

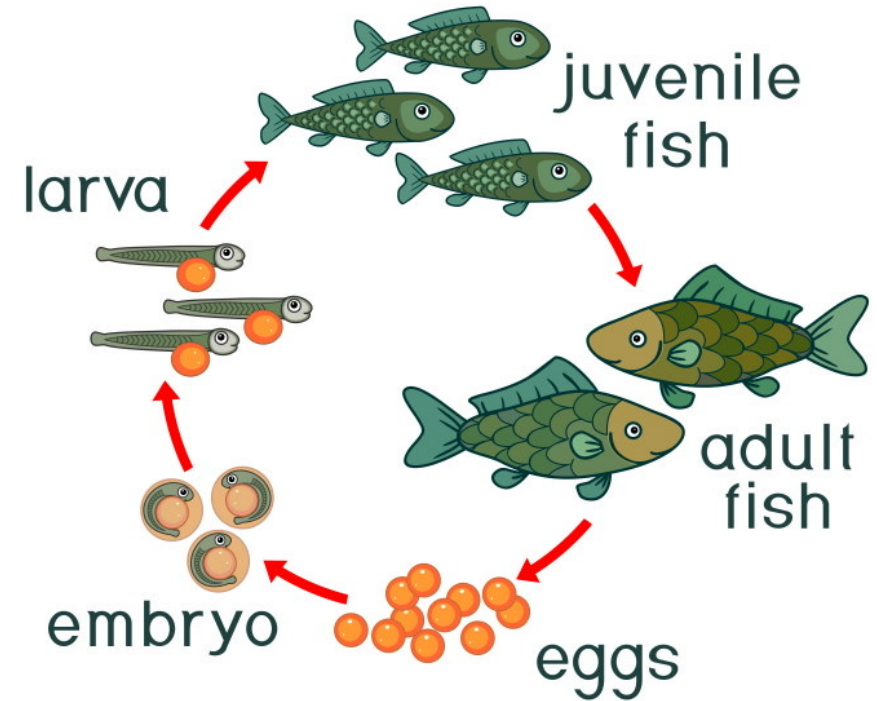
Modul 3 – Android App Entwicklung mit Kotlin

Activity Lifecycle



Gliederung

- Activity Lifecycle
- Zustände und deren Events
- Wann geschieht ein Zustandswechsel?
- Anwendung



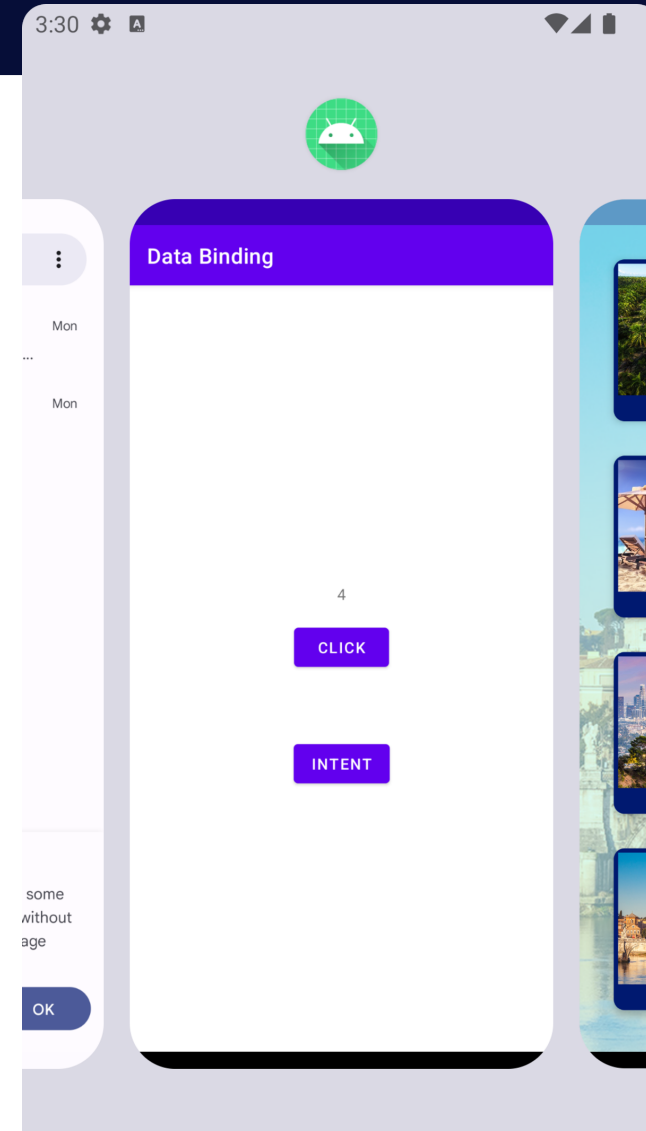
Quelle: <https://www.earthlife.net/fish/development.html>

Activity Lifecycle

Lifecycle - „Lebenszyklus“

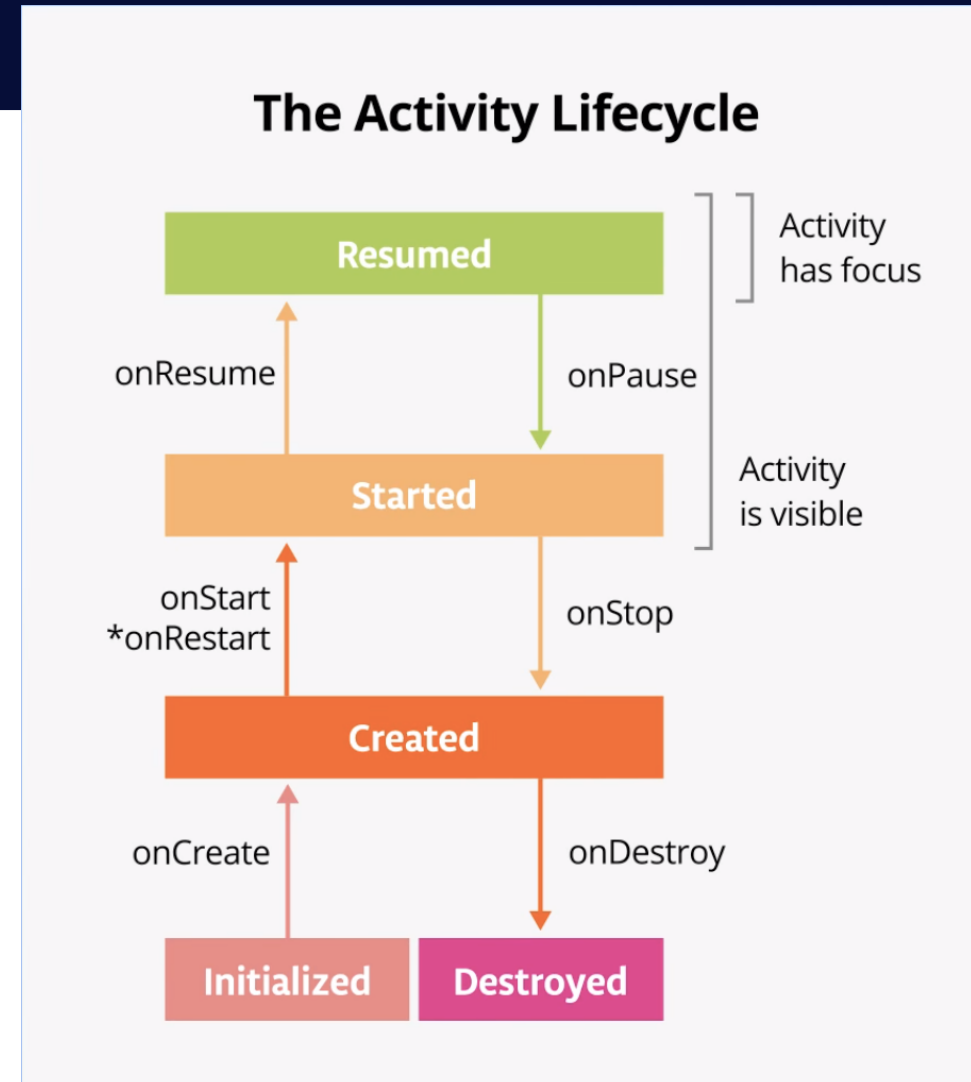
Vom ersten Start der App über verschiedene UserFlows bis zum beenden durchlebt die eine Activity verschiedene Zustände

Wichtig um Verhalten von Activities zu verstehen



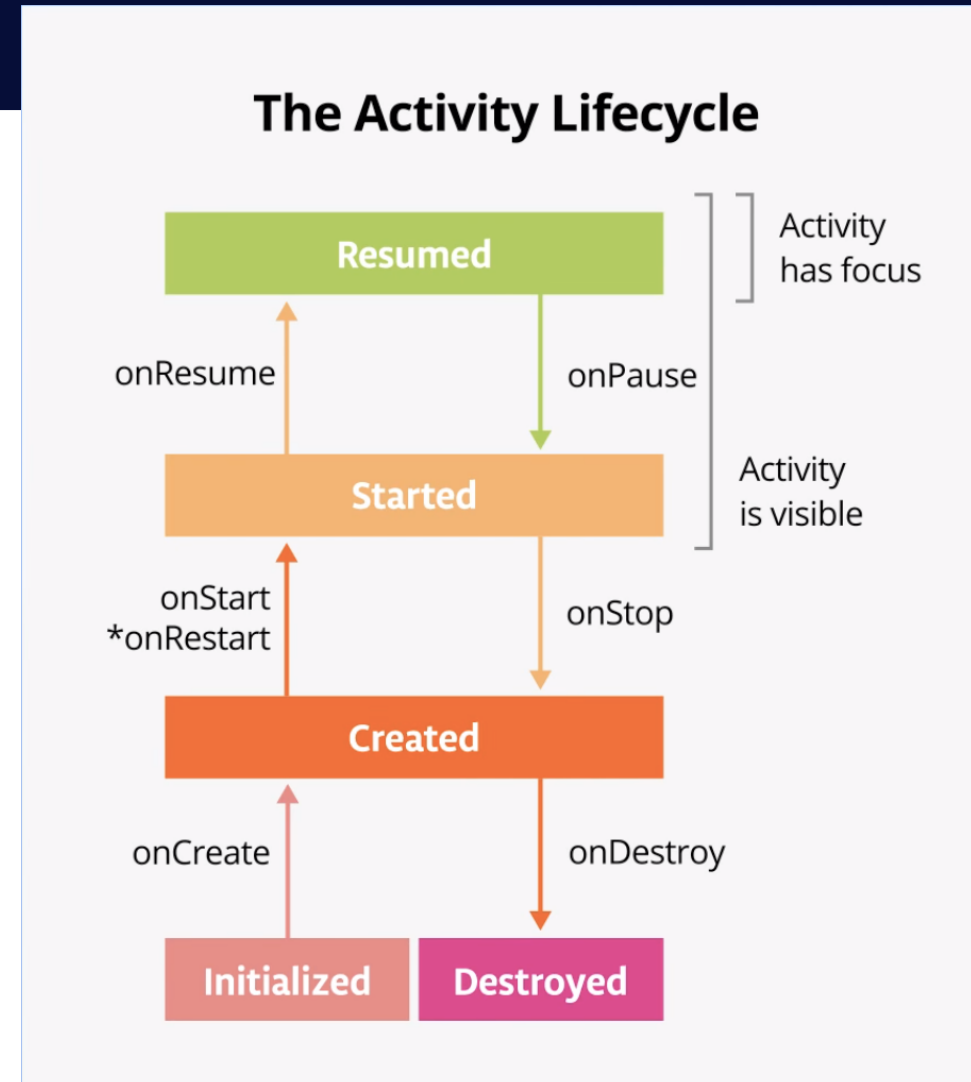
Zustände und Events

- Vorbereitet – **Initialized**
- Erstellt – **Created**
- Gestartet – **Started**
- Fortgesetzt – **Resumed**
- Zerstört – **Destroyed**



Zustandswechsel

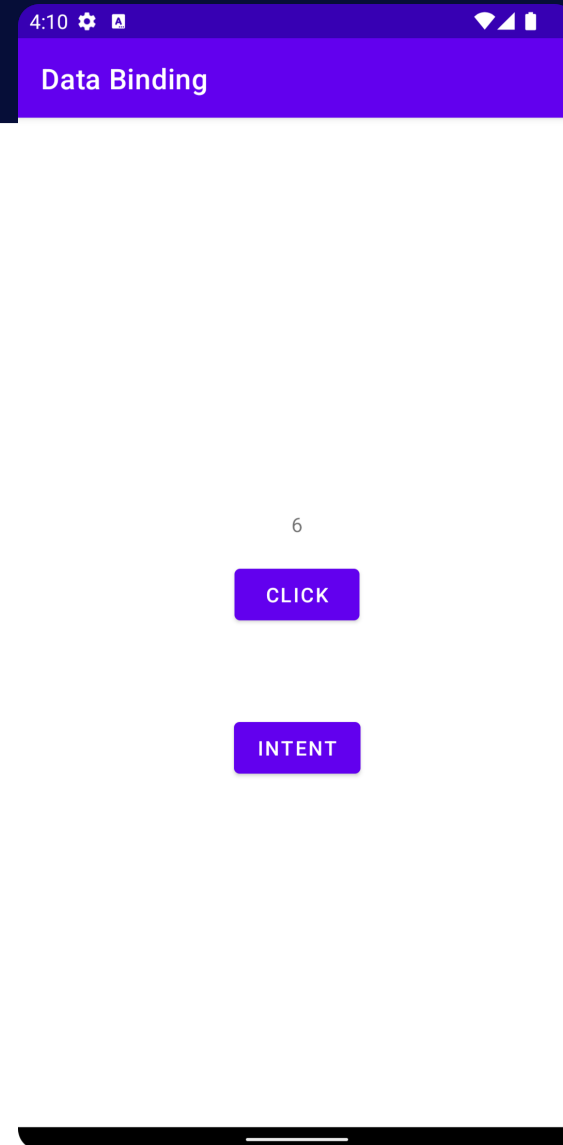
- `onCreate()`
einmaliges erstellen von Variablen, Layout laden, `onClickListeners` usw.
- `onStart() – onStop()`
aufgerufen wenn Apsichtbar/unsichtbar wird
- `onResume() – onPause()`
aufgerufen wenn App fokussiert/defokussiert wird
- `onDestroy()`
wenn App wirklich beendet wird



Quelle: <https://developer.android.com>

Beispiel: Click Counter

- Log Events um Lifecycle zu verfolgen
- Was passiert beim Start der App?
- Was passiert beim Umschalten zwischen Apps?
- Was passiert wenn man das Smartphone dreht?



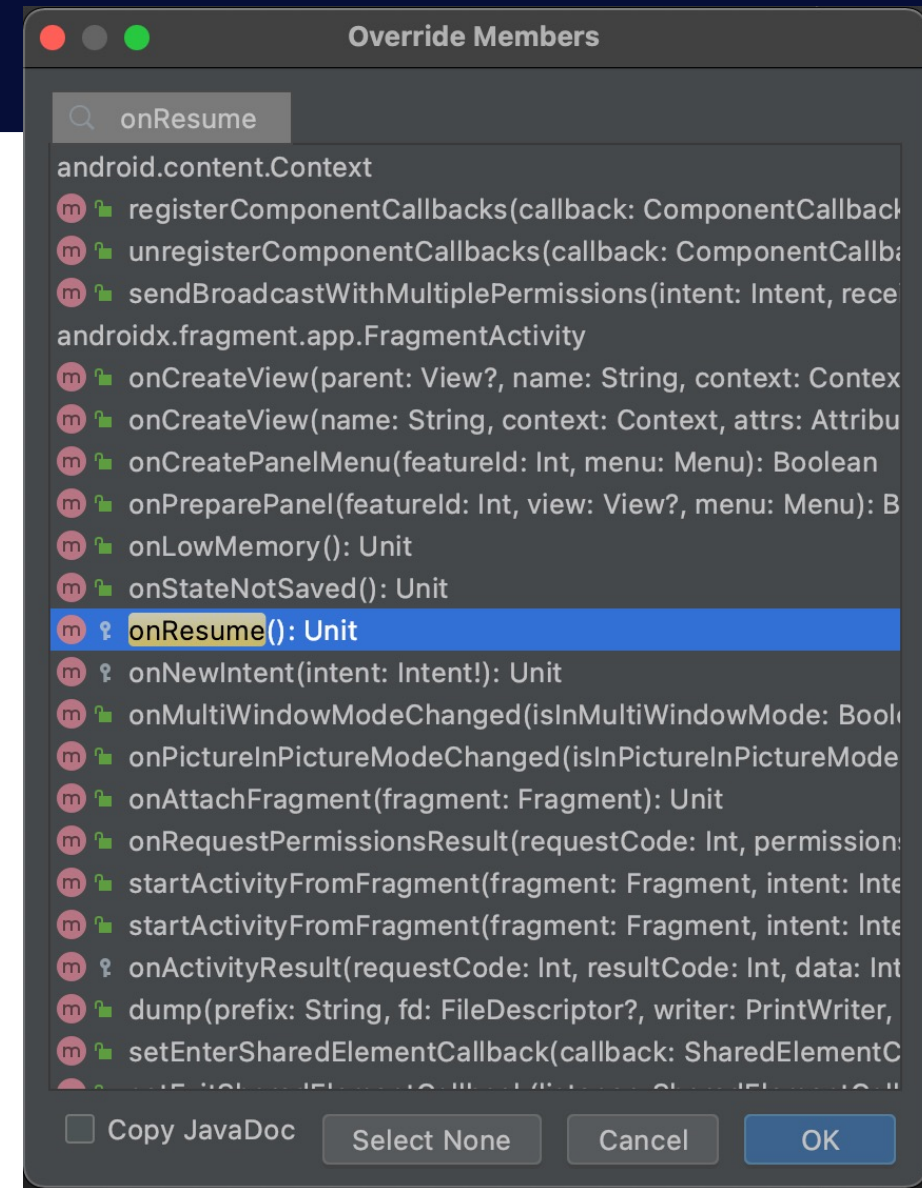
Beispiel: Click Counter

neben `onCreate()` werden auch alle weiteren LifecycleCallbacks mit einer LogNachricht versehen damit wir nachvollziehen können was geschieht

```
class MainActivity : AppCompatActivity() {  
  
    private var counter = 0  
  
    override fun onCreate(savedInstanceState: Bundle?) {...}  
  
    override fun onResume() {  
        super.onResume()  
        Log.e(TAG, msg: "onResume called")  
    }  
  
    override fun onPause() {  
        super.onPause()  
        Log.e(TAG, msg: "onPause called")  
    }  
  
    override fun onStop() {  
        super.onStop()  
        Log.e(TAG, msg: "onStop called")  
    }  
  
    override fun onDestroy() {  
        super.onDestroy()  
        Log.e(TAG, msg: "onDestroy called")  
    }  
  
    override fun onRestart() {  
        super.onRestart()  
        Log.e(TAG, msg: "onRestart called")  
    }  
}
```

Beispiel: Click Counter

mittels **Ctrl + O** kann eine überschreibbare Funktion der Activity auswählen und sie wird automatisch erstellt

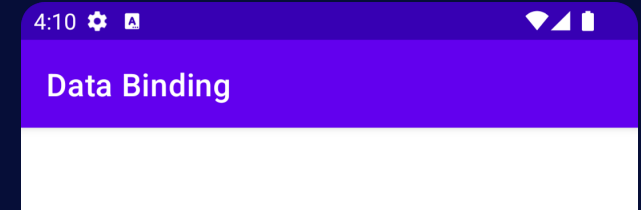


Beispiel: Click Counter

im Log kann nun der Activity Lifecycle
verfolgt werden

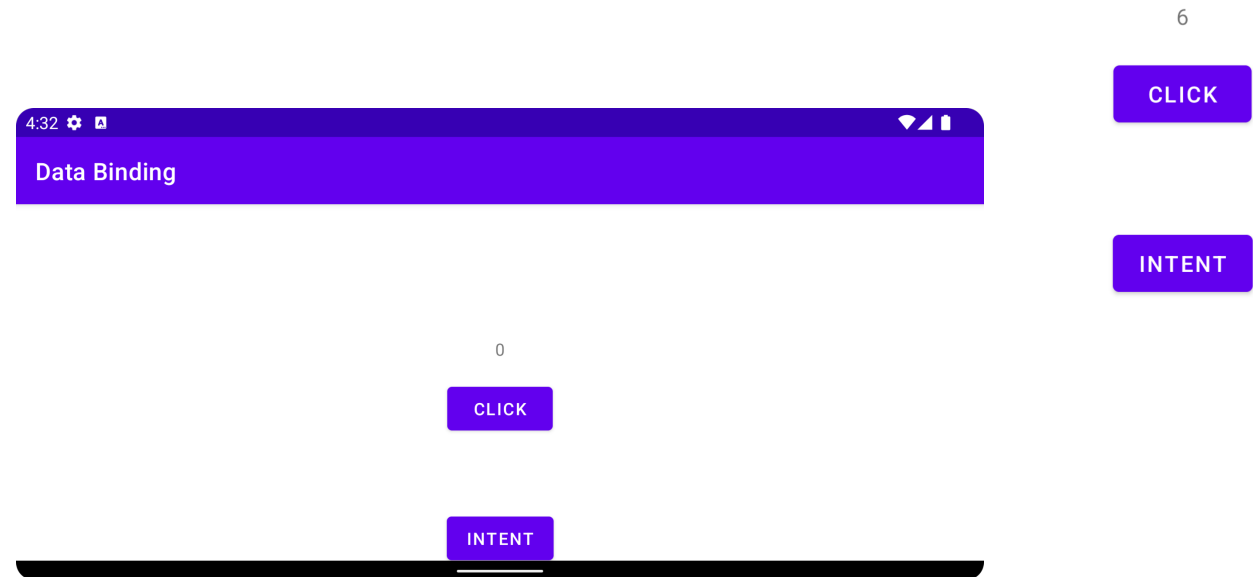
```
2022-03-29 16:25:32.147 9452-9452/com.example.databinding E/MainActivity: onCreate called
2022-03-29 16:25:32.190 9452-9452/com.example.databinding E/MainActivity: onResume called
2022-03-29 16:25:38.750 9452-9452/com.example.databinding E/MainActivity: onPause called
2022-03-29 16:25:39.125 9452-9452/com.example.databinding E/MainActivity: onStop called
2022-03-29 16:25:42.973 9452-9452/com.example.databinding E/MainActivity: onRestart called
2022-03-29 16:25:42.973 9452-9452/com.example.databinding E/MainActivity: onResume called
2022-03-29 16:26:05.687 9452-9452/com.example.databinding E/MainActivity: onPause called
2022-03-29 16:26:05.688 9452-9452/com.example.databinding E/MainActivity: onStop called
2022-03-29 16:26:05.688 9452-9452/com.example.databinding E/MainActivity: onDestroy called
2022-03-29 16:26:05.715 9452-9452/com.example.databinding E/MainActivity: onCreate called
2022-03-29 16:26:05.732 9452-9452/com.example.databinding E/MainActivity: onResume called
2022-03-29 16:26:14.934 9452-9452/com.example.databinding E/MainActivity: onPause called
2022-03-29 16:26:14.935 9452-9452/com.example.databinding E/MainActivity: onStop called
2022-03-29 16:26:14.935 9452-9452/com.example.databinding E/MainActivity: onDestroy called
```

Anwendung



Wenn die App in die Queransicht wechselt
wird sie zerstört und neu erstellt

UI Daten gehen damit auch verloren



onSaveInstanceState()

zum Zwischenspeichern von kleinen
Datenmengen

wird jedes mal aufgerufen wenn App in
den Hintergrund wechselt und droht bald
zerstört zu werden

```
override fun onSaveInstanceState(outState: Bundle) {  
    super.onSaveInstanceState(outState)  
    Log.e(TAG, msg: "onSaveInstanceState called")  
  
    outState.putInt(KEY_COUNTER, counter)  
}
```

savedInstanceState

in `onCreate()` können aus
`savedInstanceState` gespeicherte Daten
über ihren Key wiedergewonnen werden

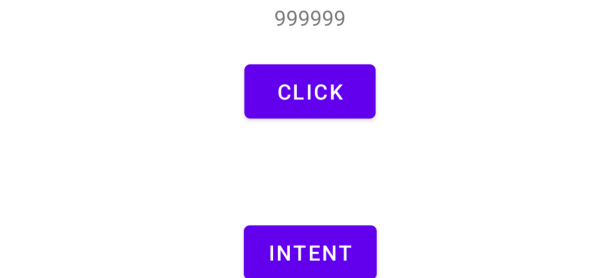
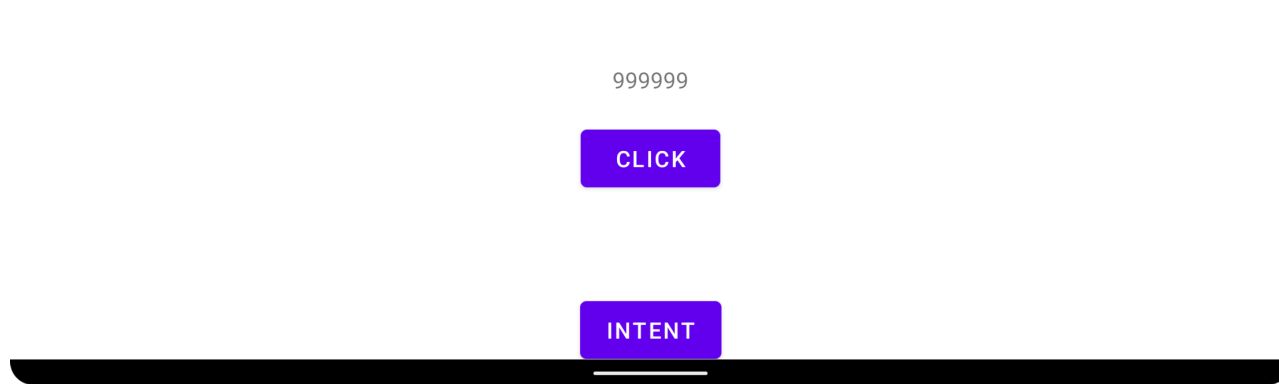
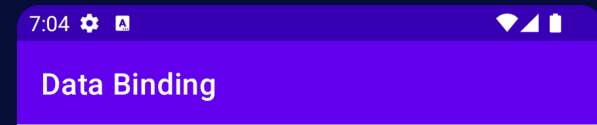
```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    Log.e(TAG, msg: "onCreate called")  
  
    if (savedInstanceState != null) {  
        counter = savedInstanceState.getInt(KEY_COUNTER, defaultValue: 0)  
    }  
  
    val binding: ActivityMainBinding =  
        DataBindingUtil.setContentView(activity: this, R.layout.activity_main)  
  
    binding.counting = counter.toString()  
  
    binding.button.setOnClickListener {...}  
  
    binding.intentButton.setOnClickListener {...}  
}
```

savedInstanceState

in `onCreate()` können aus
`savedInstanceState` gespeicherte Daten
über ihren Key wiedergewonnen werden

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    Log.e(TAG, msg: "onCreate called")  
  
    if (savedInstanceState != null) {  
        counter = savedInstanceState.getInt(KEY_COUNTER, defaultValue: 0)  
    }  
  
    val binding: ActivityMainBinding =  
        DataBindingUtil.setContentView(activity: this, R.layout.activity_main)  
  
    binding.counting = counter.toString()  
  
    binding.button.setOnClickListener {...}  
  
    binding.intentButton.setOnClickListener {...}  
}
```

Fertig



Activities Lifecycle

Wiederholung - Was haben wir heute gelernt?

1

Activity Lifecycle

2

Zustände und Events

3

SaveInstanceState

Viel Spaß!



TM

HIGH SCORES

👑

1ST	FANCY	1202860
2ND	JORGE M	271040
3RD	ALESSAN	257945
4TH	JD	239870
5TH	CARLO	225300
6TH	PT	213640
7TH	IK	210560
8TH	CA	208190
9TH	ICWI	192640
10TH	CARLO	181460

Quelle: https://www.reddit.com/r/DaveAndBusters/comments/cnzz6i/ive_peaked_on_my_space_invaders_frenzy_high_score/