

#3

Android Studio

</>



Android Studio

" Let's get started! "

INNEHÅLLSFÖRTECKNING

01

Översikt

03

Sending Data

02

Navigate
activities &
Intent

04

Uppgifter
&
Övningar

01

ÖVERSIKT

Navigering, skicka data, schema mm..



Navigering

Applikationer är oftast inte SPA
(singlePageApplication)
Hur kan vi göra för att navigera?

- Navigering
- Skicka data
- Ta emot data
- Ändra på text inom komponenter såsom textView



Frågor?



Datatypes alternatives



Datatypes

```
val button: Button = findViewById(R.id.myButton)  
val button2 = findViewById<Button>(R.id.mainActivity)
```

Det finns två sätt vi kan definiera datatyper på ^

Display ImageView Properly



ImageView

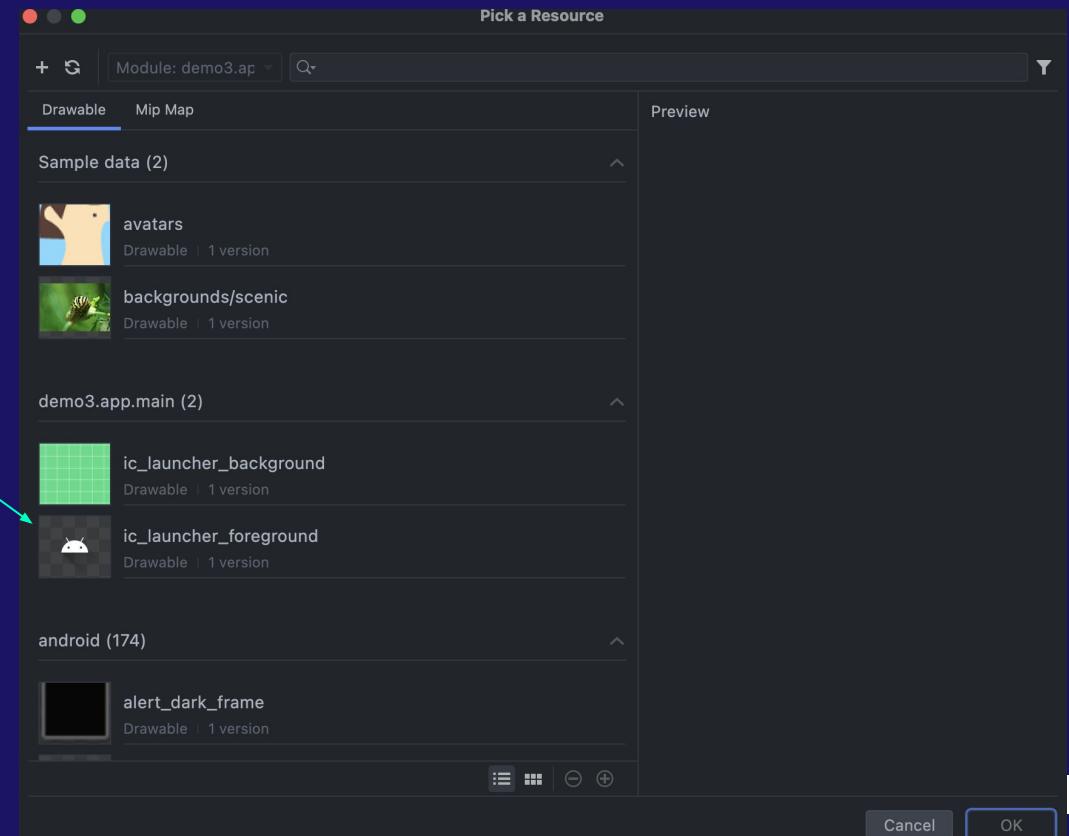
Om ni försöker skriva ut denna
bild så kommer det ej att gå!



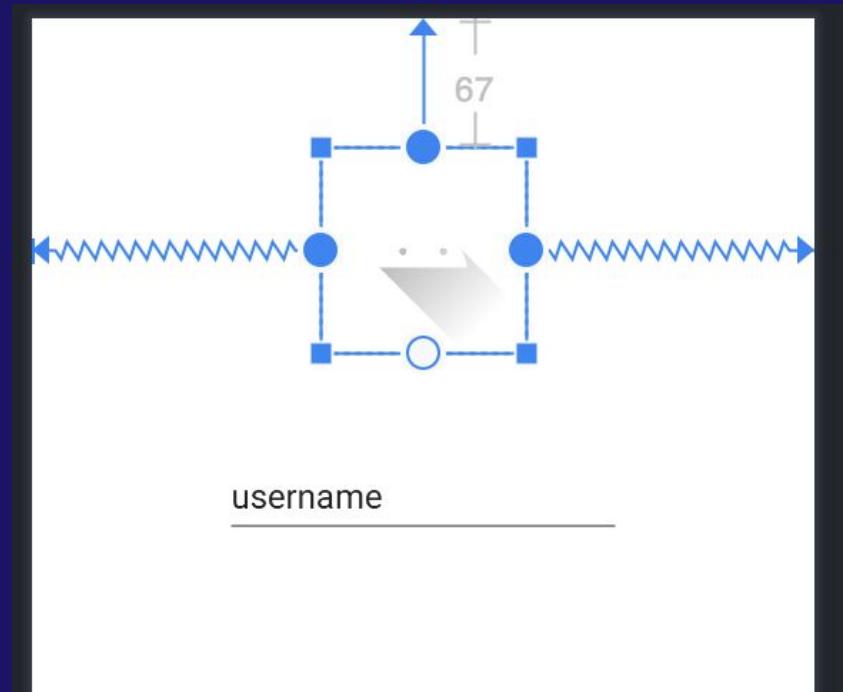
username

ImageView

Välj istället den här!



ImageView



Formning av kurs Resultat -poll-



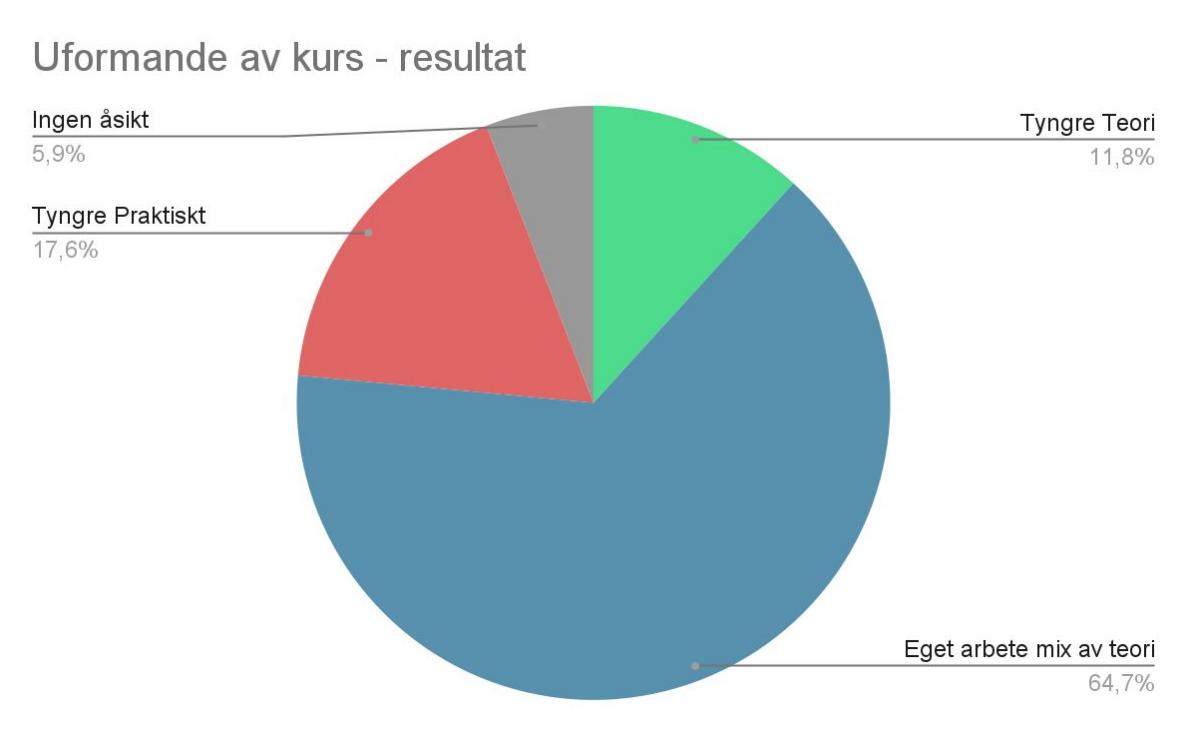
Kurs röstning

Tyngre teoretiska moment - 2

Eget arbete mix av undervisning - 11

Tyngre praktiska moment - 3

Ingen åsikt - 1



Kurs röstning

Vad innebär detta för er?

Vi kommer köra som vanligt och kommer sikta på eget arbete som vanligt
på eftermiddagar.

Jag kommer fortsätta ge ut uppgifter i slutet på varje lektion

Efter 14:00 kör vi handledning på distans

Upplägget ser ut som vanligt!

Slutligen så tittar vi på schemat!

Vecka	Innehåll	Veckodag	Datum	Längd	Sal
V 6	Kotlin - Intro	Torsdag	9/2	9-16	Sal 3
V 7	Android Studio - Intro	Tisdag	14/2	9-16	Sal 2
	Navigation & Data	Torsdag	16/2	9-16	Sal 6
	Självstudier - Lab 1 intro	Fredag	17/2	9-16	Distans
V 8	Switch Buttons	Måndag	20/2	9-16	Sal 2
	Classes & Objects	Onsdag	22/2	9-16	Sal 4
	Självstudier	Fredag	24/2	9-16	Distans
V 9	Google Play	Måndag	27/2	9-16	Sal 4
	Design Patterns & ListView	Tisdag	28/2	9-16	Sal 4
	Självstudier - Lab 2 intro	Torsdag	2/3	9-16	Distans
V 10	RecyclerView	Måndag	6/3	9-16	Sal 5
	Navigation Controller	Onsdag	8/3	9-16	Sal 4
V 11	SQLite	Tisdag	14/3	9-16	Sal 4
	API	Onsdag	15/3	9-16	Sal 5
	Självstudier	Fredag	17/3	9-16	Distans
V 12	Styles & Themes	Måndag	20/3	9-16	Sal 4
	Notifications	Tisdag	21/3	9-16	Sal 4
	Tentamen	Torsdag	23/3	9-16	Sal 4

Schema

Vi kommer sikta på självstudier på de sista dagarna i veckan för eget arbete.

OBSERVERA: Schemat som ni ser här kan komma att ändras och är inte på något sätt slutgiltigt!

Varför?

"Nya koncept, teknologier förändras, alternativa lösningar."

02

Navigate Activities

Activity

Mer om 'activity', vi gör ju inte mycket här...

- Navigering
- Skapar aktivitet
- ID från andra aktiviteter?



Instantiate Object

```
curl< curl_easy_setopt(conn, CURLOPT_URL, url);
curl< curl_easy_setopt(conn, CURLOPT_FOLLOWLOCATION,
curl< curl_easy_setopt(conn, CURLOPT_WRITEFUNCTION,
curl< curl_easy_setopt(conn, CURLOPT_WRITEFUNCTION,
curl< curl_easy_setopt(conn, CURLOPT_WRITEFUNCTION,
```

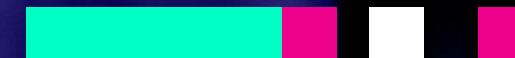
Instantiate

När vi skapar nya instanser, så behövs inte ett nyckelord för 'new'. Detta kan dock göra det lite förvirrande och svårt att se när vi instansierar något..

```
val cat = Cat()          // Kotlin  
private Cat cat = new Cat() // JAVA
```

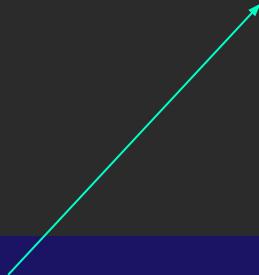
Vi ser här också att vi behöver inte specificera 'datatypen', då den är 'inferred', vilket antyder att den automatiskt känner och tilldelar rätt datatyp utan att behöva skriva extra kod.

ID naming convention



ID naming convention

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    val button: Button = findViewById(R.id.myButton)  
  
    button.setOnClickListener {  
        println("Hello World!")  
    }  
}
```



Detta var en god start men det blir snabbt rörigt om vi har med flertal komponenter!

ID naming convention



Button	-	btn_submit
TextView	-	tv_username
EditText	-	et_username
ImageView	-	iv_avatar

Det finns ingen officiell 'guide', men här följer vi iaf en best practice som liknar XML's standarder.

Vissa skriver hela styck t.ex.

txtView_username

Eller

textView_username

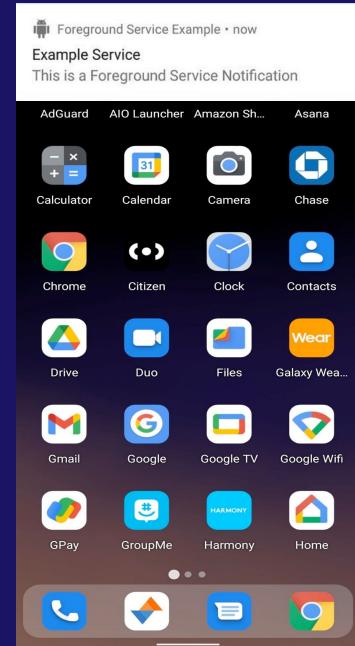
Hitta det som fungerar för dig!

Intent #1



Intent #1

Start An Activity



An [Intent](#) is a messaging object you can use to request an action from another app component. Although intents facilitate communication between components in several ways, there are three fundamental use cases:

[source: android docs](#)

Intent #1

Eftersom att det är en klass så måste vi instansiera den.

Vi vet även att ett Intent tar in parametrar beroende på 'vad' som skall göras.

Låt oss utforska detta med tiden!

```
val exampleIntent = Intent()
```





Problem

Vad är ett Intent?

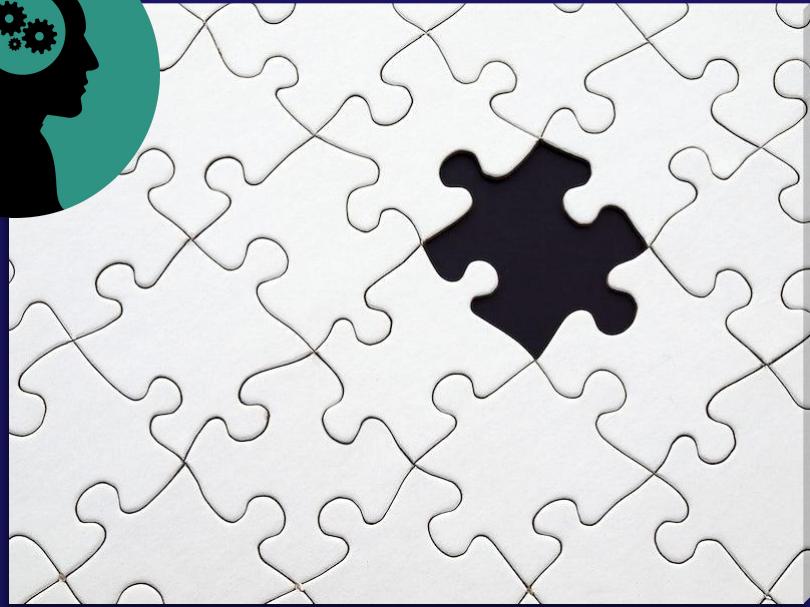


Solution

Ett intent har många ansvarsområden, vi kommer idag titta på hur man specifikt arbetar med just vidare skickande av data.



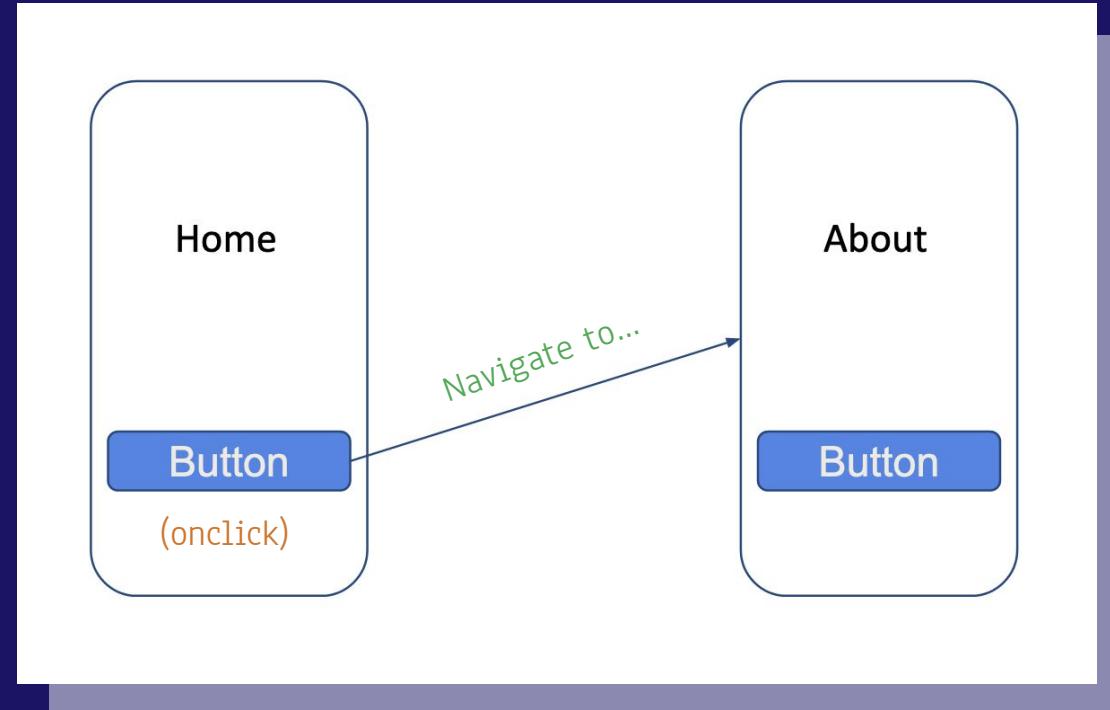
Frågor?



Navigate to Activity



Navigate



Navigate

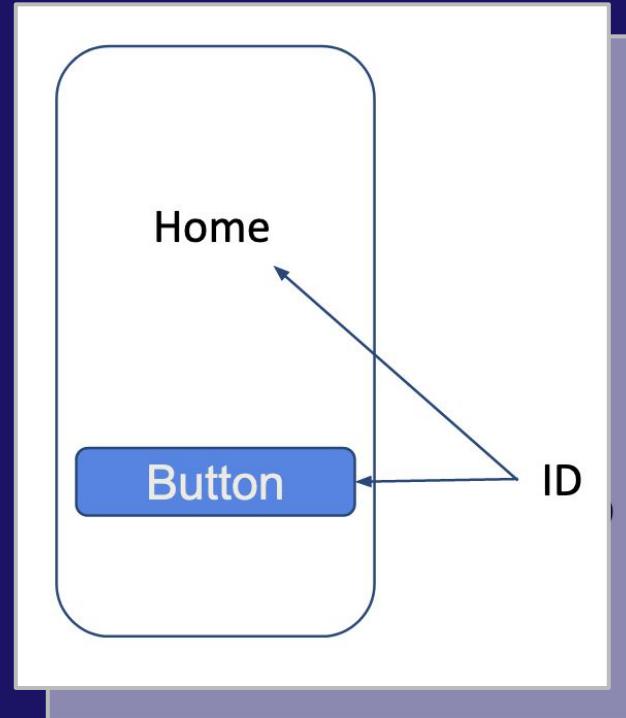
I Android Studio måste vi:

- #1 - Skapa **ID** till Komponent
- #2 - Skapa en **onClickListener**
- #3 - Navigera

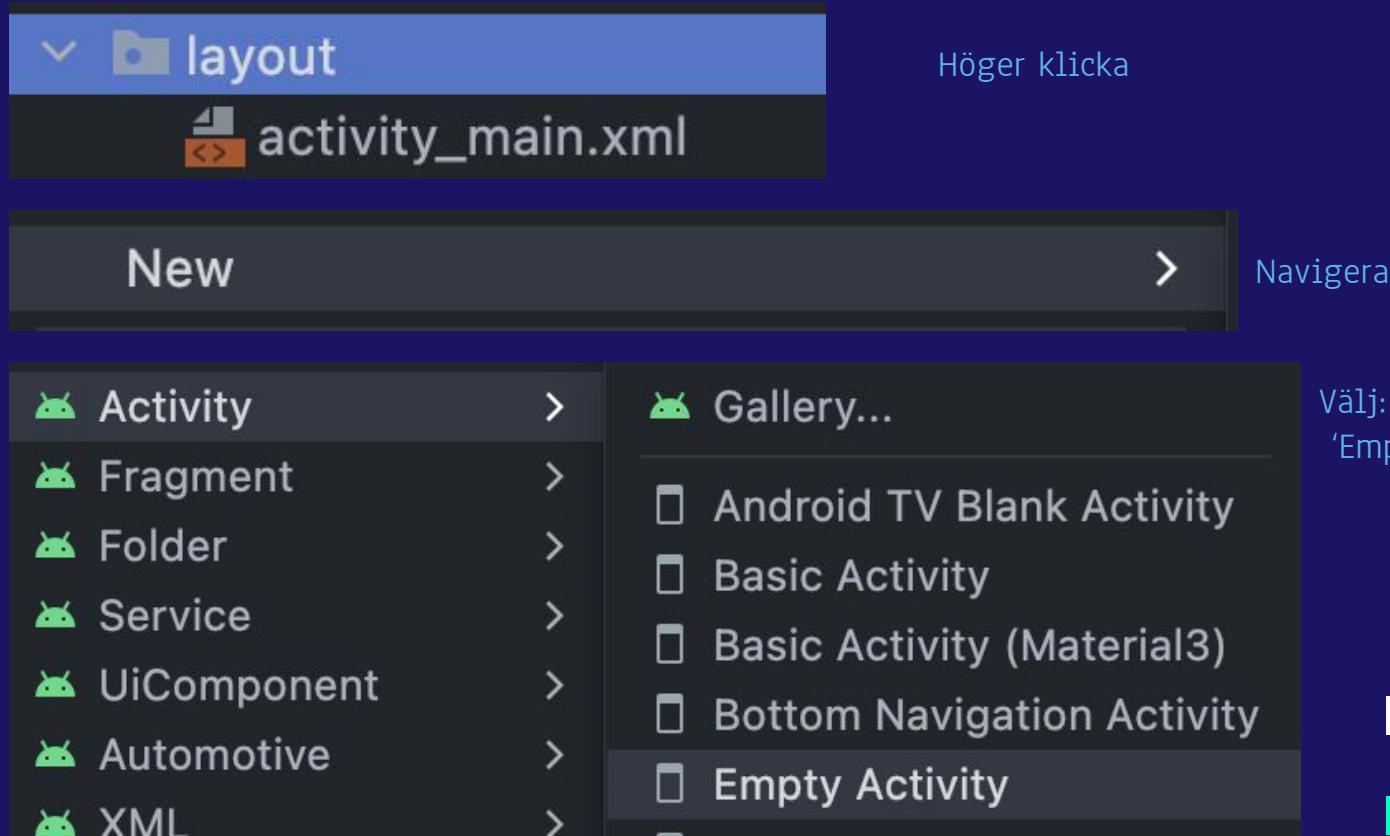
Varje komponent har ett **ID**

Vi kan checka om 'något' händer
med en **lyssnare**.

Slutligen **navigera** med 'Intent'



Create Activity



Empty Activity

Creates a new empty activity

Activity Name

AboutActivity

Generate a Layout File

Layout Name

activity_about

Launcher Activity

Package name

com.example.demo3

Source Language

Kotlin

'BestPractice' är att ha med ordet 'Activity' i slutet

Resterande är helt orört.

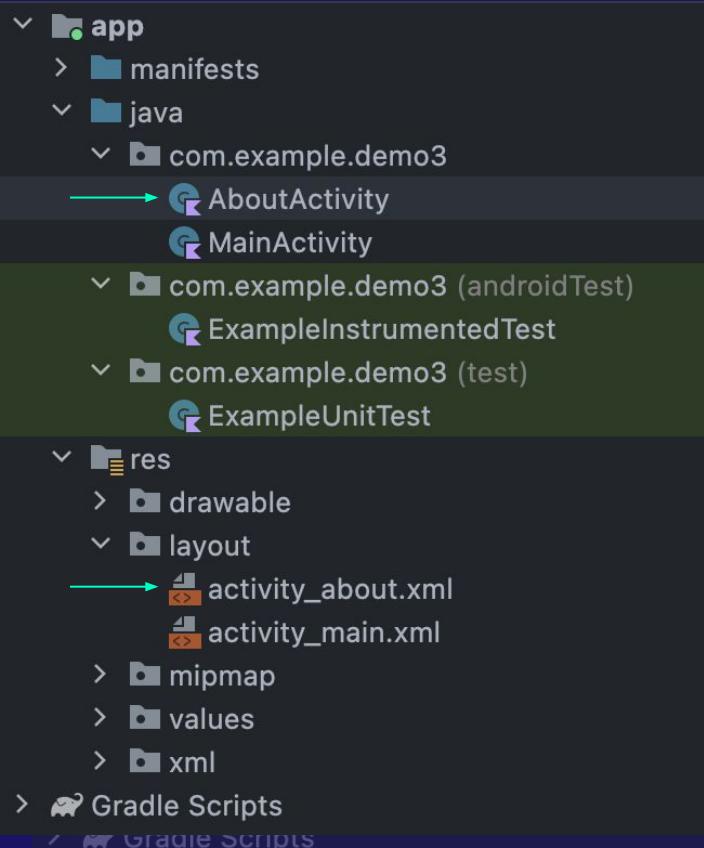
Cancel

Previous

Next

Finish

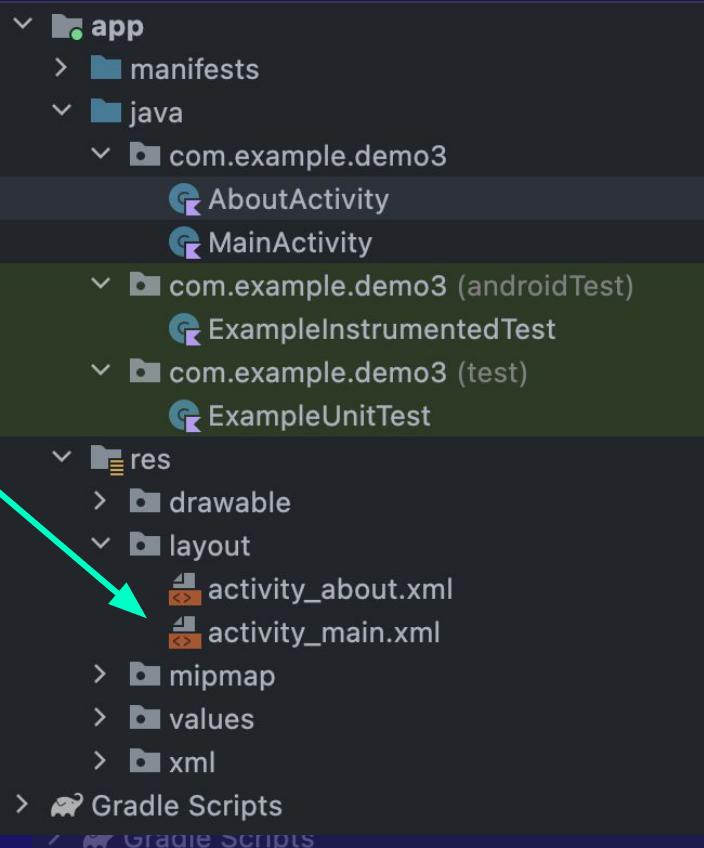
Structure



Resultat: Två extra filer!

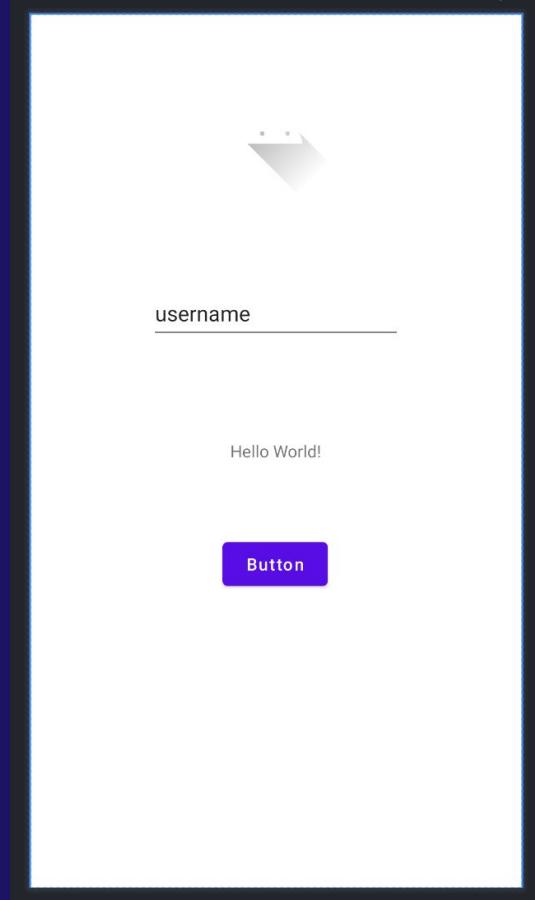
Navigate

Gå hit!



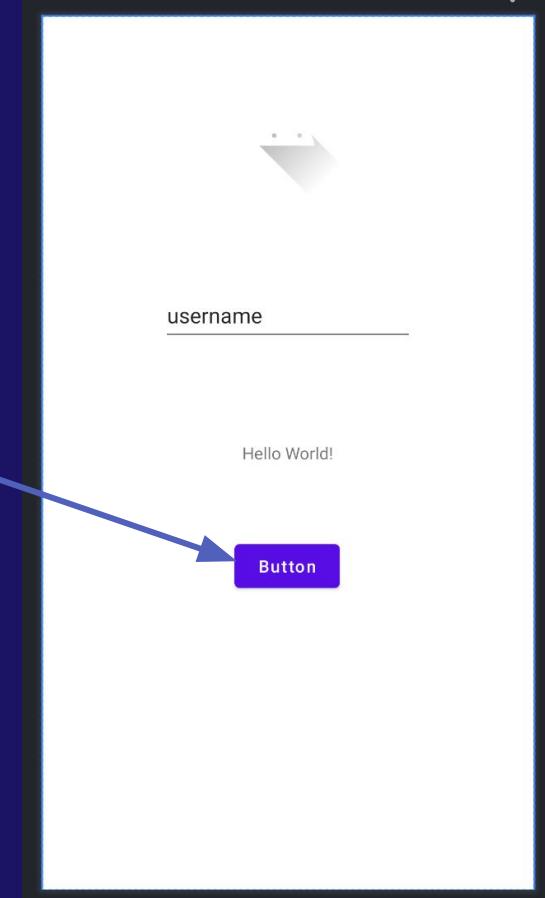
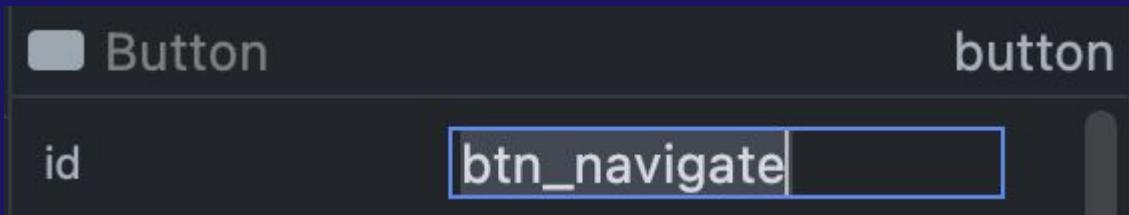
Resultat: Två extra filer!

Our Screen



Our Screen

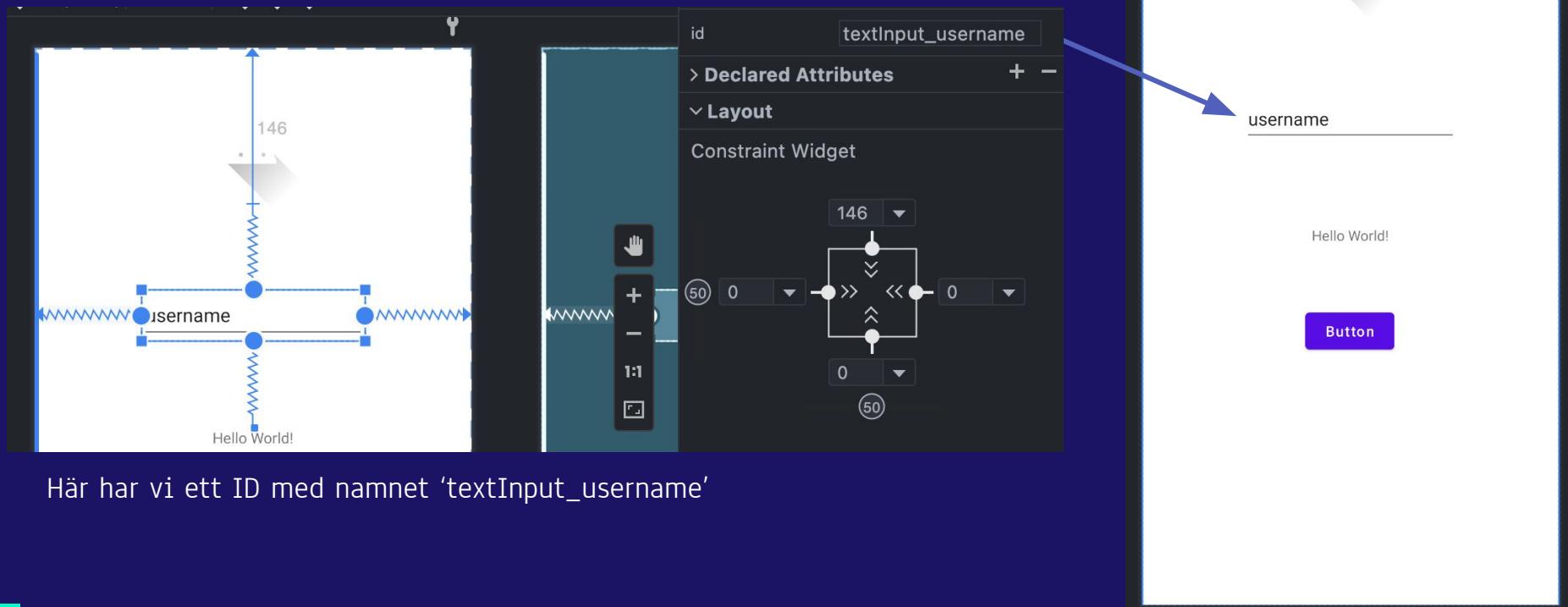
Lägg till en knapp som vi kan navigera till



Namnkonvention jag följer:

komponentNamn_funktionalitet

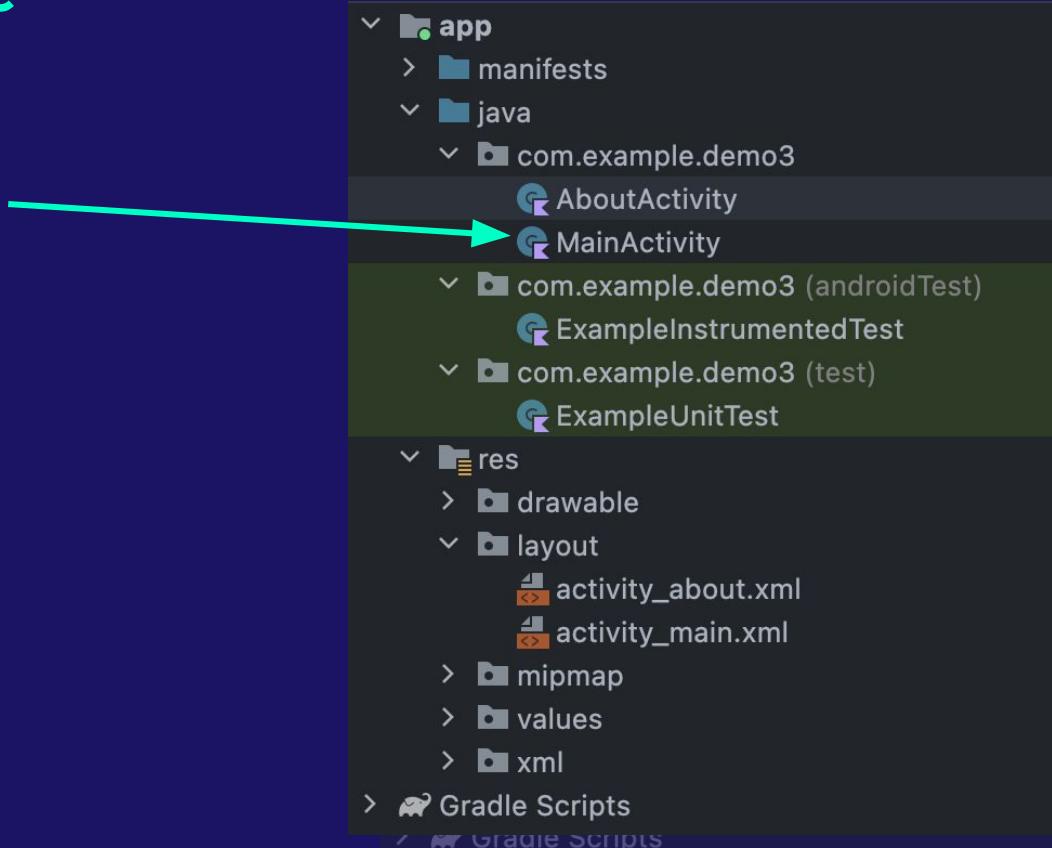
Our Screen



Här har vi ett ID med namnet 'textInput_username'

Navigate

Navigera in hit!



Resultat: Två extra filer!

Navigate

Initialisera knappen + onClick metod

```
val button: Button = findViewById(R.id.btn_navigate)

button.setOnClickListener {

}
```

Navigate

Här har vi ett intent!

```
val button: Button = findViewById(R.id.btn_navigate)

button.setOnClickListener {
    val intent = Intent()
}
```

Navigate

Parametrar

```
val button: Button = findViewById(R.id.btn_navigate)

button.setOnClickListener {
    val intent = Intent(this, AboutActivity::class.java)
}
```

Här ser vi också något som heter *Intent()*

Här kan vi mata in 'context' & *klassen* vi skall navigera mot'

Navigate

Instanzierung!

```
val button: Button = findViewById(R.id.btn_navigate)

button.setOnClickListener {
    val intent = Intent(this, AboutActivity::class.java)
}
```

Navigate

Starta ny aktivitet

```
val button: Button = findViewById(R.id.btn_navigate)

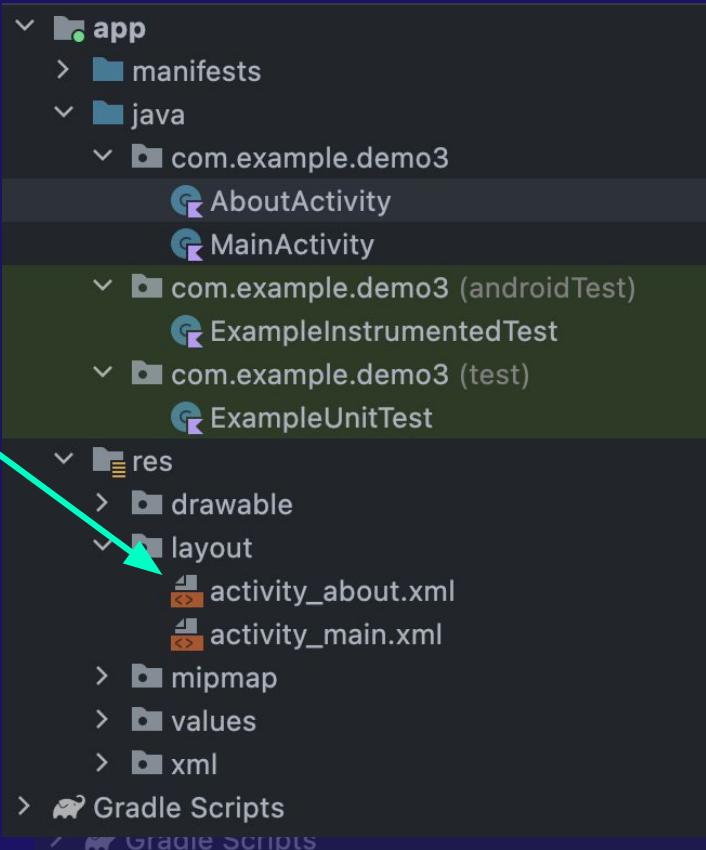
button.setOnClickListener {
    val intent = Intent(this, AboutActivity::class.java)
    startActivity(intent)
}
```

Avsluta med att starta den nya aktiviteten!

Navigate

Navigera in hit!

Låt oss ändra på 'About' så att vi kan se att vi faktiskt nавигират!



Resultat: Två extra filer!

Navigate

```
<TextView  
    android:id="@+id/textView2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="ABOUT PAGE" ←  
    tools:layout_editor_absoluteX="181dp"  
    tools:layout_editor_absoluteY="225dp" />
```

Ignorera
varning

Vi inkluderar en väldigt enkel text bara så att vi kan se ett resultat.
Ignorera mina 'absoluteX & absoluteY' och lägg till constraints istället!

12:22



demo3

Navigate

Nu har vi successivt lyckats navigera!

Hello World!

BUTTON

Om något inte fungerar:

- + Kör applikation i DEBUG MODE
- + Följ felmeddelandet
- + Följ slides igen!
- + Google sökning om något ej ger svar!



Problem

Hur kan vi navigera?

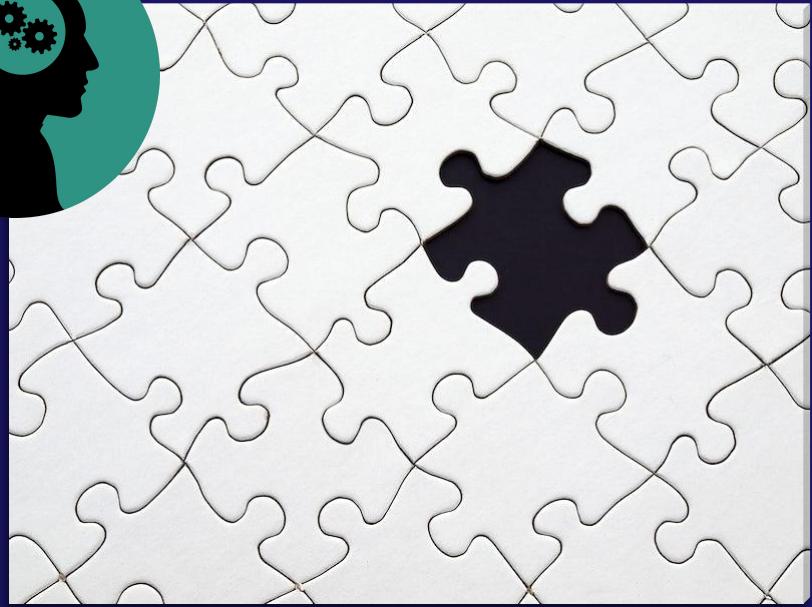


Solution

Med 'INTENT' och startActivityForResult(intent)
Här behöver vi också såklart en sekundär aktivitet att navigera TILL!



Frågor?



03

Sending Data

Scope Functions



Scope Functions

The Kotlin standard library contains several functions whose sole purpose is to execute a block of code within the context of an object. When you call such a function on an object with a lambda expression provided, it forms a temporary scope. In this scope, you can access the object without its name. Such functions are called *scope functions*. There are five of them: `let` , `run` , `with` , `apply` , and `also` .

<https://kotlinlang.org/docs/scope-functions.html>

Scope Functions

```
+  
val adam = Person("Adam").apply {  
    age = 20 // same as this.age = 20  
    city = "London"  
}  
println(adam)
```



[Open in Playground →](#)

Target: JVM Running on v.1.8.10

Scope Functions

```
// JAVA SYNTAX
School sti = new School();

sti.name = "Stockholms Tekniska Institut"
sti.nameShort = "STI"
sti.tag = "S.T.I"
```

```
// KOTLIN SYNTAX
val sti = School() {
    this.name = "Stockholms Tekniska Institut"
    this.nameShort = "STI"
    this.tag = "S.T.I"
}
```

Scope Functions

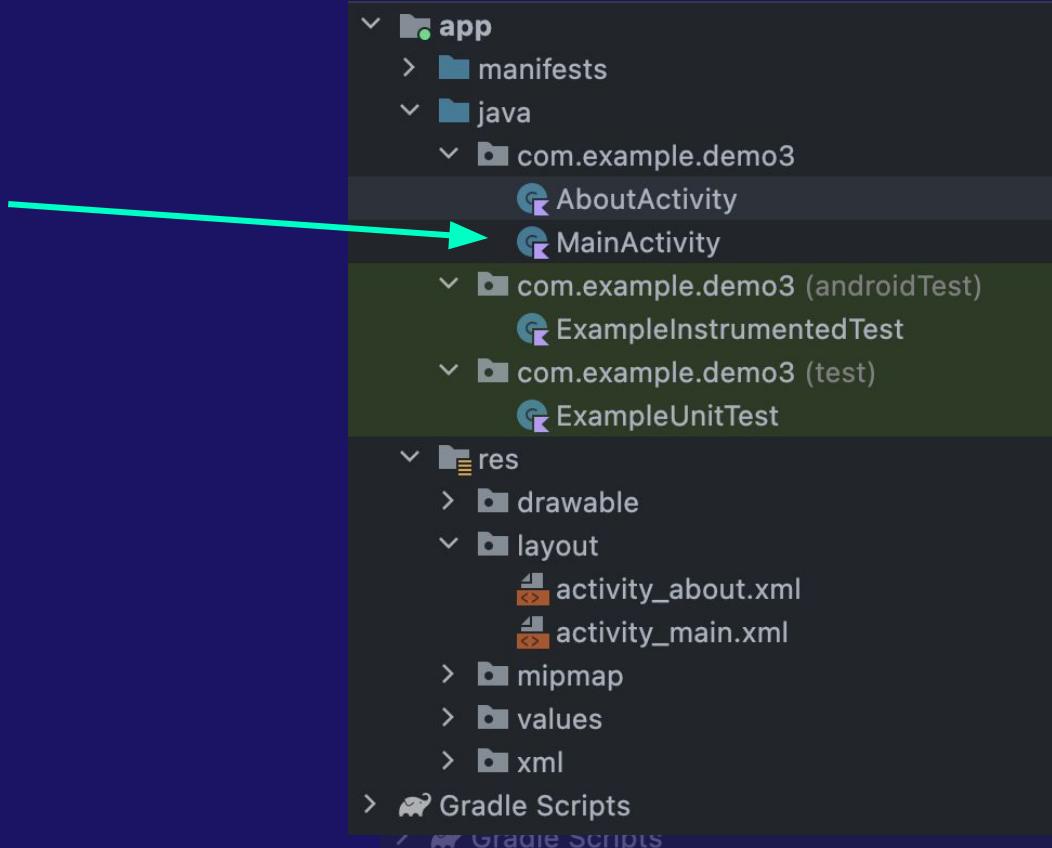
```
// KOTLIN SYNTAX - Without Scope Function
val sti = School() {
    this.name = "Stockholms Tekniska Institut"
    this.nameShort = "STI"
    this.tag = "S.T.I"
}
```

```
// KOTLIN SYNTAX - WITH SCOPE FUNCTION
val sti = School().apply {
    name = "Stockholms Tekniska Institut"
    nameShort = "STI"
    tag = "S.T.I"
}
```

Datatypes #2

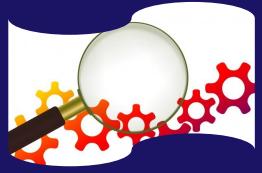


Navigera in hit!



Resultat: Två extra filer!

Analys



Låt oss analysera följande kod....

```
val button: Button = findViewById(R.id.btn_navigate)  
val username: EditText = findViewById(R.id.textInput_username)
```

Analys



```
val username: EditText = findViewById(R.id.textInput_username)
```

button är en Button
username är en EditText

Men vad är innehållet av en 'EditText'?



Change Text

```
val button: Button = findViewById(R.id.btn_navigate)  
val username: EditText = findViewById(R.id.textInput_username)  
  
button.setOnClickListener { it: View!  
    val myUsername: String = username  
    val myUsername2 = username
```

Märk av att det går att skriva:

example = username
Men inte
Example : String = username

Change Text

Type mismatch.

⋮

Required: String

Found: EditText

Change type of 'username' to 'EditText' ⌂ ↗ More actions... ⌂ ↗

EditText är inte en 'STRING' utan 'EditText'



Problem

Hur kan vi hämta sträng
innehållet av en EditText?



Solution

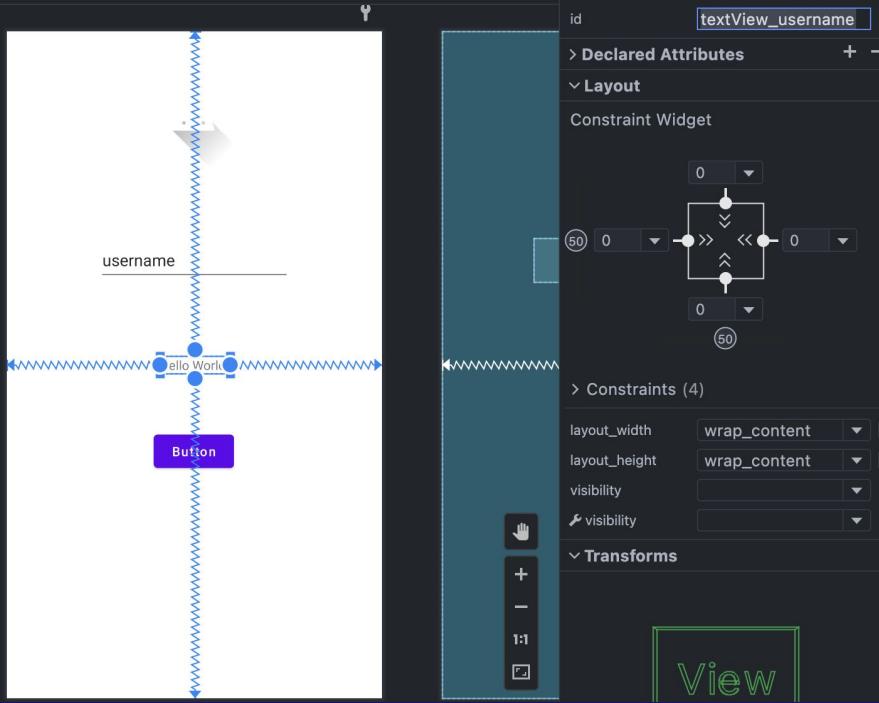
`toString()` konverterar om resultatet
till en sträng!



Change Text



Change Text



TextView

Id: textView_username

Change Text

```
val button: Button = findViewById(R.id.btn_navigate)
val username: EditText = findViewById(R.id.textInput_username)
val displayUsername: TextView = findViewById(R.id.textView_username)

button.setOnClickListener { it: View! }

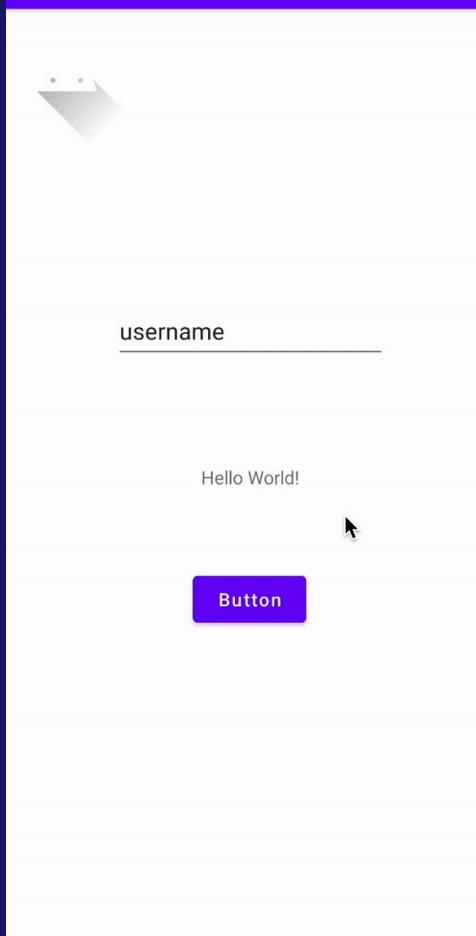
    println(username)

    displayUsername.text = username.toString()

/*
    val intent = Intent(this, AboutActivity::class.java).apply {
        putExtra("username", username.toString())
    }
    startActivity(intent)*/
```

textView.text =

Värdet inom 'editText'



However...

```
button.setOnClickListener { it: View!  
  
    println(username)  
  
    println(username.toString())  
    displayUsername.text = username.toString()
```

androidx.appcompat.widget.AppCompatEditText{53d25f2
FED..CL 250,621-830,753 #7f0801b9 app:id/
...}

username



Hello World!

Button

However...

```
button.setOnClickListener { it: View!  
  
    println(username)  
  
    println(username.toString())  
    displayUsername.text = username.text.toString()
```

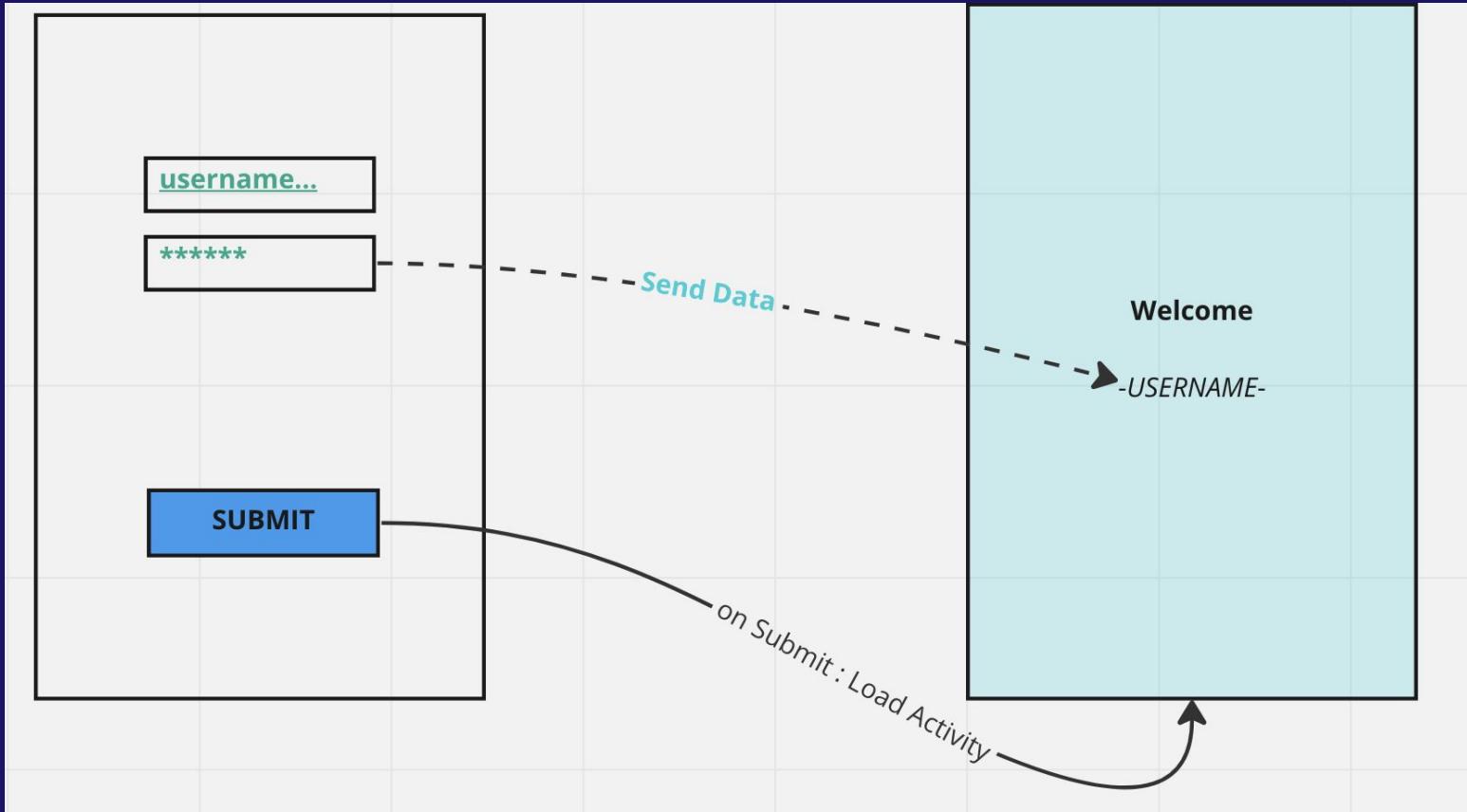
Frågor?



Change Text



Sending Data



Manipulate Data

```
// Define Components
val button: Button = findViewById(R.id.btn_navigate)
val username: EditText = findViewById(R.id.textInput_username)

button.setOnClickListener {
    println(username)

    val intent = Intent(this, AboutActivity::class.java)
    startActivity(intent)
}
```

Skapandet av ett Intent är viktigt för att kunna starta en ny aktivitet!

Manipulate Data

```
// Define Components
val button: Button = findViewById(R.id.btn_navigate)
val username: EditText = findViewById(R.id.textInput_username)

button.setOnClickListener {
    println(username)

    val intent = Intent(this, AboutActivity::class.java).apply {
        putExtra("username", username.toString())
    }

    startActivity(intent)
}
```

*Slutligen lägger vi till .apply{}
Och en 'putExtra()'*

Analys

```
val intent = Intent(this, AboutActivity::class.java).apply {  
    putExtra("username", username.toString())  
}
```

putExtra()

Lägger till information inom nyckeln "username"

Detta är viktigt, för att när vi skall hämta ut informationen senare, så använder vi oss utav denna nyckel!

Vad är det vi skall lagra inom nyckeln?

Jo: username.toString()



Problem

Hur kan vi skicka vidare
information genom
aktiviteter?

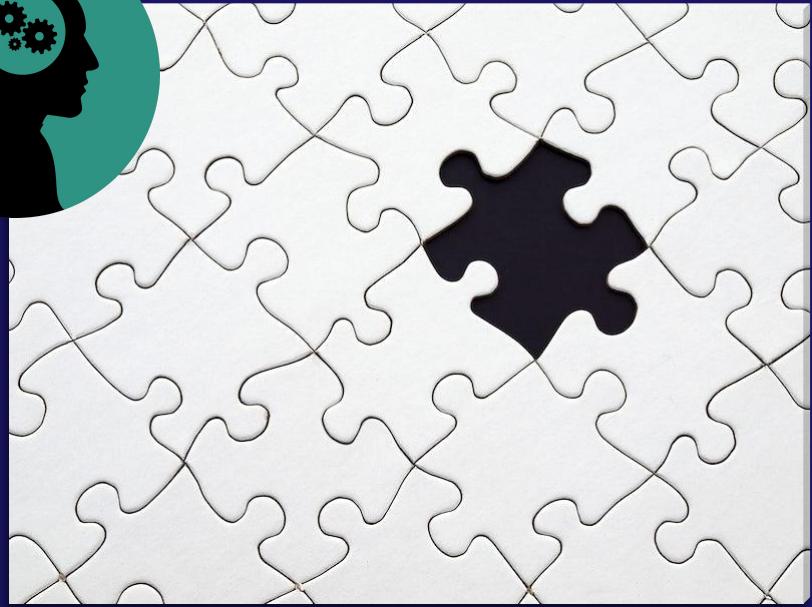


Solution

Ett enkelt sätt är med:
`putExtra()`, vi ett Intent!

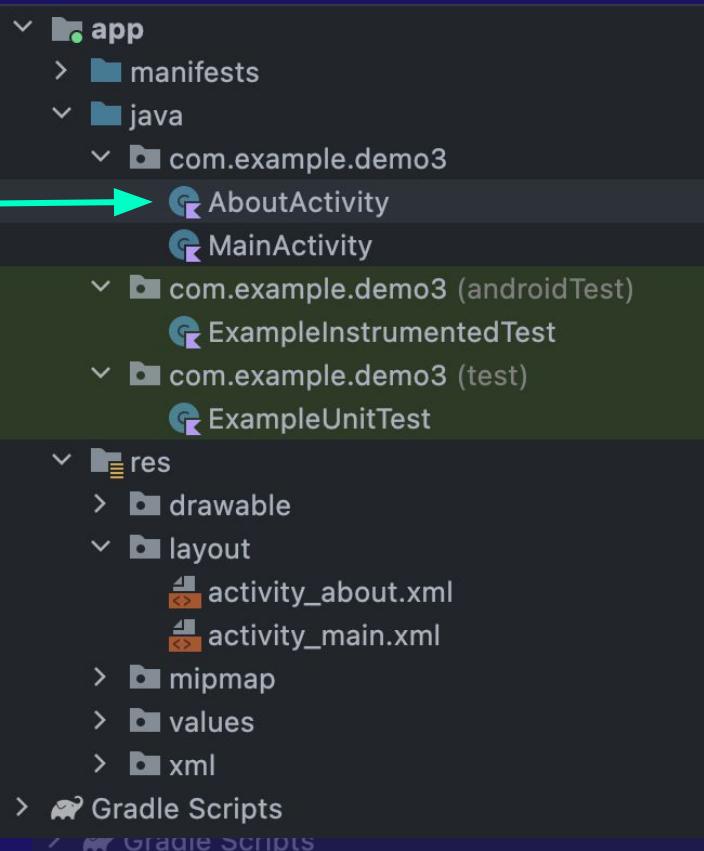


Frågor?



Navigate

Navigera in hit!



Resultat: Två extra filer!

Navigate

```
class AboutActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_about)  
  
        val myIntent = intent.getStringExtra("name: "username")  
    }  
}
```

Hämta ut ett resultat via en nyckel 'username'
myIntent är nu samma som 'String' username!





Problem

Hur kan vi packa upp data
inom en annan aktivitet via
vårt Intent?



Solution

`intent.getStringExtra("key")`

Här måste nyckeln stämma överens
med den information ni matat in!



Frågor?



04

Uppgifter

&

Eget Arbete

Uppgifter

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.



MINNS DU?

```
// Nämn 3 redskap som tillkommer med  
ett Intent
```

```
// Varför behöver vi ett ID'n till  
komponenter?
```

```
// Vad gör stringGetExtra()  
// Vad gör putExtra()
```

```
1 // -Uppgift #1- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Skapa ett nytt projekt!
```

```
6
```

```
7     Döp projektet till: Lektion_3_uppgifter
```

```
8
```

```
9     Navigera till din activity_main.xml
```

```
10    Efterlikna bilden på höger →
```

```
11
```

```
12    Glöm inte att tillämpa ID'n till komponenter
```

```
13    som du vill arbeta med!
```

```
14 */
```

```
15
```

```
16
```

```
17
```

```
18
```

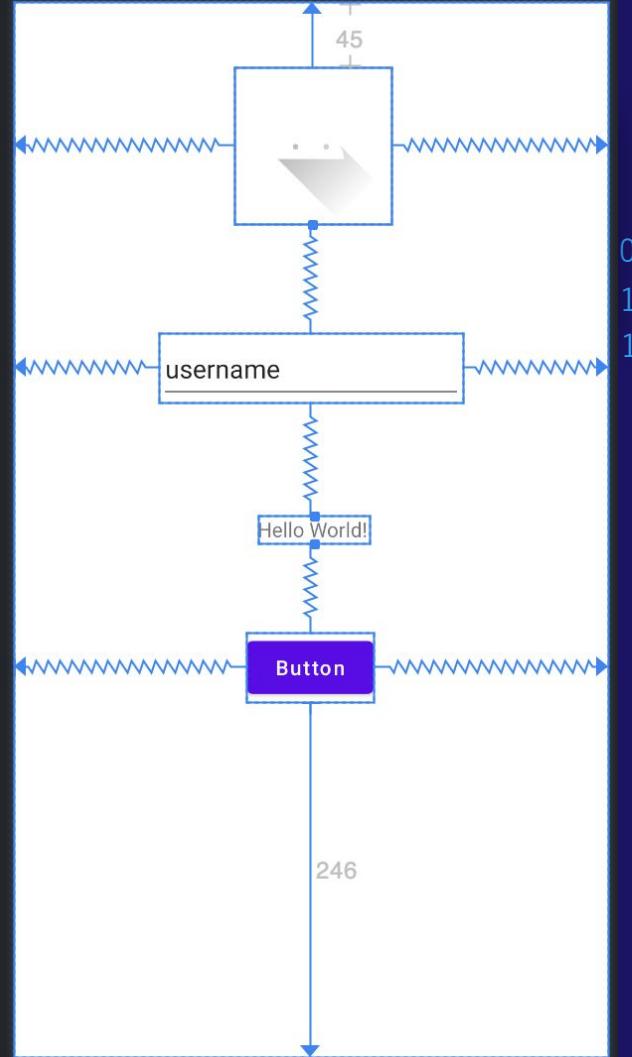
```
19
```

```
20
```

```
21
```

```
22
```

```
23
```



```
1           // -Uppgift #2- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Skapa en ny aktivitet!
```

```
6
```

```
7     Detta gör du enklast med högerklick på
```

```
8     mappen för dina xml aktiviteter.
```

```
9 */
```

```
10
```

```
11 // HINT & Examples
```

```
12 hint(" Glöm ej att det skall bli en
```

```
13 'Empty Activity'
```

```
14
```

```
15     Slide #32 ")
```

```
16
```

```
17
```

```
18
```

```
19
```

```
20
```

```
21
```

```
22
```

```
23
```



Att skapa en ny aktivitet är
väldigt enkelt!

Om du går in till din
manifest.xml fil, så ser du
att aktiviteten automatiskt
inkluderats!

```
1           // -Uppgift #3- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Försök nu att navigera till den
```

```
6     nya aktiviteten vi ett 'Intent'
```

```
7
```

```
8     Glöm ej onClickListener{}
```

```
9 */
```

```
10 // HINT & Examples
```

```
11 hint("
```

```
12     Glöm ej att du behöver dina id'n
```

```
13
```

```
14     Kommer du ihåg hur man instansierar ett objekt
```

```
15     som t.ex. Ett Intent?
```

```
16
```

```
17     Slide #41
```

```
18     ")
```

```
19
```

```
20
```

```
21
```

```
22
```

```
23
```



Intent delen kan vara klurig, här
ser vi ny syntax för just Kotlin
som kan känna svår att bekanta
sig med.

Notera och analysera varje ord i
koden, lista ut vad dem gör så
blir kunskapen din...

```
1 // -Uppgift #4- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4     Skicka vidare data via ett Intent
```

```
5
```

```
6     Packa sedan up datan inom aktiviteten
```

```
7     du skickat vidare till.
```

```
8
```

```
9     Logga resultatet via en SOUT
```

```
10
```

```
11 */
```

```
12
```

```
13 // HINT & Examples
```

```
14 hint("
```

```
15     putExtra
```

```
16     intent.getStringExtra()
```

```
17 ")
```

```
18
```

```
19
```

```
20
```

```
21
```

```
22
```

```
23
```



Nu blir det mycket, minns du hur man gjorde?

Glöm inte .apply() på intent!

```
1           // -Uppgift #5- //
2           THE TOUGH NUT
3
4 /* INSTRUCTIONS
5
6     Skapa en ny aktivitet för 'REGISTER'
7
8     Denna aktivitet skall ha ett input fält för:
9         'Användarnamn'
10        'Lösenord'
11
12     När 'Submit' knappen trycks, läggs
13     denna användare in i en Array.
14 */
15
16 // HINT & Examples
17 hint("
18     Du kan kolla så att strängen inte är tom med
19     username.isEmpty() metoden
20
21     Inom OOP så är ju såklart användaren
22     ett objekt.
23     ")
```



Lycka till, champ!

THANKS !

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

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)*