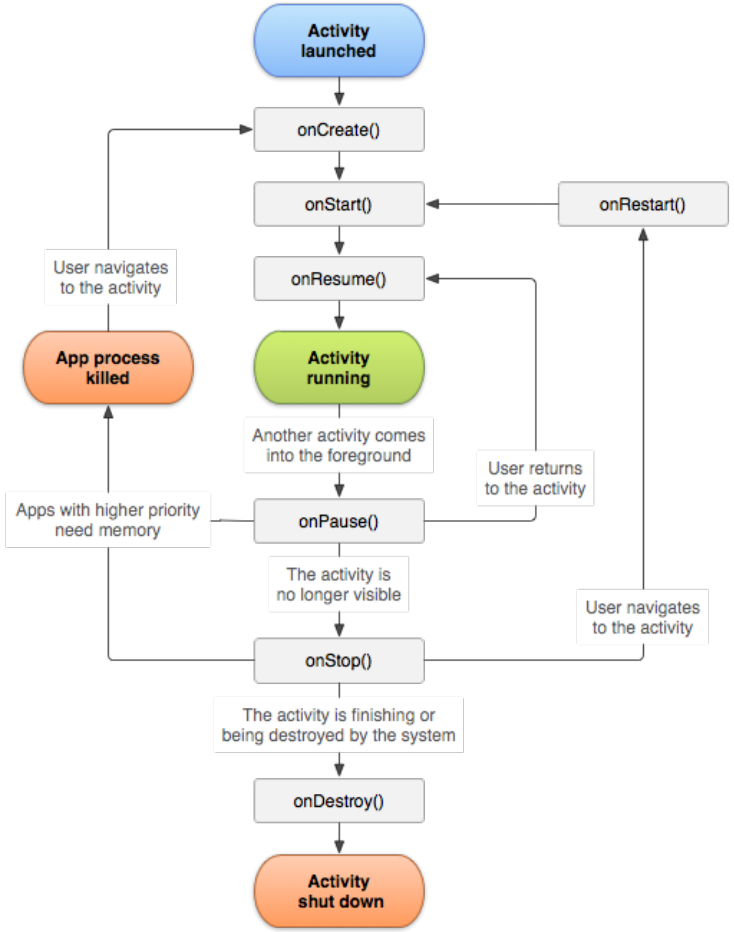
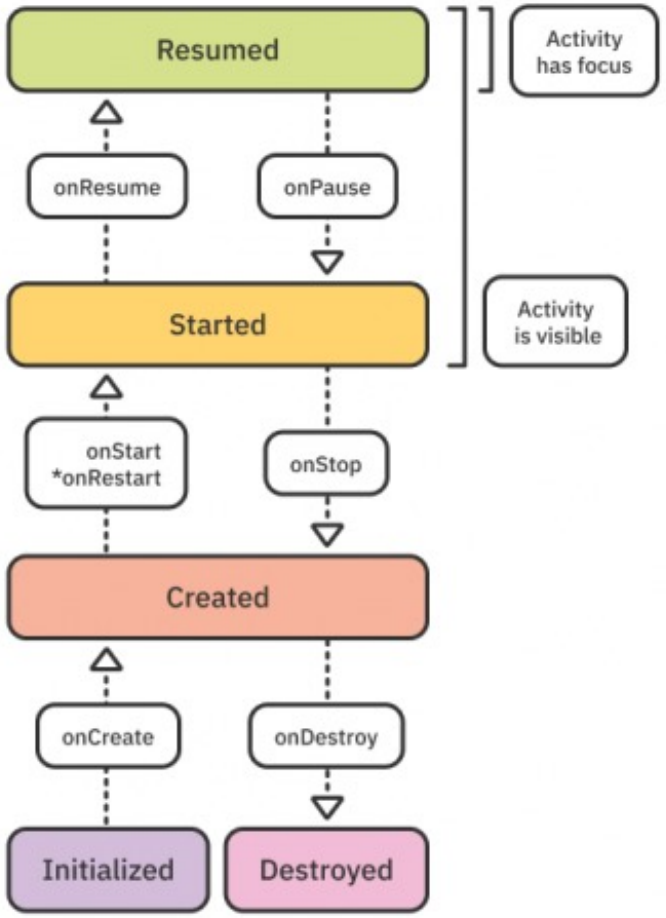
Ciclo di vita delle Activity

Log, Toast

Ciclo di vita





- Testiamo il comportamento di un'activity durante il suo intero ciclo di vita
  1. Creiamo una nuova app
  2. Nella `MainActivity`, implementiamo tutti i metodi legati al lifecycle
  3. Inseriamo un **Log** e un **Toast** in ogni metodo
  4. Testiamo l'app

Log

- Android fornisce la classe **Log** per effettuare logging con vari livelli di priorità:

Priorità	**Metodo**
ERROR	`Log.e()`
WARN	`Log.w()`
INFO	`Log.i()`
DEBUG	`Log.d()`
VERBOSE	`Log.v()`

Log

- API

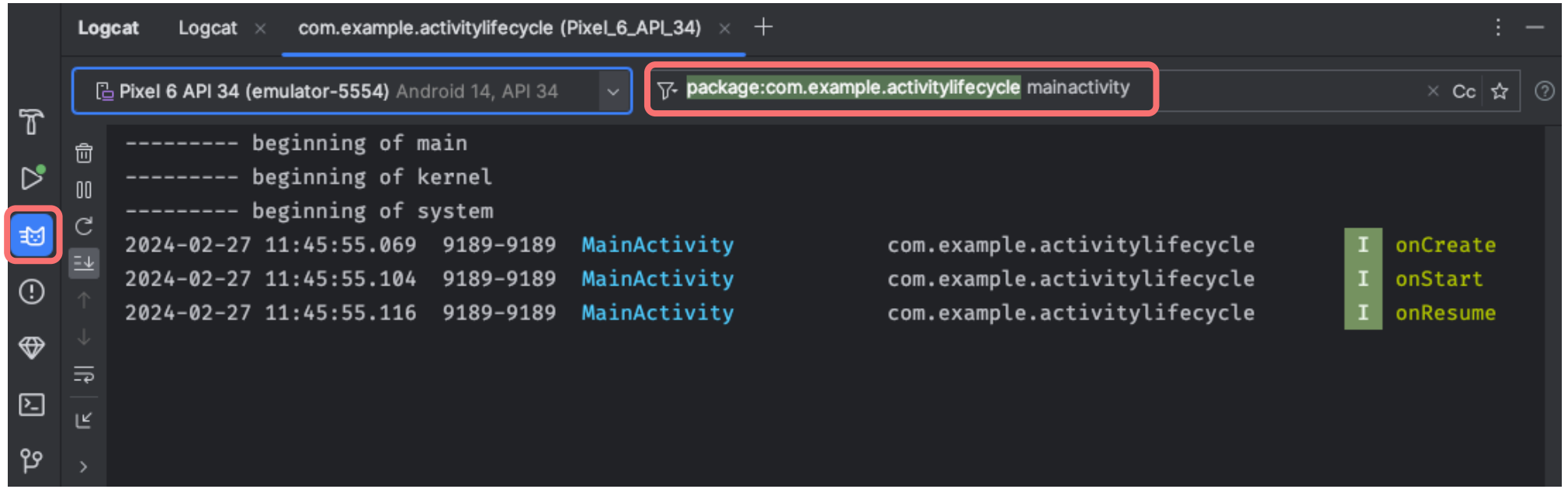
```
Log.i(TAG, ”Hello, World!")
```

- Una convenzione è quella di creare il tag come costante nello stesso file dell’activity

```
private const val TAG = "MyActivity"
```

Test

- Avviamo l’app
- Apriamo il tab Logcat
- Filtriamo i log in base al nome del nostro tag o dell’activity



Test – Avvio dell’app

- I primi tre metody di lifecycle sono già stati chiamati in seguito all’avvio dell’app

```
2024-02-27 10:57:14.931  8596-8596  MainActivity            com.example.activitylifecycle    I  onCreate
2024-02-27 10:57:14.957  8596-8596  MainActivity            com.example.activitylifecycle    I  onStart
2024-02-27 10:57:14.959  8596-8596  MainActivity            com.example.activitylifecycle    I  onResume
```

Test – Tasto/gesture Home

- Cosa succede se torniamo alla home (utilizzando l'apposito tasto o gesture)?

Test – Tasto/gesture Home

- Cosa succede se torniamo alla home (utilizzando l'apposito tasto o gesture)?

```
2024-02-27 10:58:11.402  8596-8596  MainActivity          com.example.activitylifecycle    I  onPause
2024-02-27 10:58:11.441  8596-8596  MainActivity          com.example.activitylifecycle    I  onStop
```

Test – Ri-apertura app

- Cosa succede se riapriamo l’app (facendo tap sulla sua icone o tramite le app recenti)?

Test – Ri-apertura app

- Cosa succede se riapriamo l’app (facendo tap sulla sua icone o tramite le app recenti)?

```
2024-02-27 10:58:14.008  8596-8596  MainActivity  com.example.activitylifecycle  I  onRestart
2024-02-27 10:58:14.008  8596-8596  MainActivity  com.example.activitylifecycle  I  onStart
2024-02-27 10:58:14.009  8596-8596  MainActivity  com.example.activitylifecycle  I  onResume
```

Test – Tasto/gesture Indietro

- Cosa succede se utilizziamo il tasto o la gesture per andare indietro?

Test – Tasto/gesture Indietro

- Cosa succede se utilizziamo il tasto o la gesture per andare indietro?
  - Android >= 12 (nella maggior parte dei casi)

```
2024-02-27 10:58:11.402  8596-8596  MainActivity  com.example.activitylifecycle  I  onPause
2024-02-27 10:58:11.441  8596-8596  MainActivity  com.example.activitylifecycle  I  onStop
```

  - Android < 12

```
2024-02-27 10:59:26.362  8596-8596  MainActivity  com.example.activitylifecycle  I  onPause
2024-02-27 10:59:26.367  8596-8596  MainActivity  com.example.activitylifecycle  I  onStop
2024-02-27 10:59:26.375  8596-8596  MainActivity  com.example.activitylifecycle  I  onDestroy
```

https://developer.android.com/about/versions/12/behavior-changes-all#back-press

Test – Rotazione del dispositivo

- Cosa succede se ruotiamo il dispositivo?

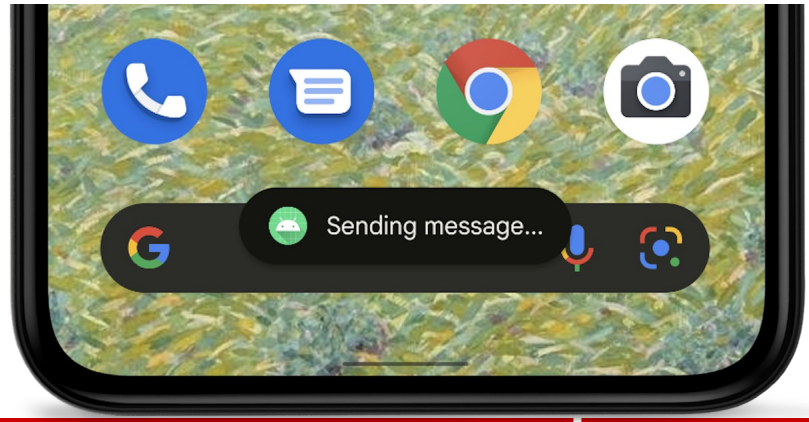
Test – Rotazione del dispositivo

- Cosa succede se ruotiamo il dispositivo?

```
2024-02-27 10:59:26.362  8596-8596  MainActivity        com.example.activitylifecycle    I  onPause
2024-02-27 10:59:26.367  8596-8596  MainActivity        com.example.activitylifecycle    I  onStop
2024-02-27 10:59:26.375  8596-8596  MainActivity        com.example.activitylifecycle    I  onDestroy
2024-02-27 10:59:26.418  8596-8596  MainActivity        com.example.activitylifecycle    I  onCreate
2024-02-27 10:59:26.427  8596-8596  MainActivity        com.example.activitylifecycle    I  onStart
2024-02-27 10:59:26.429  8596-8596  MainActivity        com.example.activitylifecycle    I  onResume
```

Toast

- Vogliamo mostrare le informazioni di lifecycle direttamente sul dispositivo
- Possiamo utilizzare i Toast
- Un Toast è un piccolo popup non interattivo, che fornisce un messaggio molto sintetico per un breve periodo di tempo



Toast - API

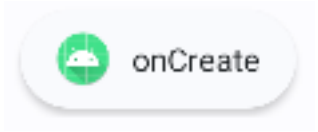
- È possibile mostrare un Toast tramite la classe omonima:

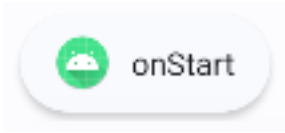
```
Toast.makeText(context, text, duration).show()
```

  - **`context`**: contesto dell'applicazione
  - **`text`**: messaggio da mostrare
  - `duration`: durata del toast, può essere **`Toast.LENGTH_SHORT`** o **`Toast.LENGTH_LONG`**

Test con toast

- Ripetendo i test delle slide precedenti, ora avremo sia i messaggi in Logcat che i Toast sul dispositivo





Accesso al ciclo di vita dell'activity da un Composable

- L'accesso al ciclo di vita dell'activity è necessario per acquisire e rilasciare risorse (es. GPS), ma presenta un problema: rende i composable dipendenti dall'activity in cui vengono utilizzati.
  - Esempio (semplificato): Se un composable che accede al GPS è utilizzato in 10 activity, ognuna di esse deve fare override dei metodi di lifecycle necessari ad acquisire e rilasciare il GPS.
- Soluzione: la **libreria Lifecycle** di Compose fornisce - con qualche piccola limitazione - accesso al lifecycle delle activity direttamente dall'interno dei composable

Libreria Lifecycle: installazione

- Nel blocco `[versions]` del file libs.versions.toml

```
lifecycleRuntimeComposeAndroid = "2.8.7"
```

- Nel blocco `[libraries]` del file libs.versions.toml

```
androidx-lifecycle-runtime-compose-android = {
    group = "androidx.lifecycle",
    name = "lifecycle-runtime-compose-android",
    version.ref = "lifecycleRuntimeComposeAndroid"
}
```

- Nel blocco `dependencies` del file build.gradle.kts (:app)

```
implementation(libs.androidx.lifecycle.runtime.compose.android)
```

Libreria Lifecycle: installazione

- Nel blocco `[versions]` del file libs.versions.toml

```
lifecycleRuntimeComposeAndroid = "2.8.7"
```

- Nel blocco [li… …oml

```
androidx-lifecyc
    group = "andr
    name = "life
    version.ref =
}
```

Queste istruzioni sono valide per installare la maggior parte delle librerie in un progetto Android

- Nel blocco `dependencies` del file build.gradle.kts (:app)

```
implementation(libs.androidx.lifecycle.runtime.compose.android)
```

Libreria Lifecycle: utilizzo

```
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Hello $name!",
        modifier = modifier
    )

    LifecycleEventEffect(Lifecycle.Event.ON_CREATE) {
        Log.i(TAG2, "onCreate")
    }
}
```

Libreria Lifecycle: limitazioni

- Non è possibile accedere a tutti gli stati del lifecycle, in particolare **`onRestart`** e **`onDestroy`** non sono disponibili.

Riferimenti

- Activity Lifecycle
  - https://developer.android.com/guide/components/activities/activity-lifecycle
  - https://developer.android.com/guide/components/activities/state-changes
  - https://developer.android.com/topic/libraries/architecture/compose
- Log https://developer.android.com/reference/android/util/Log
- Toast https://developer.android.com/guide/topics/ui/notifiers/toasts