Syntax Institut

Viel Spaß!

Quelle: https://images.indianexpress.com/2019/08/home-alone-759.jpg