Demo storyline

Setup Project

1. New Project
2. Install Kotlin Plugin
3. Configure Kotlin in Project
4. Create `/app/src/main/kotlin`

Hallo Welt

- main <tab>
- println ("Hallo Welt!")

```
fun main(args: Array<String>) {
    println("Hallo Welt!")
}
```

- Shift-F10

> - Top-Level functions
> - Prägnanz: Kein Semikolon

Hallo Kotlin

```
import java.util.*

fun main(args: Array<String>) {
    val name = if (args.size > 0) args[0] else "Publikum"
    val zuschauer = Gast(name, title = Title.wertes)

    println(zuschauer)
    println("Hallo ${zuschauer.title} ${zuschauer.name}")
}

data class Gast(val name: String, var zeit: Date = Date(), val title: Title?)
enum class Title { Herr, Frau, wertes }
```

1. String interpolation
2. `if`-Expression
3. Enum & Data Class
4. String interpolation erweitern, `println()`
    - Alt-Shift-F10 - Configure run

- Prägnanz: Kein Semikolon
- Kein `new`,
- `if` ist ein Ausdruck
- Properties
- `val` und `var`
- ggf. `Nullify?`

Hello Null

```
class Greeter(val s: String) {
    fun doHello() {
        println(toUp())
    }

    fun toUp(): String = s.toUpperCase()
}

fun main(args: Array<String>) {
    Greeter("Leute!").doHello()
    Greeter(null).doHello()
}
```

1. `Greeter(null).doHello()` und `String?`
2. `!!`
3. `fun toUp(): String? = s?.toUpperCase()`
4. `fun toUp(): String = s?.toUpperCase() ?: "NULL"`
5. Smart Cast

```
    fun toUp(): String {
        return if (s != null) {
            s.toUpperCase()
        } else {
            "null"
        }
    }
```

Hello Null mit Java

1. Java Klasse:

```
public class JavaClass {
    public String strValue() {
        return "Welt";
    }
}
```

1. `String?` → `String`,
   `null` → `JavaClass.strValue()`
2. Klasse null returned lassen

Null Ausgebaut:

```
    fun toUpper1(): String = nullable!!.toUpperCase()
    fun toUpper2(): String? = nullable?.toUpperCase()
    fun toUpper3(): String = nullable?.toUpperCase() ?: "NULL"
    fun toUpper4(): String = nullable?.toUpperCase() ?: throw
IllegalStateException("Nah!")
    fun toUpper5(): String? {
        if (nullable != null) {
            return nullable.toUpperCase()
        } else {
            return "NULL5"
        }
    }
    fun javaString(): String = System.getenv("PWD")
```

Idiome

Ausgang

```
public class WeatherIndicator {
    void rateWeather(int celsius) {
        String status;
        Color color;
        if (celsius < 5) {
            status = "Saukalt!";
            color = Color.BLUE;
        } else if (celsius >= 5 && celsius <= 20) {
            status = "Geht so!";
            color = Color.ORANGE;
        } else {
            status = "Urlaub!";
            color = Color.RED;
        }
    }

    enum Color {BLUE, ORANGE, RED}
}
```

Ergebnis

```
class WeatherIndicatorKt {
    fun rateWeather(celsius: Int) {
        val (status, color) =
            when {
                celsius < 5 -> Pair("Saukalt!", Color.BLUE)
                celsius in 5..20 -> Pair("Geht so!", Color.ORANGE)
                else -> Pair("Urlaub!", Color.RED)
            }
    }

    enum class Color { BLUE, ORANGE, RED }
}
```

Nullable in Android

1. Convert Activity to Kotlin
2. TextView ID zuweisen
3. Ressourcenstring einfügen
4. Verdrahten

activity_main.xml

```
...
    <TextView
        android:id="@+id/hellolabel"
...
```

strings.xml

```
<resources>
    <string name="app_name">My Application</string>
    <string name="meinGruss">Hallo liebe Kotliner!</string>
</resources>
```

MainActivity.kt

```
package de.exxcellent.myapplication

import android.app.Activity
import android.os.Bundle
import android.widget.TextView
import kotlin.properties.Delegates

class MainActivity : Activity() {

    //var myString: String by Delegates.notNull()
    lateinit var myString: String

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        myString = getString(R.string.meinGruss)
        val helloView = findViewById(R.id.hellolabel) as TextView
        helloView.setText(myString)
    }
}
```

Kotlin Android Extension

build.gradle

```
apply plugin: 'kotlin-android-extensions'
```

MainActivity.kt

```
import kotlinx.android.synthetic.main.activity_main.*
...
 hellolabel.setText(myString)
```

Extension Methods

android-extension.kt

```
import android.app.Activity
import android.widget.Toast

fun Activity.toast(message: String, duration: Int = Toast.LENGTH_SHORT) {
    Toast.makeText(applicationContext, message, duration).show()
}
```

MainActivity.kt

```
    hellolabel.setOnClickListener { toast("Hi") }
```

Zeigt:

- Lambda / SAM
- Extension Methods

Anko DSL

build.gradle

```
    compile 'org.jetbrains.anko:anko-sdk19:0.9'
```

MainActivity.kt

```
package de.exxcellent.myapplication

import android.app.Activity
import android.os.Bundle
import org.jetbrains.anko.*
import kotlin.properties.Delegates

class MainActivity : Activity() {

    lateinit var myString: String

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        myString = getString(R.string.meinGruss)

        verticalLayout {
            padding = dip(16)
            textView {
                text = myString
                onClick { toast("Hi")  }
            }
        }
    }
}
```