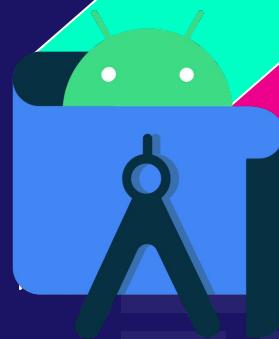


#8

Android Studio



# Android Studio

" Reusable components, programmatically"

# INNEHÅLLSFÖRTECKNING

01

Översikt

03

Fragments  
Programmatically

02

Fragments

04

Uppgifter  
&  
Övningar

# UPPDATERAD LAB #2

## BESKRIVNING





01

ÖVERSIKT

# What's Wrong with INTENT?

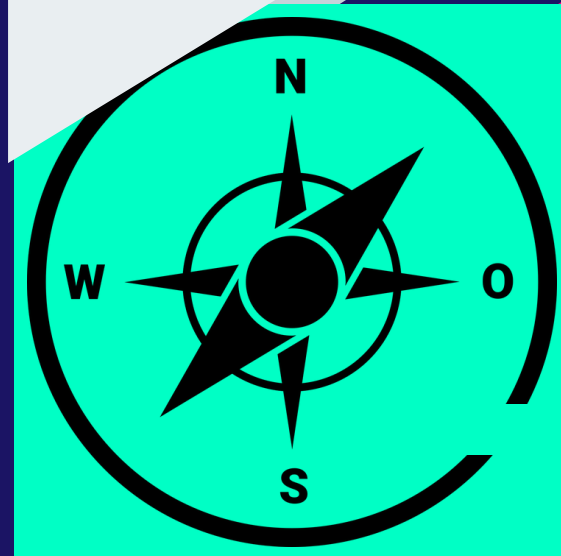


# Intent...

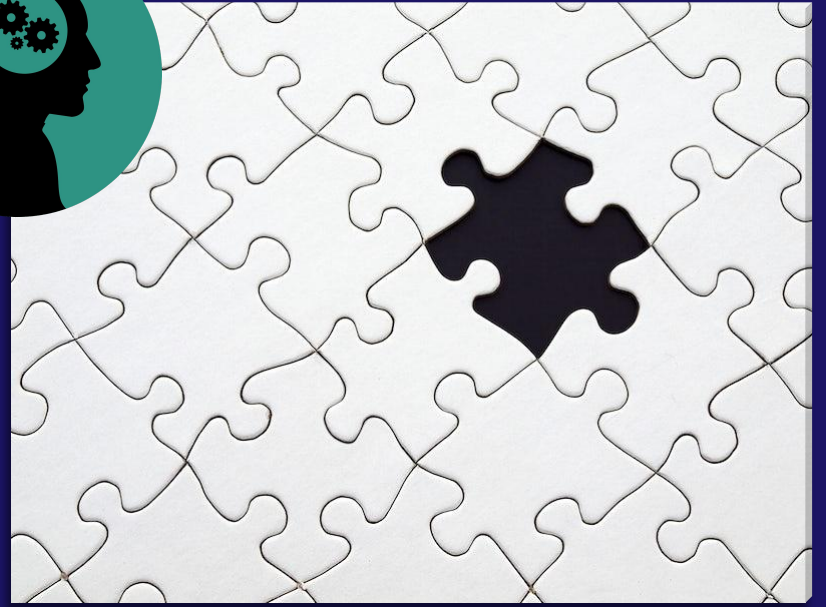
Varför är det fel med Intent?

- Inte fel, bättre alternativ för navigering!
- Fragments + komponenter
- Intuitiv design och navigering
- Grafisk design

*Innan vi går vidare till Navigation Graph, så tittar vi på fragment!*



*Frågor?*





02

Fragments





**“A fragment represents a  
reusable portion of your app's  
UI”**

- Android Developers docs

# Fragments!

Komponenter inom React kunde vi lägga uppe på en annan!

- Har sin egen livscykel
- Är UI som kan läggas uppe på en annan aktivitet
- De måste 'hosta's av en aktivitet eller ett annat fragment
- Hanterar sina egna input 'events'



# Get Started



# Get Started

*Dependency*

```
implementation "androidx.fragment:fragment-ktx:1.5.5"
```

build.gradle



# Get Started

*'Res' folder*

New	>	Kotlin Class/File
Add C++ to Module		Layout Resource File
		Sample Data Directory
Cut	⌘X	File
Copy	⌘C	Scratch File
Copy Path/Reference...		Directory
Paste	⌘V	Image Asset
		Vector Asset
Find Usages	⇧F7	Kotlin Script
Find in Files...	⇧⌘F	Kotlin Worksheet
Replace in Files...	⇧⌘R	CMakeLists.txt
Analyze	>	
Refactor	>	Activity
Bookmarks	>	Fragment
Show In Resource Manager	⇧⌘T	Folder
Reformat Code	⇧⌘L	Service
Optimize Imports	⇧⌘O	UiComponent
Delete...	⌘⌫	Automotive
		XML
Open In	>	Wear
Local History	>	AIDL
Repair IDE		Widget
Reload from Disk		Google
Compare With...	⌘D	Compose
		Other

Gallery...

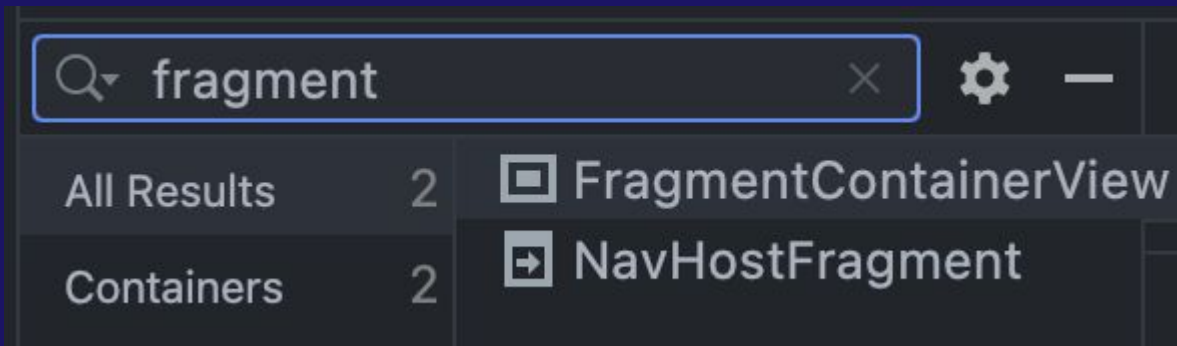
- Fragment (Blank)
- Fragment (List)
- Fragment (with ViewModel)
- Fullscreen Fragment
- Google AdMob Ads Fragment
- Google Maps Fragment
- Login Fragment
- Modal Bottom Sheet
- Scrolling Fragment
- Settings Fragment

Att skapa ett nytt fragment är super enkelt!  
Utför samma steg som med ny aktivitet, fast med  
ett fragment istället!



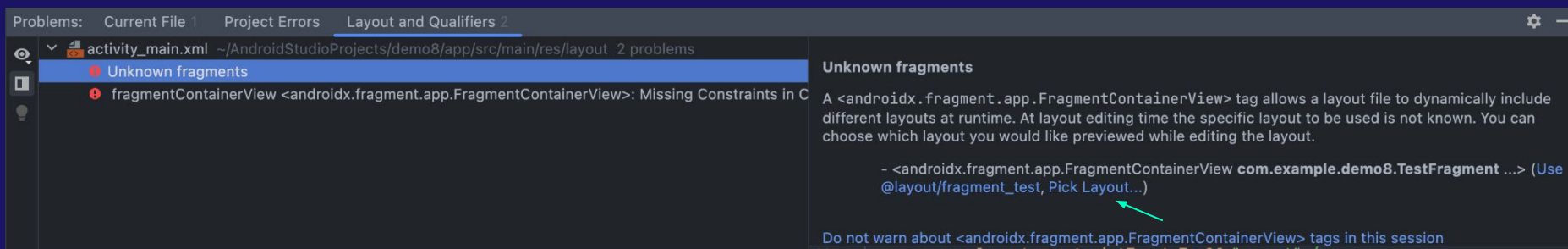
# Get Started

Gå till er Aktivitet och lägg till en:  
'FragmentManagerView'



En Container krävs för att visa upp ett fragment.

# Get Started



Problemet är Android Studio vet inte vilket fragment den ska lägga till inom 'Layout Editor' för att fragmentet lades till inom 'runtime'

Lösning

Tryck på 'Pick Layout'

## Choose a Layout




Module: demo8.ap

🔍 fra



### Layout

demo8.app.main (1)

 **fragment\_test**  
Layout | 1 version

appcompat (0)

Nothing to show

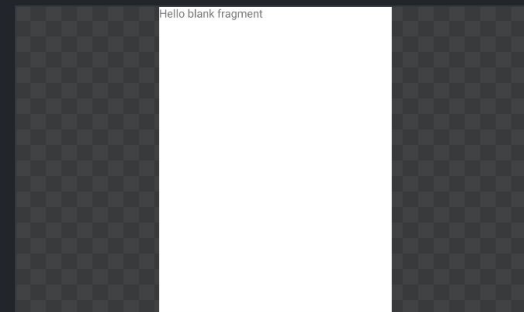
android (0)

Nothing to show

Theme attributes (0)

Nothing to show

### Preview



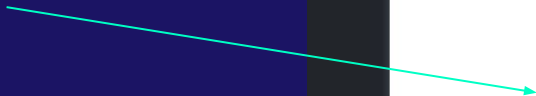
Name:	fragment_test
Reference:	@layout/fragment_test
Type:	FrameLayout
Configuration:	default
Value:	fragment_test.xml

Cancel

OK



Voila!



Hello blank fragment

Hello World!



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     xmlns:app="http://schemas.android.com/apk/res-auto"
7     tools:context=".TestFragment">
```

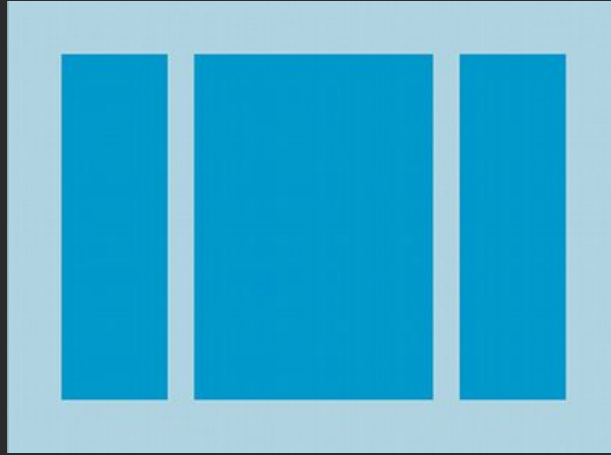
Ny Layout -> FrameLayout

# Layouts



# LinearLayout

[LinearLayout](#) is a view group that aligns all children in a single direction, vertically or horizontally. You can specify the layout direction with the [android:orientation](#) attribute.



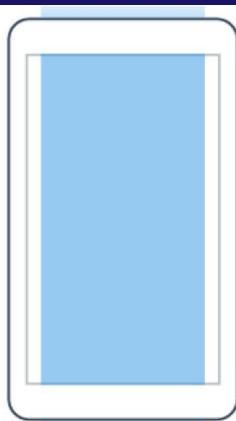
**Note:** For better performance and tooling support, you should instead [build your layout with ConstraintLayout](#).



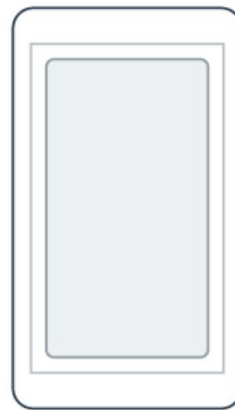
ContentPresenter



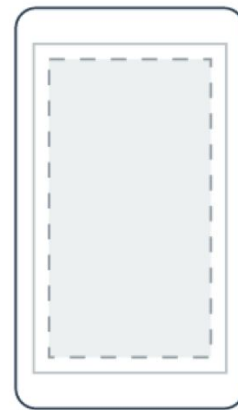
ContentView



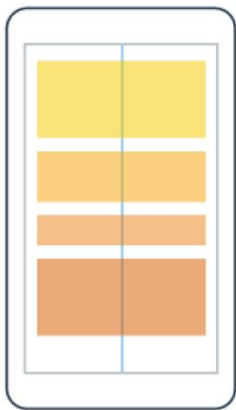
ScrollView



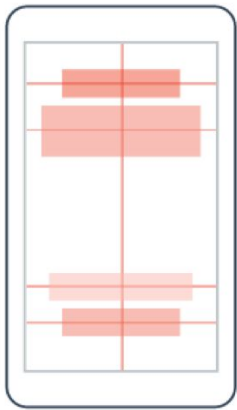
Frame



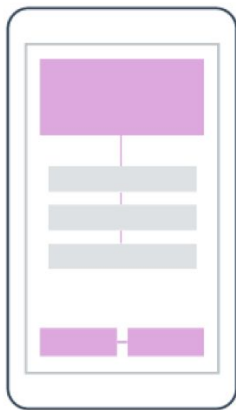
TemplatedView



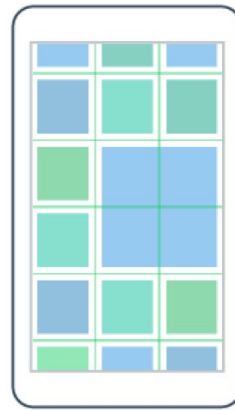
StackLayout



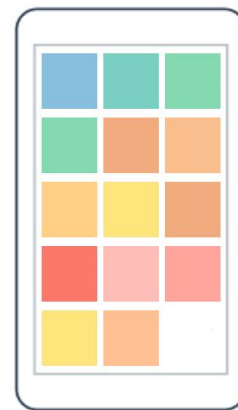
AbsoluteLayout



RelativeLayout



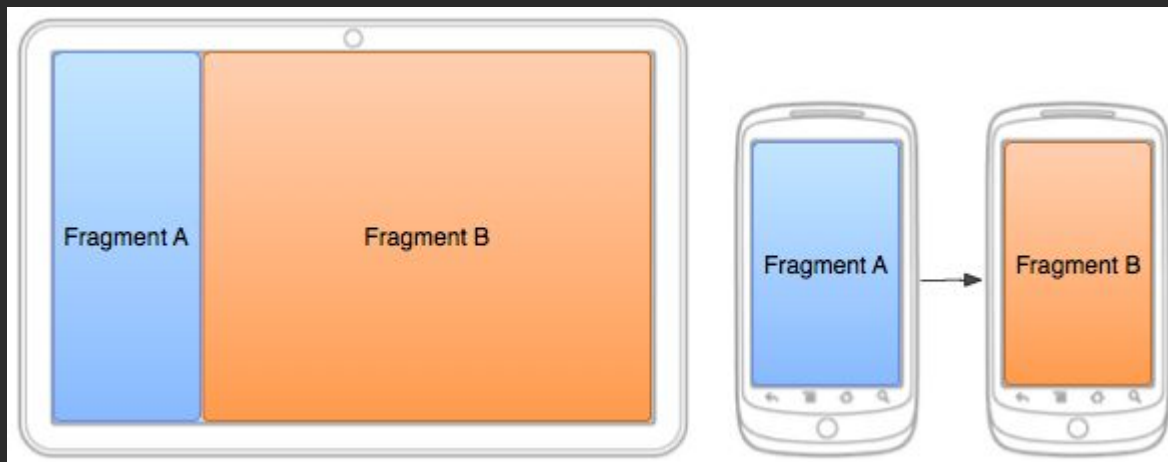
Grid



FlexLayout

# FrameLayout

*FrameLayout* is designed to block out an area on the screen to display a single item.



Child views are drawn in a *stack*, with the most recently added child on top. The size of the FrameLayout is the size of its largest child (plus padding), visible or not (if the FrameLayout's parent permits).

# What is a stack?

`bookList[0]`

`bookList[10]`



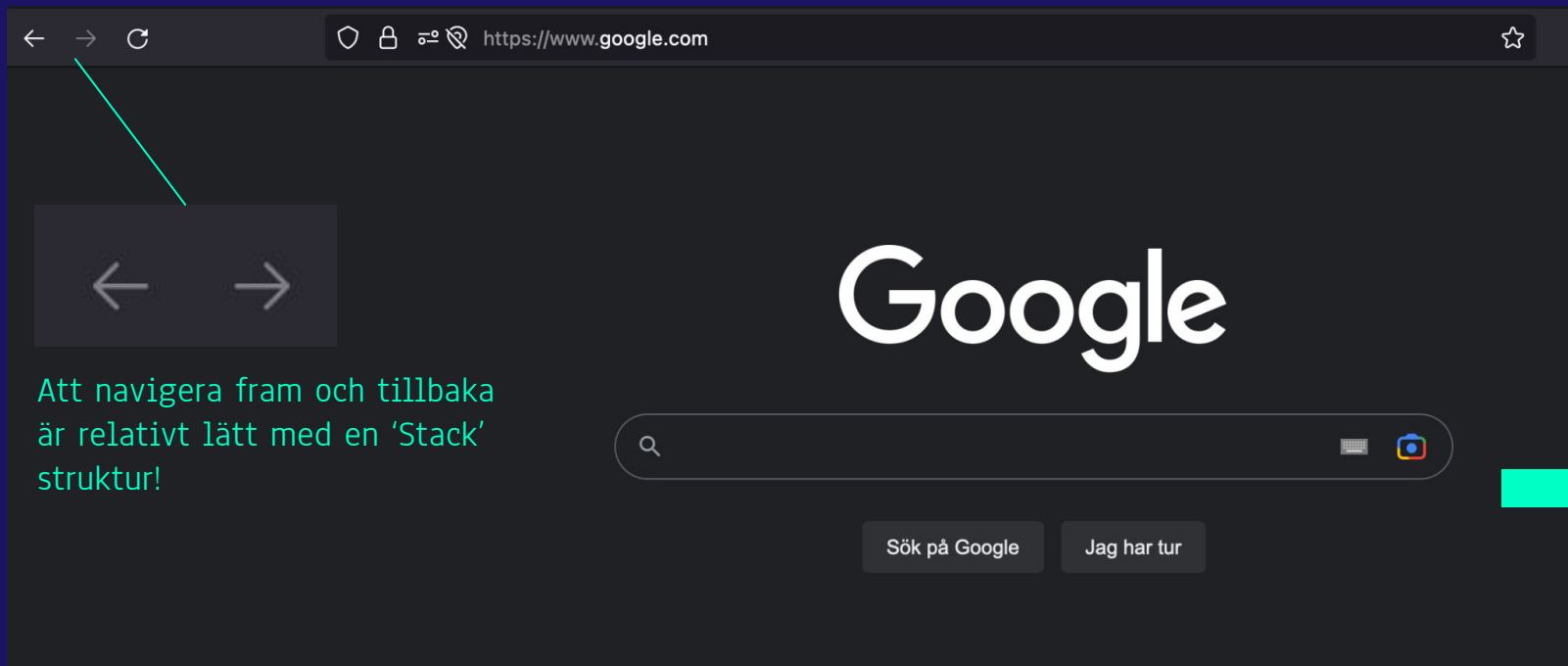
En Arrays datastruktur är linjär.

What is a stack?



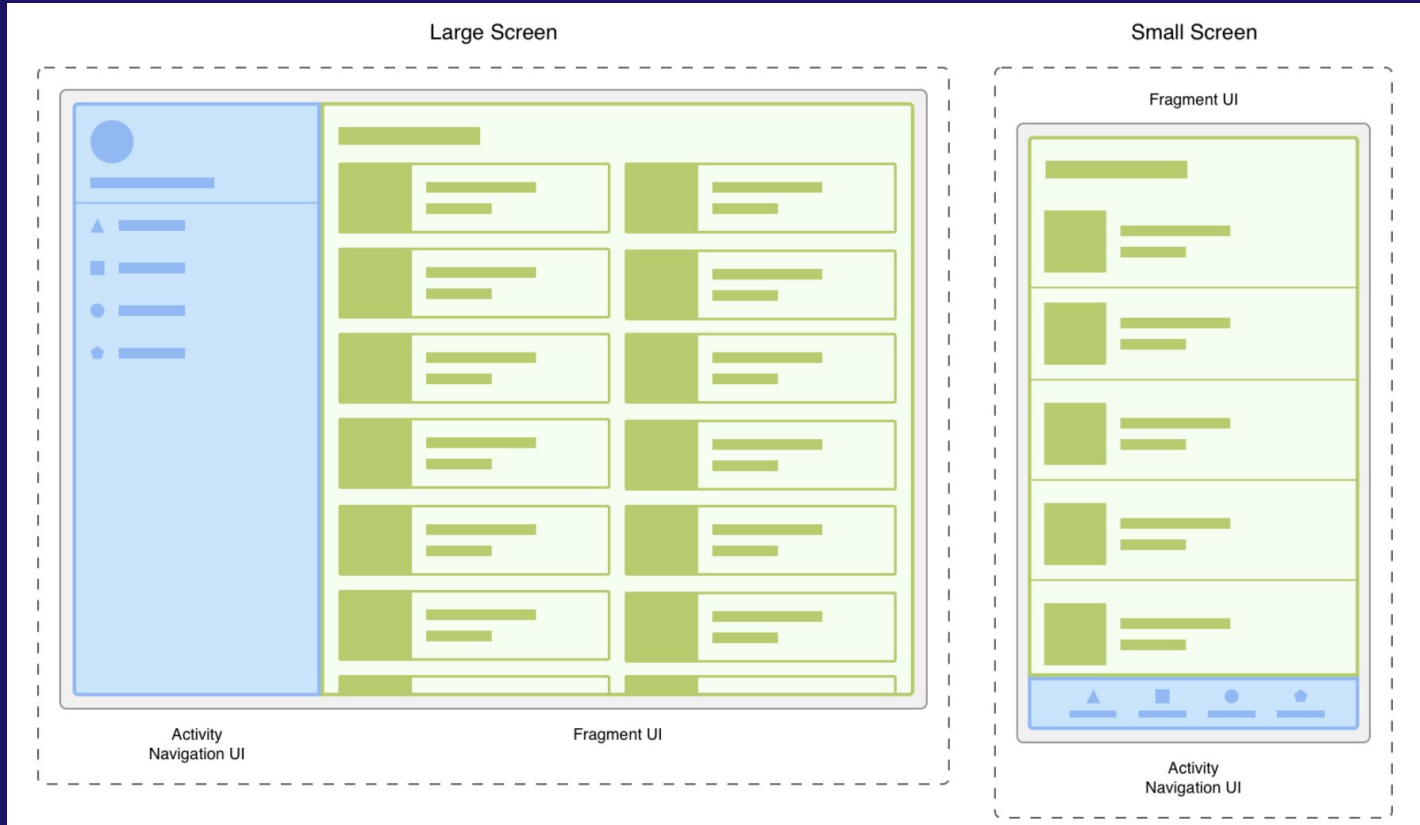


# What is a stack?



Att navigera fram och tillbaka  
är relativt lätt med en 'Stack'  
struktur!

# Fragments!





## Problem

Vad är ett 'FrameLayout'?

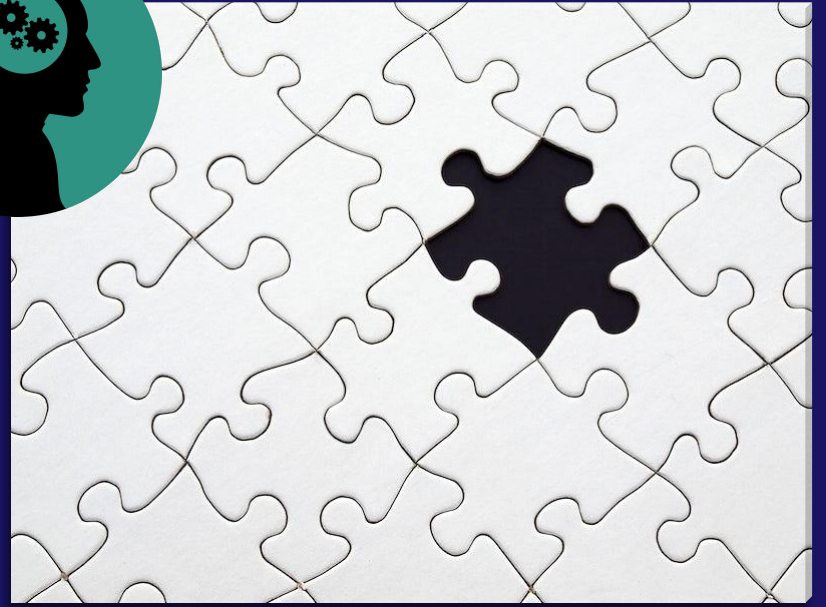


## Solution

En Layout som enbart är till för att visa upp en mindre 'vy' inom en annan!



*Frågor?*



# FrameLayout & LinearLayout DESIGNING TIME



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     xmlns:app="http://schemas.android.com/apk/res-auto"
7     tools:context=".TestFragment">
```

Ny Layout -> FrameLayout

FrameLayout is designed to block out an area on the screen to display a single item. Generally, FrameLayout should be used to hold a single child view, because it can be difficult to organize child views in a way that's scalable to different screen sizes without the children overlapping each other. You can, however, add multiple children to a FrameLayout and control their position within the FrameLayout by assigning gravity to each child, using the `android:layout_gravity` attribute.

Child views are drawn in a stack, with the most recently added child on top. The size of the FrameLayout is the size of its largest child (plus padding), visible or not (if the FrameLayout's parent permits). Views that are `View.GONE` are used for sizing only if `setConsiderGoneChildrenWhenMeasuring()` is set to true.

```
1      <?xml version="1.0" encoding="utf-8"?>
2      <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3                  xmlns:tools="http://schemas.android.com/tools"
4                  android:layout_width="match_parent"
5                  android:layout_height="match_parent"
6                  xmlns:app="http://schemas.android.com/apk/res-auto"
7                  tools:context=".TestFragment">
8
9          <androidx.appcompat.widget.LinearLayoutCompat
10              android:layout_width="match_parent"
11              android:layout_height="match_parent"
12              android:gravity="center"
13              android:background="@color/purple_200" >
14
```

Det går att kombinera Layouts!  
Här har vi `FrameLayout` + `LinearLayout`!



```
16      <TextView
17          android:layout_width="wrap_content"
18          android:layout_height="wrap_content"
19          android:text="@string/hello_blank_fragment" />
20
21      <com.google.android.material.floatingactionbutton.FloatingActionButton
22          android:id="@+id/floatingActionButton"
23          android:layout_width="wrap_content"
24          android:layout_height="wrap_content"
25          android:clickable="true"
26          android:src="@drawable/ic_launcher_foreground"
27          app:fabSize="mini"
28          />
29  </androidx.appcompat.widget.LinearLayoutCompat>
30 </FrameLayout>
```

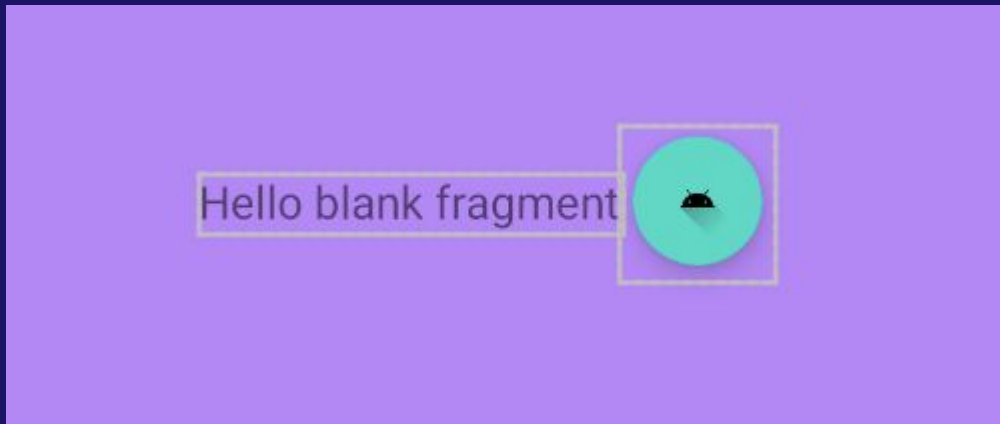
Längre ner har vi en enkel TextVy + FBA



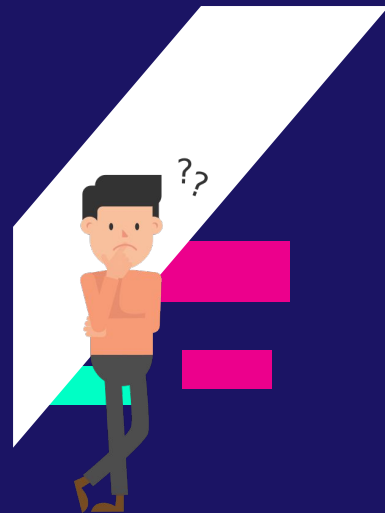


# Första Fragment

Resultat!



*Med 'Gravity Center', har vi lyckats centrera allt helt!*



*Förbättra design?*

Mål:



Hello blank fragment



*Jag vill ändra färgen på knappen samt storleken:*

- FAB är mycket mindre
- Färgen har ändrats
- Ny XML fil

*Se nästa slides för kod*

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/floatingActionButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:clickable="true"

    app:fabCustomSize="20dp"
    android:src="@drawable/ic_cross"
    app:maxImageSize="10dp"
    app:fabSize="mini"

    android:backgroundTint="@color/red_200"
    android:background="@color/red_200"
    app:backgroundTint="@color/red_200"
/>
```

*Det kommer inte fungera att bara Copy/paste:a denna kod, se nästa slide för mer information!*

- ▼ res
  - ▼ drawable
    - ic\_cross.xml
    - ic\_launcher\_background.xml
    - ic\_launcher\_foreground.xml (v24)
  - ▼ layout
    - activity\_main.xml
    - fragment\_test.xml
  - > mipmap
  - ▼ values
    - colors.xml
    - strings.xml
    - > themes (2)
    - > xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="red_200">#FF6161</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
</resources>
```

▼ drawable

ic\_cross.xml

ic\_launcher\_background.xml

ic\_launcher\_foreground.xml (v24)

*Detta ritar upp ett kryss, experimentera gärna med koden och titta vad de olika attribut gör!*

```
<?xml version="1.0" encoding="utf-8"?>

<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <rotate
            android:fromDegrees="-45"
            android:pivotX="50%"
            android:pivotY="50%"
            android:toDegrees="-45"
        >
            <shape android:shape="line" >
                <stroke android:width="1dp" android:color="@android:color/white" />
            </shape>
        </rotate>
    </item>
    <item>
        <rotate
            android:fromDegrees="45"
            android:pivotX="50%"
            android:pivotY="50%"
            android:toDegrees="45"
        >
            <shape android:shape="line" >
                <stroke android:width="1dp" android:color="@android:color/white" />
            </shape>
        </rotate>
    </item>
</layer-list>
```

Hello blank fragment



Hello World!

Den lila bakgrunden är såklart onödig här och väldigt ful, men det vi är ute efter är att se resultat.

Nu ser vi att hela bakgrunden ÄR HELA FRAGMENTET!



## Problem

Hur skapar vi ett Fragment?

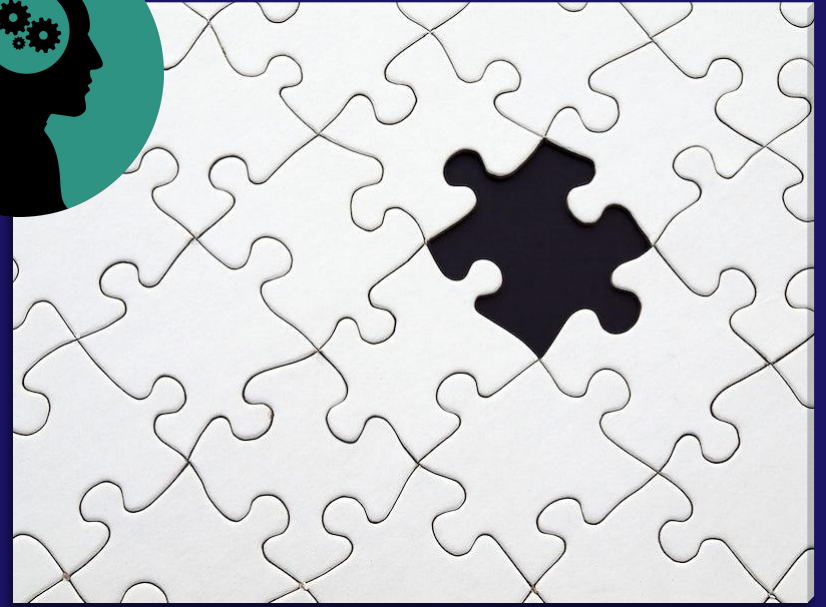


## Solution

Likt Aktiviteter, så kan vi högerklicka och trycka på 'ny: tomt fragment' FragmentContainerView krävs även då 'Fragments' hosted by views'



*Frågor?*







03

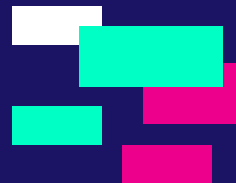
# Fragments Programmatically

# Adding / Removing Fragments Programmatically



# Viktigt!

*För att kunna manipulera fragment så använder vi oss av en getter som heter: `supportFragmentManager`*



# Container

## FragmentManager

Vi skriver om vår Container och tar bort referensen till vår klass.

Vi vill inte ha referenser då vi ska programmatiskt skapa en!

```
<androidx.fragment.app.FragmentContainerView
    android:id="@+id/fragmentContainerView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:layout_marginBottom="24dp"
```

```
app:layout_constraintBottom_toTopOf="@+id/textView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```



FILL

## FragmentManager

Om ni vill att fragmentet skall täcka HELA vy'n (kommer täcka över andra komponenter)

Gör bredd och höjd till 'match\_parent'

*Inom kommande exempel utför vi EJ denna kod.*

```
<androidx.fragment.app.FragmentManager
    android:id="@+id/FragmentManager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="24dp"
    android:layout_marginBottom="24dp"
```

```
app:layout_constraintBottom_toTopOf="@+id/textView"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

# Button

Skapa en knapp!


```
<Button
    android:id="@+id/btn_createFragment"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Spawn Fragment"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintTop_toBottomOf="@+id/textView"
    app:layout_constraintVertical_bias="0.269" />
```

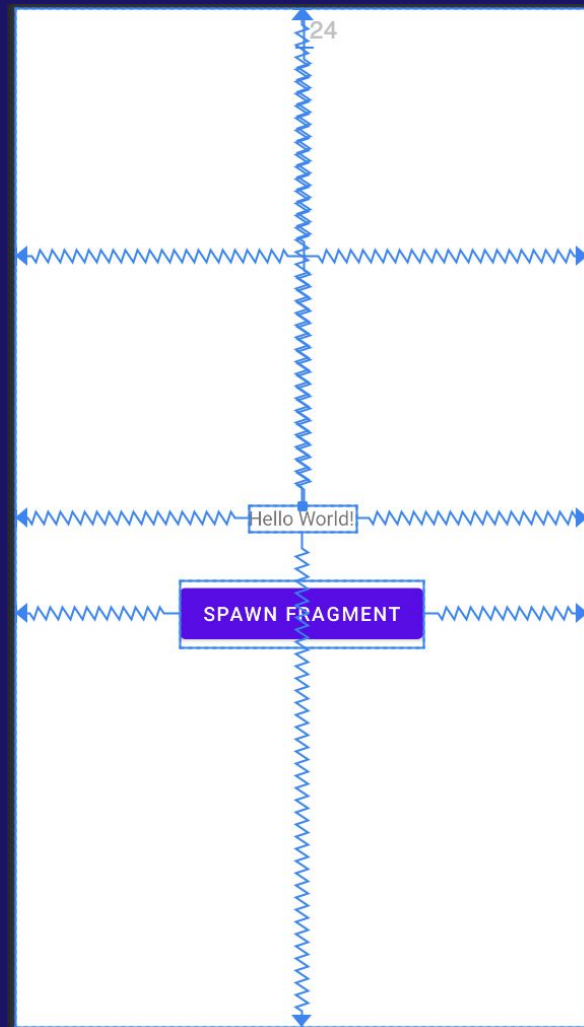


← Tidigare vy

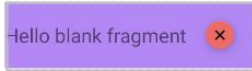
Hello blank fragment 

Hello World!

Nuvarande Vy →



← Tidigare Vy

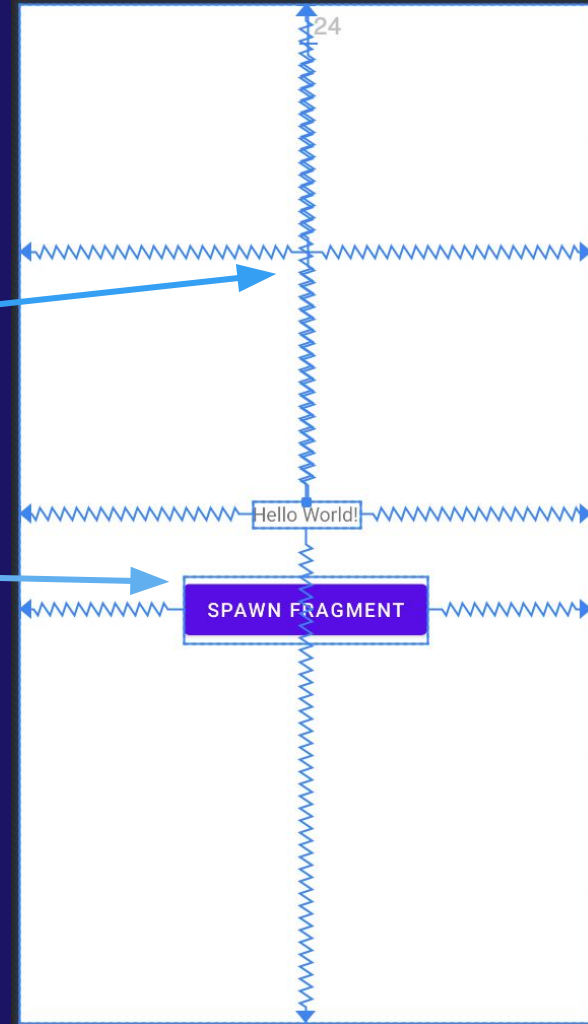


Hello World!

*Detta är  
FragmentManager!*

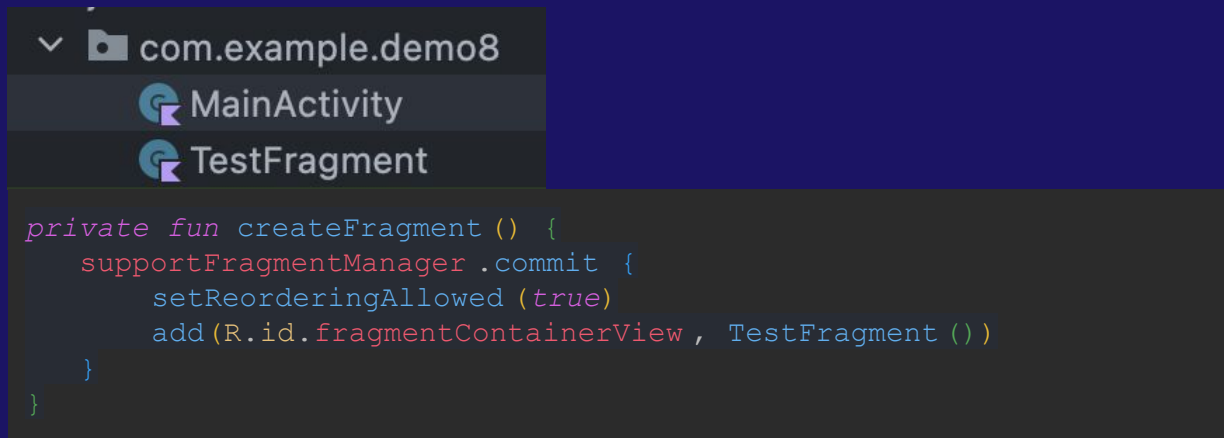
*OnClick skall skapa ett  
nytt fragment!*

Nuvarande Vy →





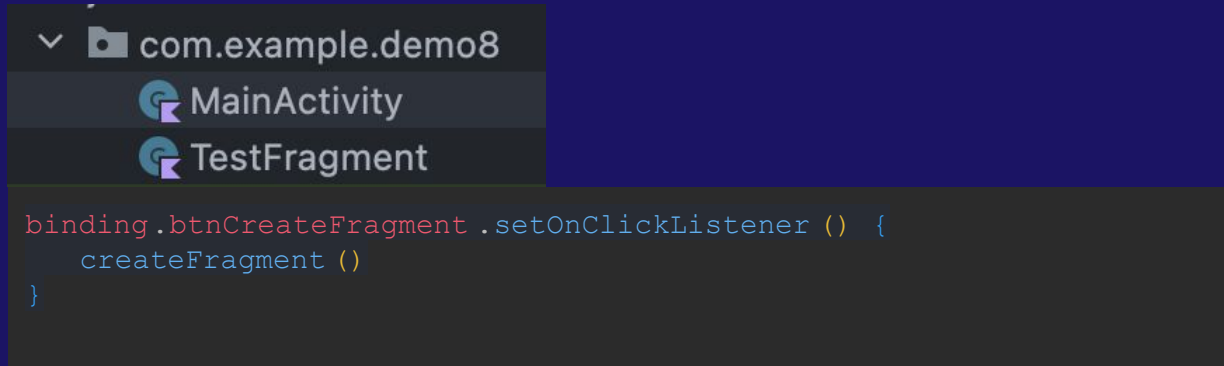
# Create



Inom denna funktion så skapar vi ett nytt fragment.

★ **Note:** You should **always** use `setReorderingAllowed(true)` when performing a `FragmentManager`. For more information on reordered transactions, see [Fragment transactions](#). [source](#)

# Create



Kalla sedan på metoden via en onClick()

# PROBLEM

Om vi nu trycker på denna knapp flertal gånger så kommer den skapa fragment uppepå fragment.

Det kan vara bra att förstöra fragmentet:

- Om vi trycker på knappen 'CREATE'
- Om vi trycker på kryss X

Hello blank fragment



# Fragment

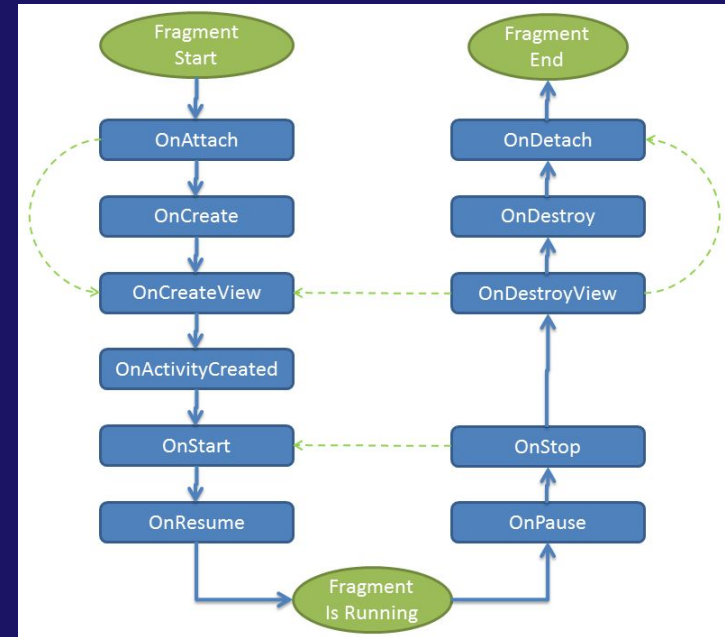
com.example.demo8

MainActivity

TestFragment

```
override fun onCreateView(  
    inflater: LayoutInflater, container: ViewGroup?,  
    savedInstanceState: Bundle?  
): View? {
```

Navigera till TestFragment -> onCreateView livscykel  
*onCreateView, det är här vi kommer definiera onClicks!*



# OnClick DESTROY

```
override fun onCreateView(  
    inflater: LayoutInflater, container: ViewGroup?,  
    savedInstanceState: Bundle?  
): View? {  
  
    // Get Fragment Reference  
    val view: View = inflater.inflate(R.layout.fragment_test, container, false)  
  
    // On click  
    view.findViewById<FloatingActionButton>(R.id.floatingActionButton).setOnClickListener()  
{  
        println("SOMETHING HAPPENED")  
        parentFragmentManager.beginTransaction().remove(this).commit() // Remove Fragment  
    }  
  
    // ViewBinding Finalization  
    return view  
}
```

# PROBLEM

Om vi nu trycker på denna knapp flertal gånger så kommer den skapa fragment uppepå fragment.

Det kan vara bra att förstöra fragmentet:

- Om vi trycker på knappen 'CREATE'
- ~~Om vi trycker på kryss ✕~~

Hello blank fragment



# Fragment == Exist?

```
private fun createFragment () {  
  
    supportFragmentManager .commit {  
  
        // DEBUGGING  
        println("Does: myFragment, not exist?" )  
        println(supportFragmentManager .findFragmentByTag ("myFragment") == null)  
  
        if (supportFragmentManager .findFragmentByTag ("myFragment") == null ) {  
            setReorderingAllowed (true)  
            add(R.id.fragmentContainerView , TestFragment () , "myFragment")  
        } else {  
            supportFragmentManager .findFragmentByTag ("myFragment")?.let {  
                supportFragmentManager .beginTransaction ().remove (it)  
            }  
        }  
    }  
}
```

Om fragment är tom: Skapa

Annars: Ta bort



## Problem

Hur skapar vi ett Fragment programmatiskt?



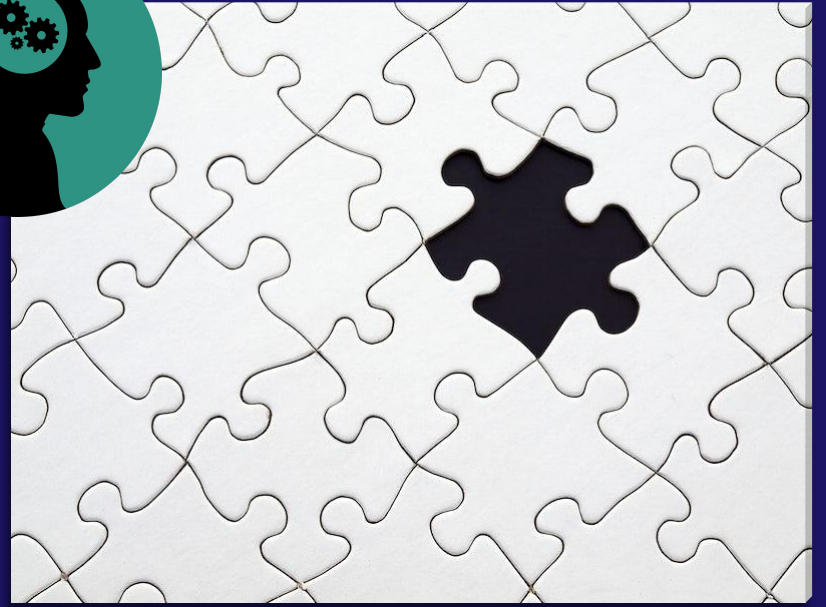
## Solution

`supportFragmentManager` låter oss manipulera fragment programmatiskt!





*Frågor?*





04

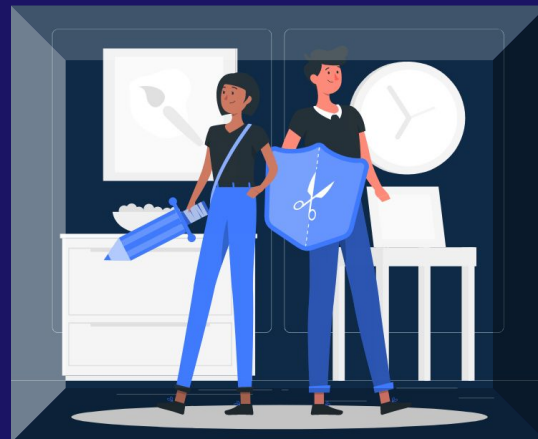
Uppgifter  
&  
Eget Arbete

## Välkommen till första uppgiften!

Uppgifterna är till för att testa dina färdigheter och kunskaper för att både öva och repetera på det vi har arbetat med under föreläsningarna.

Dessa är **INTE** obligatoriska.  
Men är starkt rekommenderat att arbeta med.

## Uppgifter



# MINNS DU?

```
// Vad är och vad används  
'supportFragmentManager' till?
```

```
// Vad är ett Fragment?  
Varför / när används dessa?
```

```
// Hur kan vi manipulera fragment  
programmatiskt?
```

```
När hade vi velat göra detta?
```

```

1      // -Uppgift #1- //
2
3      /* INSTRUCTIONS
4
5          Skapa ett nytt projekt!
6
7          Döp projektet till: Lektion_8_uppgifter
8
9          Högerklicka på 'layouts' inom 'src'
9          Välj nu att skapa ett nytt blank 'Fragment'
10
11          Navigera till Main Activity, drag och släpp
12          sedan in en 'FragmentManager'
13
14      */
15
16      // HINT & Examples
17      hint(" FragmentManager, agerar som en punkt
18      där dina fragment kommer skapas.
19
20      Glöm inte att fixa error:et där du manuellt måste
21      säga till vilket fragment som skall visas upp.
22      Har du glömt: gå till Slide #15 ")
23

```

## Uppgift #1

Kom igång enkelt med uppgift #1

```

1          // -Uppgift #2- //
2
3      /* INSTRUCTIONS
4
5          Designa nu ditt Fragment!
6
7      Tips:
8          + Gravity - positionerar komponenter
9          + LinearLayoutCompat - arbeta inom Grid
10         + Välj valfria komponenter
11
12         När du är klar, navigera till MainActivity
13         och skapa en onClickListener för ditt
14         fragment.
15
16         Denna onClick kan skriva ut en enkel 'SOUT'
17     */
18
19     // HINT & Examples
20     hint(" Det går att kombinera layouts, det är inte
21     ovanligt att man gör detta när man designar.
22
23     Läs mer om LinearLayout: android.devs ")

```

## Uppgift #2

*Fragments är läskiga i början, mycket kod och väldigt finurlig design.*

*Låt inte detta skrämma dig, när du bemästrar dessa så har du uppnått en otrolig milstolpe!*

*You can do it!*

```
1      // -Uppgift #3- //
2      The Tough Nut
3
4      /* INSTRUCTIONS
5
6          Lägg till ett fragment programmatiskt.
7
8          När du lägger till ett fragment via
9          en onclick då visas en Snackbar upp med en
10         'Undo'.
11
12         Om användaren väljer att trycka på 'undo',
13         tas detta fragment bort.
14
15     */
16
17     // HINT & Examples
18     hint(" Ta bort / Lägg till fragments: Slide #30")
19
20
21
22
23
```

## Uppgift #3

# THANKS!

Do you have any questions?  
kristoffer.johansson@sti.se

[sti.learning.nu/](https://sti.learning.nu/)

CREDITS: This presentation template was created by  
Slidesgo, including icons by Flaticon, and  
infographics & images by Freepik.

*You can also contact me VIA Teams (quicker response)*  
*Du kan också kontakta mig VIA Teams (Snabbare svar)*