Native Apps I

Fragments

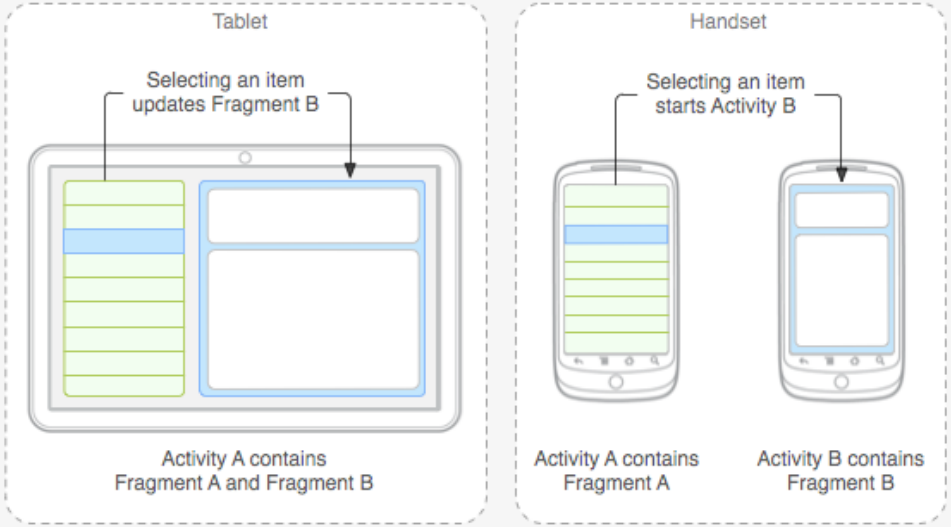
HO
GENT

Wat?

https://developer.android.com/guide/components/fragments

Deel van de UI

Meerdere Fragments kunnen gecombineerd worden in 1 activity



https://github.com/codepath/android_guides/wiki/Creating-and-Using-Fragments

HO GENT

Waarom?

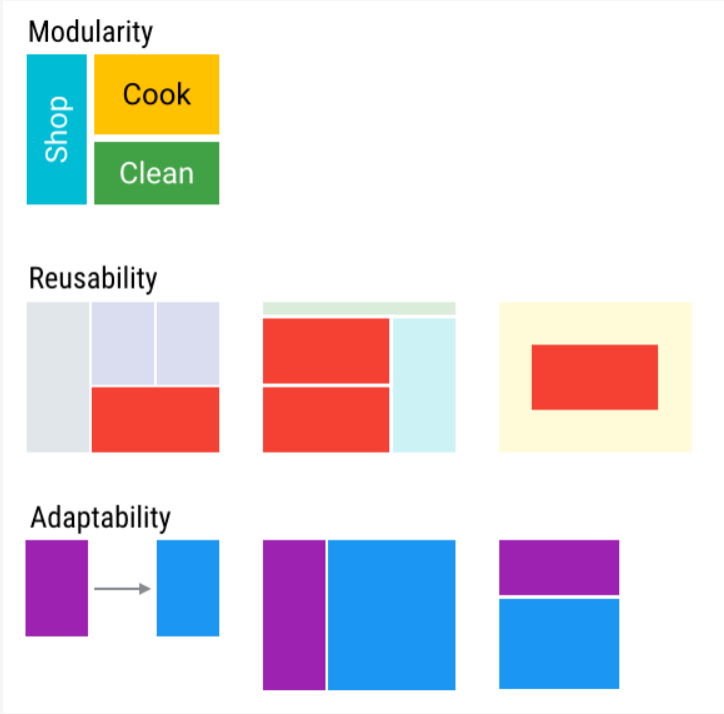
https://developer.android.com/guide/components/fragments

- **Modularity**

  Complexe activities kunnen opgesplitst worden in meerdere fragments
- **Reusability**

  Een fragment kan in verschillende schermen hebruikt worden
- **Adaptability**

  Fragments helpen bij het bouwen van responsive layouts



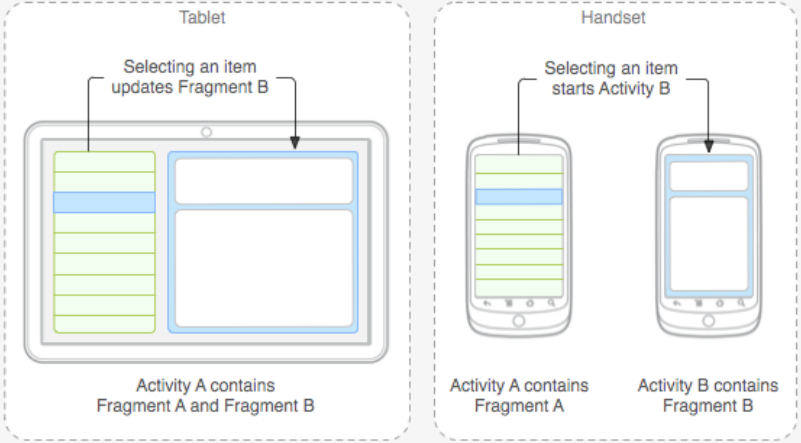
https://www.raywenderlich.com/361-android-fragments-tutorial-an-introduction-with-kotlin

HO GENT

Hoe?

https://developer.android.com/guide/components/fragments

- **Verschillende activity layout afhankelijk van schermbreedte (resource qualifier)**
- **Tabletlayout heeft lijst met items en een FrameLayout waarin Fragments geplaatst worden**
- **Handsetlayout heeft enkel de lijst en opent nieuwe Activity bij klikken**



https://www.studytonight.com/android/fragments-in-android

HO GENT

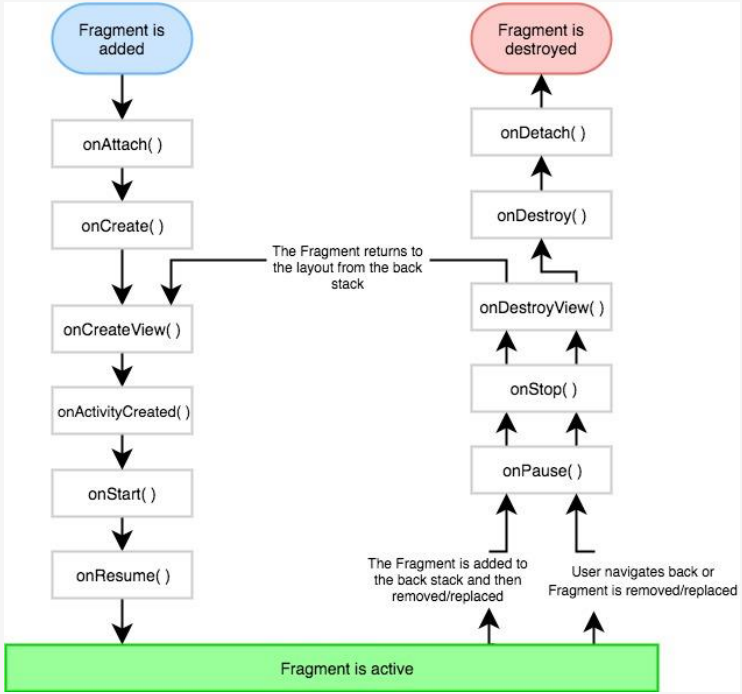
Fragment Lifecycle

Nog complexer dan Activities!

Fragment kan los van een Activity bestaan dus nog meer methodes

Mits gestructureerde aanpak niet allemaal nodig

- onCreate
- onCreateView
- onStart
- onResume
- onPause
- onStop



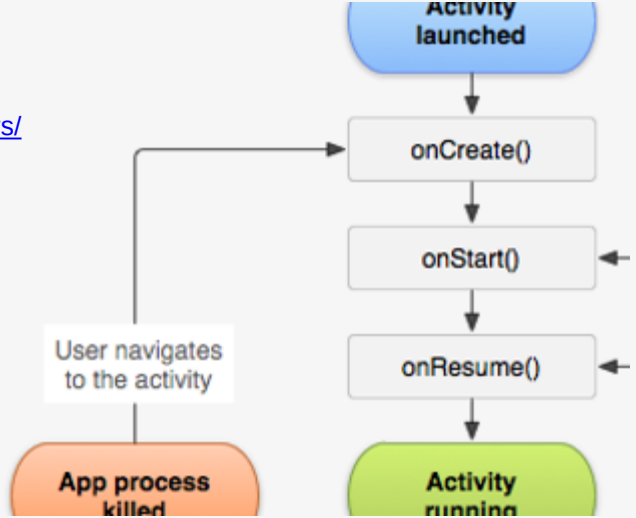
HO GENT

Fragment Lifecycle – onCreate

https://www.techyourchance.com/android-fragment-lifecycle-for-professional-developers/

Maakt Fragment aan (~constructor)

- **dependencies instellen**
- ***Saved state* herinitialiseren met *savedInstanceState***
- ***NIETS MET VIEWS***



HO GENT

Fragment Lifecycle – onCreateView

https://www.techyourchance.com/android-fragment-lifecycle-for-professional-developers/

Aparte methode om Viewhierarchie te initialiseren

Nodig omdat Fragment in meerdere hierarchien kan terecht komen doorheen zijn levensduur

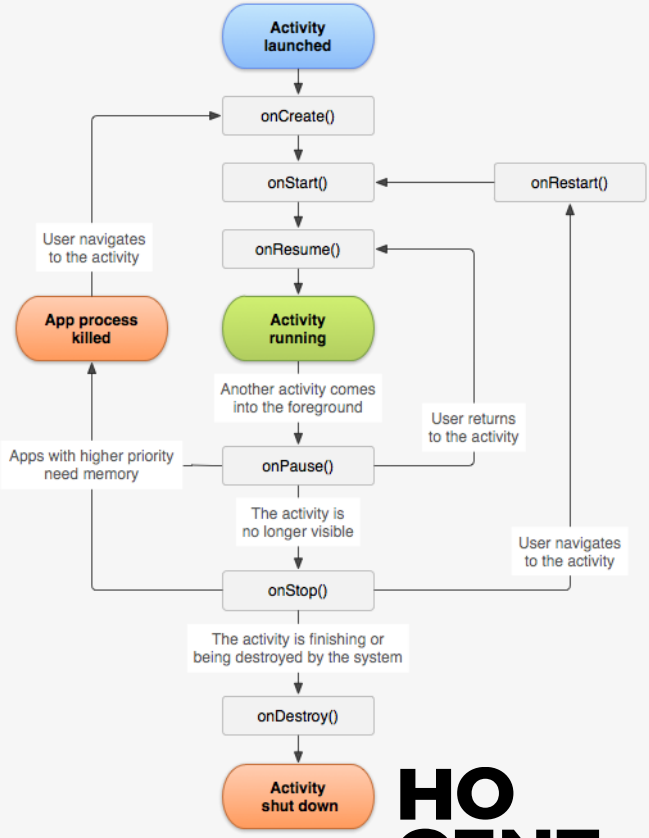
```
override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {
      val view = inflater.inflate(R.layout.fragment_crime_list, container, false)
      return view
}
```

HO GENT

Fragment Lifecycle – andere methodes

https://www.techyourchance.com/android-fragment-lifecycle-for-professional-developers/

Exact dezelfde aanpak zoals in Activities!



HO GENT

Native Apps I

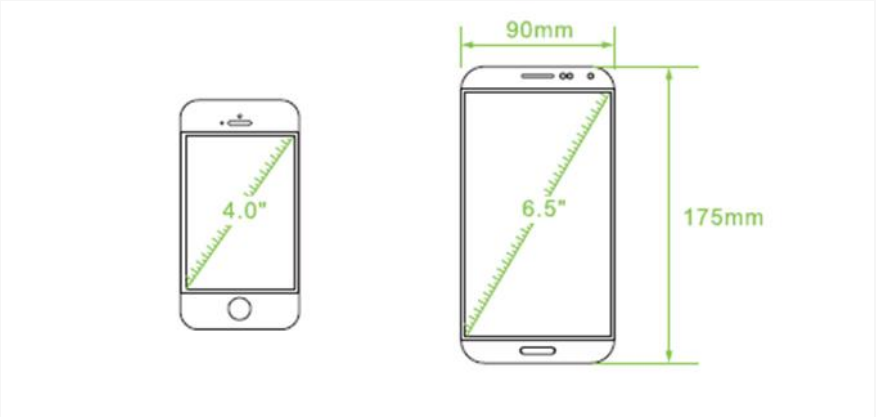
UI en Layouts

HO
GENT

Terminologie

Screen Size

Schermdiagonaal (inches)



http://www.vox-vr.com/vox-gear-plus-vr-virtual-reality-headset-vr-glasses-specs.html

HO GENT

Terminologie

Density

Pixels per oppervlakte

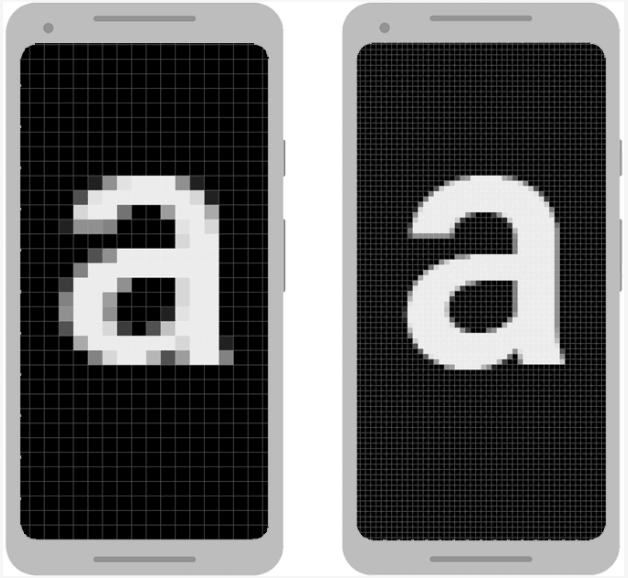
DPI (Dots Per square Inch)

Density Independent Pixel

1dp = 1 pixel op scherm van 160 dpi

Scalable Pixel

op zelfde gebaseerd als dpi, maar schaalt mee met preferred text size



https://developer.android.com/training/multiscreen/screendensities

HO GENT

Terminologie

Density

Pixels per oppervlakte

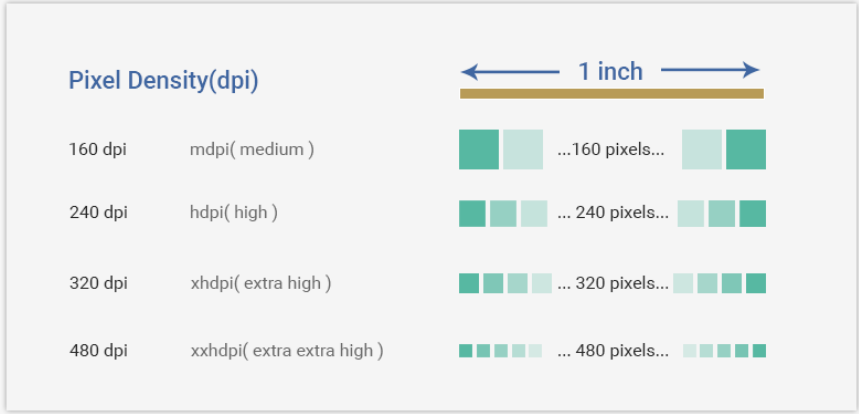
DPI (Dots Per square Inch)

Density Independent Pixel

1dp = 1 pixel op scherm van 160 dpi

Scalable Pixel

op zelfde gebaseerd als dpi, maar schaalt mee met preferred text size



http://findnerd.com/list/view/What-should-a-designer-must-know-before-starting-design-for-an-android-app-/2969/

HO GENT

Schermgroottes

Volgens screen size

- Small
- Normal
- Large
- Xlarge

Volgens dpi

- ldpi
- mdpi
- tvdpi
- hdpi
- xhdpi
- xxhdpi

HO
GENT

Best practices

Gebruik geen afmetingen in px, maar dp en sp

Gebruik verschillende layouts afhankelijk van schermgroottes

Gebruik afbeeldingen in verschillende resoluties voor verschillende schermresoluties

HO
GENT

Basiscomponenten

- View
- ViewGroup
- Activities
- Fragments

HO
GENT

Basiscomponenten

View

- Base class voor alle visuele componenten
- Een object dat iets op het scherm tekent en interactie toelaat

HO
GENT

View

Alles dat getekend kan worden op het scherm extend View

Properties van elke View

- **Focus**
- **Listeners**
- **Visibility**

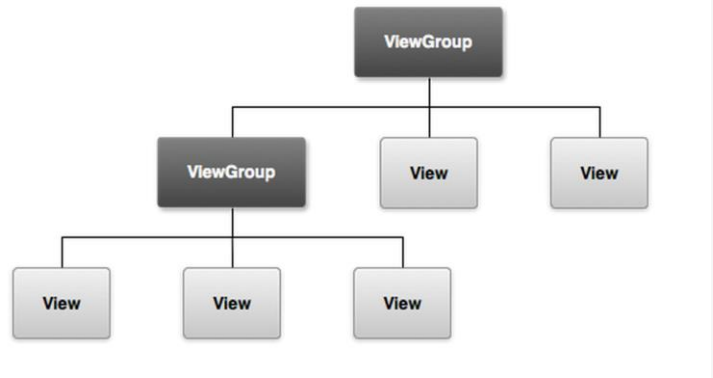
- **layout_width & layout_height**

HO
GENT

Basiscomponenten

ViewGroup

- Verzameling Views
- Beslist hoe die Views in UI geplaatst worden
  - Samen met LayoutManager



https://stackoverflow.com/questions/27352476/difference-between-view-and-viewgroup-in-android

HO GENT

Basiscomponenten

Activity

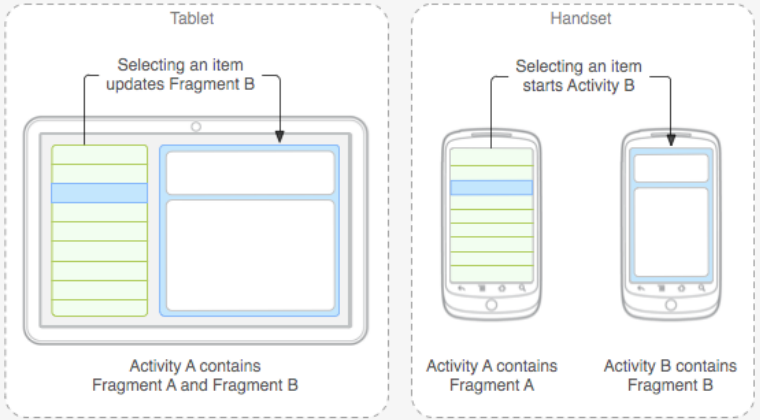
- 1 scherm van de applicatie
- Kan data uitwisselen met andere schermen

HO
GENT

Basiscomponenten

Fragment

- Encapsuleert 1 onderdeel van de UI
- Herbruikbaar
  - Bvb. bij verschillende schermgroottes



https://github.com/codepath/android_guides/wiki/Creating-and-Using-Fragments

HO GENT

layout_width & layout_height

https://developer.android.com/guide/topics/ui/declaring-layout

Mogelijke waarden

- MATCH_PARENT = vul zover als parent toelaat
- WRAP_CONTENT = net groot genoeg
- waarde in DP

HO GENT

Containers & LayoutManagers

https://developer.android.com/guide/topics/ui/declaring-layout

LayoutManagers beslissen hoe de Views in de layout geplaatst worden

- **LinearLayout**
- **TableLayout**
- **ConstraintLayout ←**
- **….**

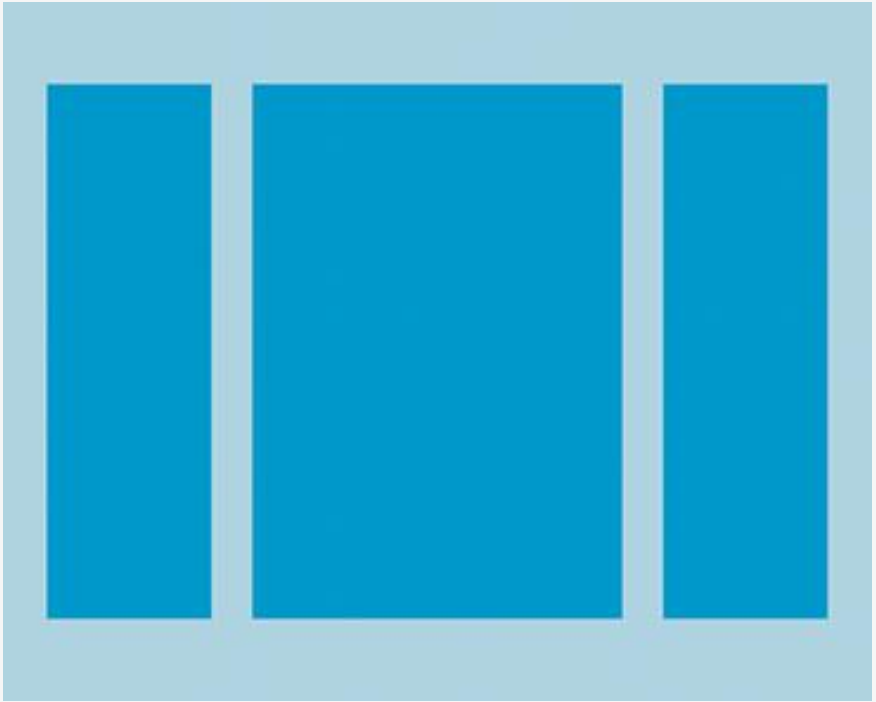
HO GENT

LinearLayout

https://developer.android.com/guide/topics/ui/layout/linear

ViewGroup waarin alle kinderen onder/naast elkaar komen te staan

- **afhankelijk van `android:orientation=vertical/horizontal`**



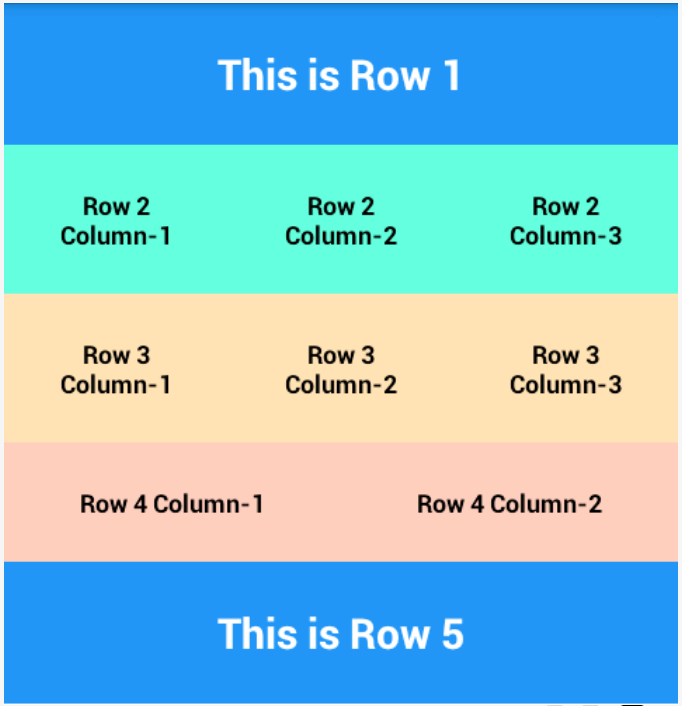
HO GENT

TableLayout

https://developer.android.com/guide/topics/ui/layout/linear

ViewGroup met tabelvorm

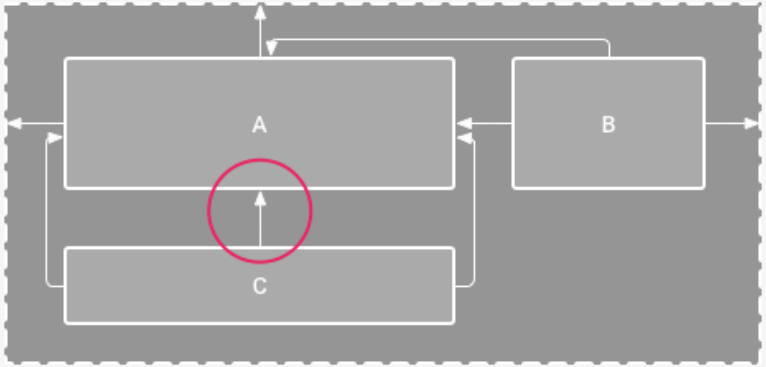
- **Meerdere `TableRows` met daarin de child views**
- **Gelijkaardig aan tabellen in HTML**



HO GENT

ConstraintLayout

https://developer.android.com/training/constraint-layout/



Eenvoudig complexe en vlakke Layouts maken

Views worden met Constraints aan elkaar gelinkt
minstens 1 horizontaal en 1 verticaal

Go-to layout als (aantal) elementen vastligt

HO
GENT

Screen &
UI performance

https://developer.android.com/training/constraint-layout/

Opletten hoe je

- **Views / Layouts opbouwt**
    - Measure/Layout/Draw
    - Mogelijks meerdere keren
- **Assets gebruikt**

- **Op het scherm tekent**

HO
GENT