

Android

where we start?

- File **manifest.xml** describes properties of the App
 - package used as id for the App in the device
`<manifest ... package="org.homenet.ehiahi.exampleapp" ...`
 - minimum Android version required by the App
`<uses-sdk android:minSdkVersion="8" />`
 - declaration of components and Intents to which they respond
`<activity android:name=".MainActivity">`
`<intent-filter>`
`<action android:name="android.intent.action.MAIN" /> ...`
 - needed permission for its functioning
`<uses-permission android:name="android.permission.INTERNET" />`

Android

what can you do if you don't need a GUI?

- **service** component doesn't need user interaction
- **but**, it is not a separate process or thread that runs in background
- It can start independent threads in the process of the App
- It is an object in the Dalvik process of the App
 - started with `Context.startService()`
 - stoppable with `Context.stopService()`
- Functionalities of the App can be exported to other Apps with **`Context.bindService()`**
 - external Apps must declare their intentions `<uses-permission>`

Android

what remains?

- Broadcast receiver
 - allows the reception of intents
 - sent with `Context.sendBroadcast()`
 - even with `Context.sendOrderedBroadcast()`
 - an extending class must register itself
 - dynamically with `Context.registerReceiver()`
 - in the manifest with `<receiver>`
- Content provider
 - they encapsulate data, given to the Apps
 - useful when sharing data with other Apps

Android

how are incorporated things outside the code?

- **resources**, that are identified by unambiguous IDs
 - automatically generated, and updated, with **aapt** command and put in a special class named **R**
 - **R.string.hello**, in Java code
 - **@string/hello**, in XML files
 - _ like in manifest.xml or layout files, or **strings.xml**
- In res/values we can find
 - **strings.xml**, that contains strings used in the App
 - a qualifier can change the language of the App (res/values-**fr**/ per le stringhe in Francese)
- In res/drawable-ldpi, -mdpi, -hdpi we can find images like icons
 - accessible with **@drawable/icon**, oppure **R.drawable.icon**
 - 36, 48, 72 pixel per lato
- In res/layout App's layouts
- In other directories (res/anim, res/color, res/menu), other things