

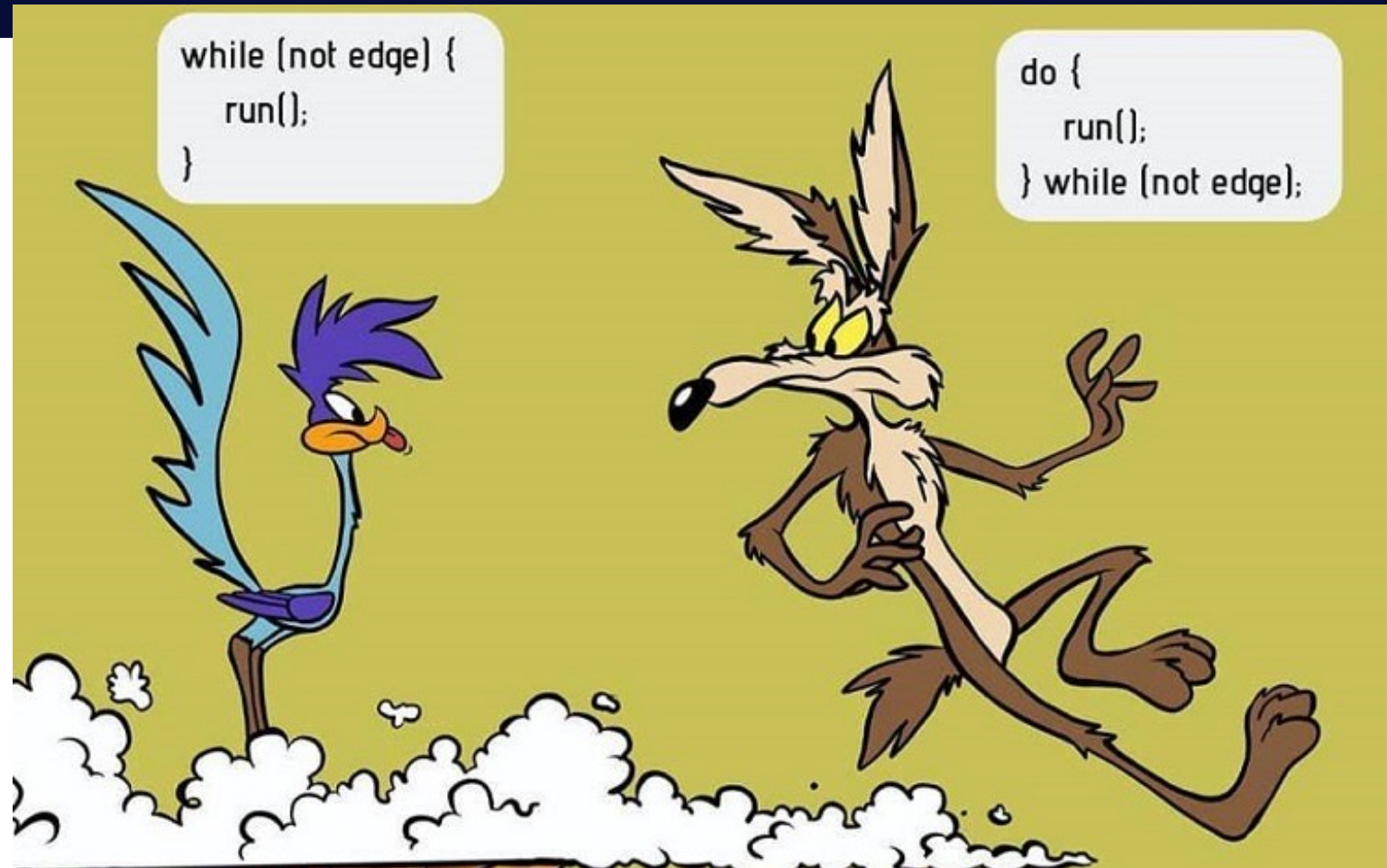
Modul 3 – Android App Entwicklung mit Kotlin

# Kotlin Basics II

(Bedingungen, Schleifen und Funktionen)

# Gliederung

- Bedingungen
- Schleifen
- Funktionen



Quelle: <https://i0.wp.com/bytEfreaks.net/wp-content/uploads/2019/08/cropped-Wile-E-Coyote-and-Roadrunner.png?fit=2559%2C1440&ssl=1>

# Bedingungen

## if - else

wenn - **if**- Bedingung erfüllt mach das  
sonst mach das andere - **else**

```
if (thirsty) {  
    // get drink  
} else {  
    // don't  
}
```

```
val wantDrink = if (thirsty) "ja bitte" else "nein danke"
```

# Bedingungen

## when

wenn es Option 1 entspricht mach das -->

wenn es Option 2 entspricht mach das -->

usw..

Sonst mach das - else ->

```
when (thirstLevel) {  
    1 -> getSmallDrink()  
    in 2..5 -> getMediumDrink()  
    in 6..10 -> getSomethingBig()  
    else -> {  
        // save some money  
    }  
}
```

# Schleifen

## for

solange sich Variable im Rahmen befindet  
mach das

(am häufigsten verwendet)

```
val againNumbers = listOf<Int>(6, 33, 6, 24, 3, 32)
```

```
for (number in againNumbers) {  
    println(number)  
}
```

```
for (i in 1..10) {  
    println(i)  
}
```

```
for (i in 10 downTo 0 step 2) {  
    println(i)  
}
```

```
for (i in againNumbers.indices) {  
    println(againNumbers[i])  
}
```

```
for ((index, value) in againNumbers.withIndex()) {  
    println("Das Element an Stelle $index hat den Wert $value.")  
}
```

# Schleifen

## while

solange Bedingung erfüllt mach das

## do - while

mach das solange Bedingung erfüllt  
(Läuft vor Check)

```
while (notEdge) {  
    run()  
}
```

```
do {  
    run()  
} while (notEdge)
```

# Schleifen

Eine Frau ruft ihren Programmierer Ehemann an und sagt: „Solange du draußen bist, kauf doch Milch.“

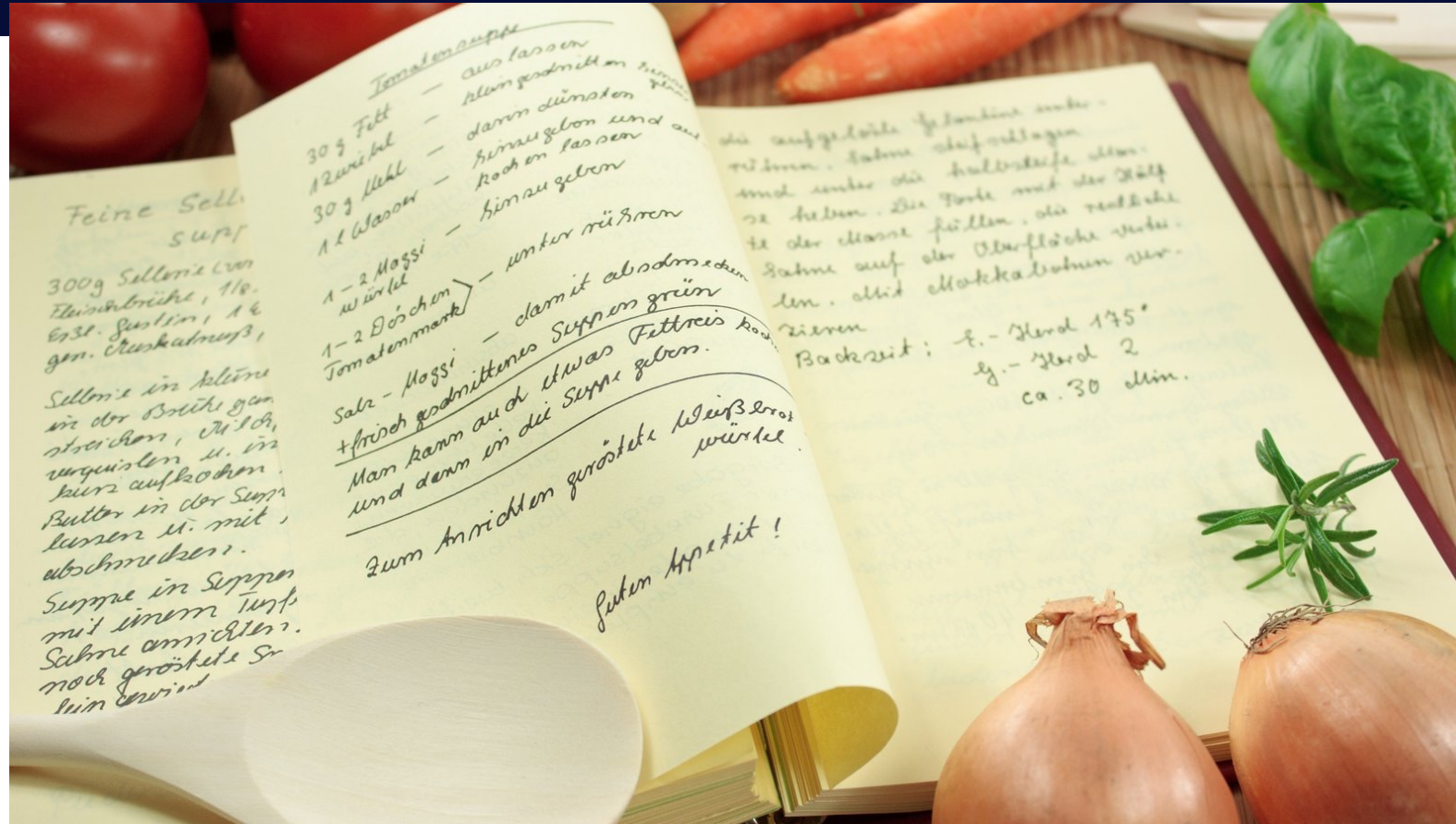
Er kam nie nach Hause.



Quelle: <https://i.pinimg.com/originals/47/c1/49/47c149e85742473433a3c16cc1cc0811.jpg>



# Funktionen



Quelle: [https://www.zugutfuerrdielonne.de/fileadmin/\\_processed\\_/e/c/csm\\_AdobeStock\\_33591155\\_Kochbuch\\_Rezept\\_Dalmatino\\_8c99df77d2jpeg](https://www.zugutfuerrdielonne.de/fileadmin/_processed_/e/c/csm_AdobeStock_33591155_Kochbuch_Rezept_Dalmatino_8c99df77d2jpeg)



# Funktionen

Standardmäßig immer für alle nutzbar - **public**

**private fun** - nur innerhalb der Klasse nutzbar

```
fun divide(x: Int, y: Int): Int {  
    val result = x / y  
    return result  
}
```

```
divide(x: 4, y: 2)
```

```
fun cook(  
    water: Essential,  
    flour: Ingredient,  
    oil: Ingredient,  
    salt: Essential  
) {  
    // make pizza dough  
}
```

# Funktionen

`findViewById<Typ>(R.id.Id_im_Layout)` - ist eine von der Activity schon vorgefertigte Funktion

Verknüpft Element (View) aus `activity_main.xml`

mit Variable im Kotlin Code in `MainActivity.kt`

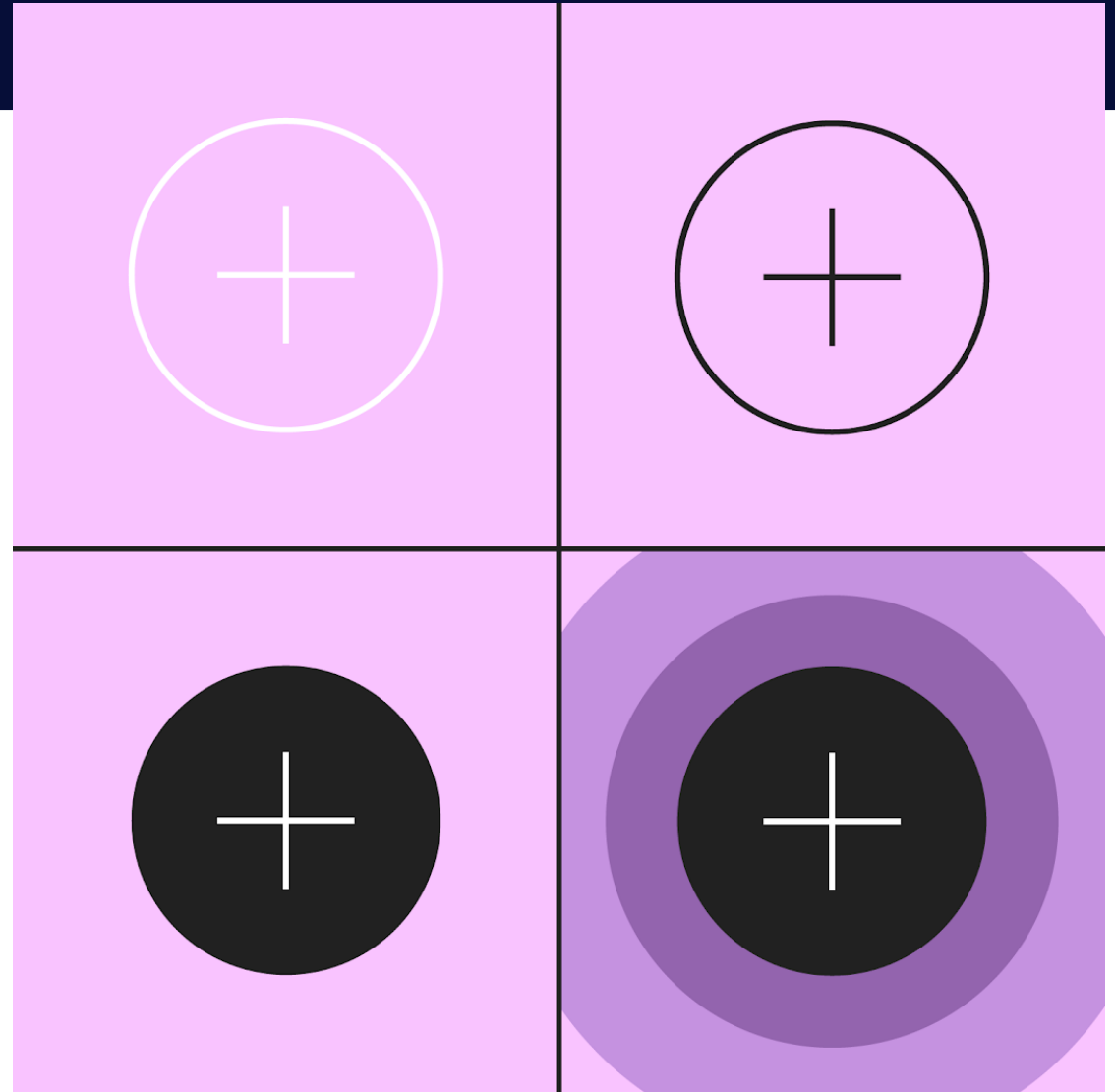
```
val backButton = findViewById<Button>(R.id.zuruck_button)
```

# Pfeilfunktionen

## Lamdas

Um Funktionen in einer Variable zu speichern oder  
um sie einer Funktion als Parameter zu übergeben

Meistens für User-Interaktion verwendet



# Pfeilfunktionen

$(A, B) \rightarrow C$

Funktion mit Parameter Typ A und Typ B gibt Typ C zurück

$() \rightarrow A$

Funktion ohne Parameter gibt Typ A zurück

$(A) \rightarrow \text{Unit}$

Funktion mit Parameter Typ A gibt Irgendwas zurück

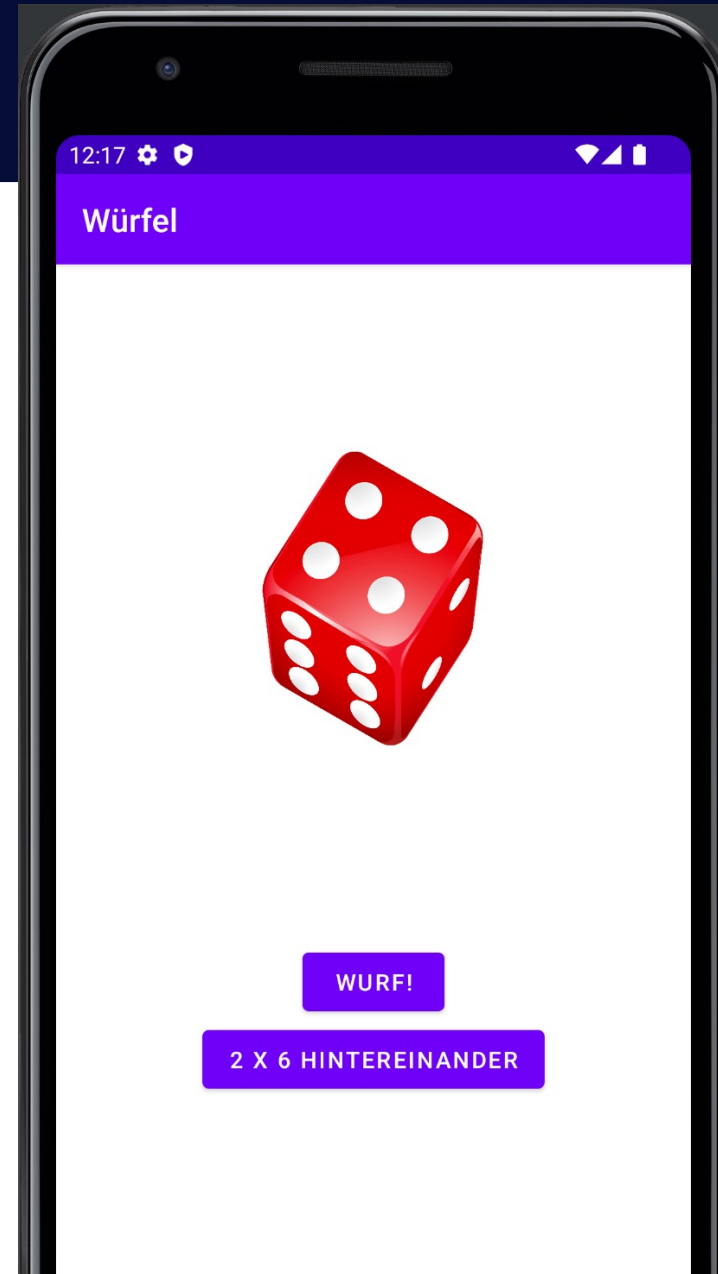
```
val colorToBlue: (View) -> Unit = { l: View -> l.setBackgroundColor(Color.BLUE) }

button.setOnClickListener(colorToBlue)

button.setOnClickListener { view ->
    view.setBackgroundColor(Color.RED)
}
```

# Beispiel: Würfel App

- Bedingungen um zu wissen welche Würfelseite angezeigt werden soll
- Schleife um so oft zu würfeln bis zwei 6er hintereinander kommen
- verpackt in verschiedensten Funktionen



# Kotlin Basics II

## Wiederholung - Was haben wir heute gelernt?

1

Bedingungen

2

Schleifen

3

Funktionen



Google

ANDROID

**BOOT  
LOOPS**



Viel Spaß!