

</>



Android Studio Intro & Kotlin

" Let's get started! "

INNEHÅLLSFÖRTECKNING

01

Översikt &
Kursplan

03

Programmering
AVD Emulator

02

Java & Kotlin,
Android Studio
Installation

04

Uppgifter
&
Övningar

01

ÖVERSIKT

Kursplan



Förväntningar?

Kursplan och Mål
[kursplan länk](#)

- Utveckling och design av Android applikation
- Publicering av app på Google Play Store
- UI element och layouter specifika för Android
- Anpassning av UI för olika förutsättningar
- Klasser och metoder i Androids API
- UI element samt skapa layouter med hjälp av grafiska verktyg och XML
- Anpassa UI för olika förutsättningar



Förväntningar?

Betygsättning

- Kursbetyget baseras på en sammanvägning av samtliga bedömningsunderlag
- Efter ordinarie tillfälle har den studerande rätten till ytterligare två omprov eller kompletteringar enligt utbildningsplan



Förväntningar?

Betygsgrundande moment:

- Kursens läranderesultat x och x: en skriftlig tentamen (IG/G) ?
- Kursens läranderesultat 1-4 och 9: en skriftliga inlämningsuppgifter (IG/G)
- Kursens läranderesultat 5-7 och 9: en skriftliga inlämningsuppgifter (IG/G)
- Kursens läranderesultat 5-11: en muntlig och en skriftlig projektredovisning (IG/G/VG)



Frågor?



02

Java & Kotlin,
Android Studio Install



“A modern programming language
that makes developers happier”

- Kotlinlang.org



Programming & scripting

Kotlin

2,9%

Go

3,5%

C

6,0%

PHP

6,6%

C++

7,1%

C#

8,8%

Bash/Shell

9,0%

Java

10,3%

JavaScript

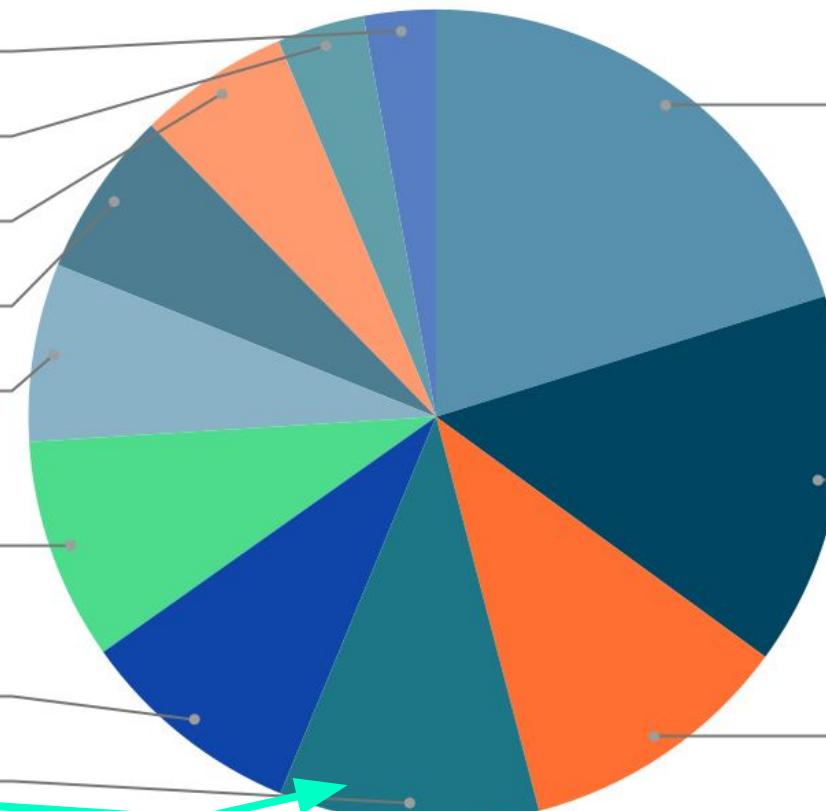
20,2%

Python

14,8%

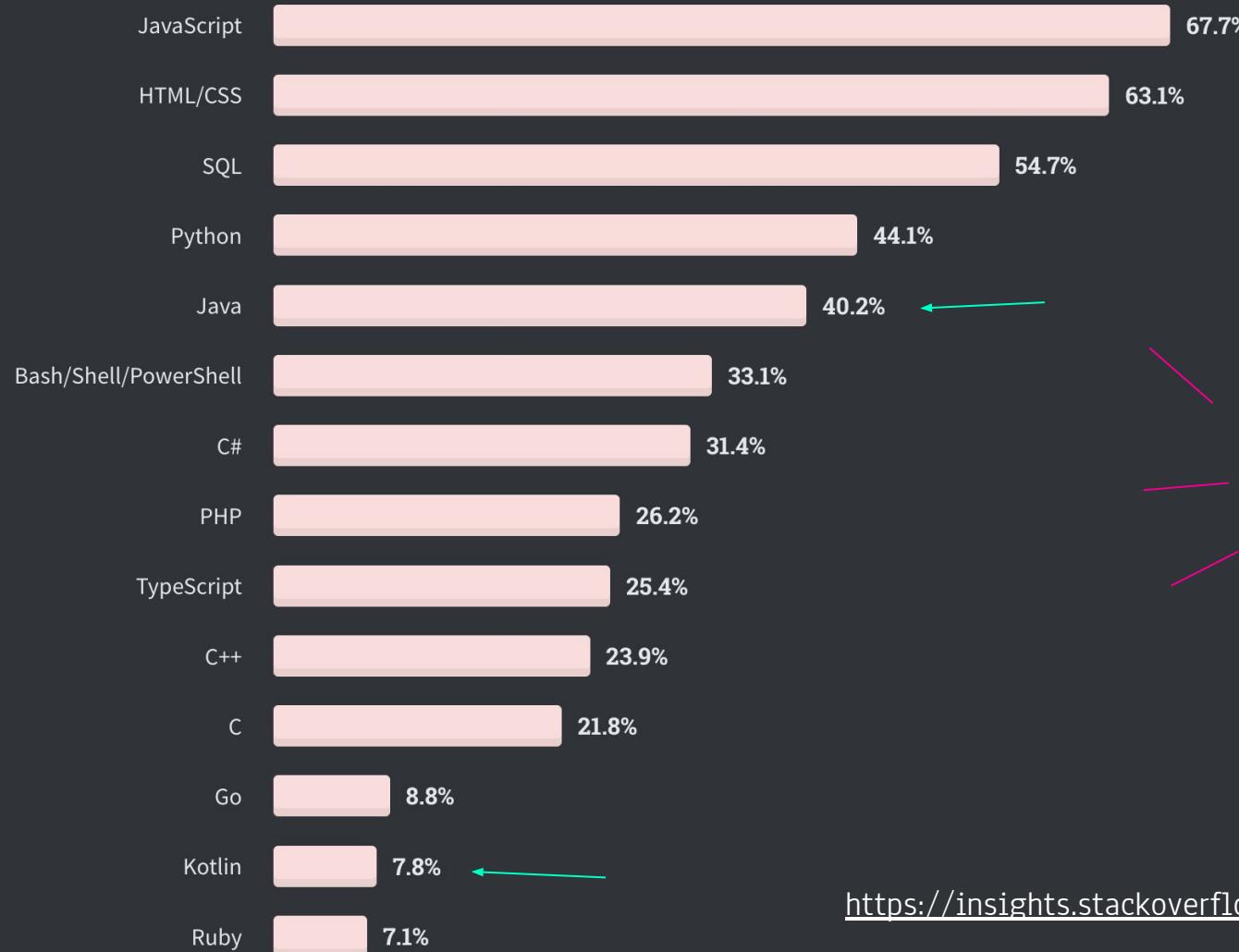
TypeScript

10,9%



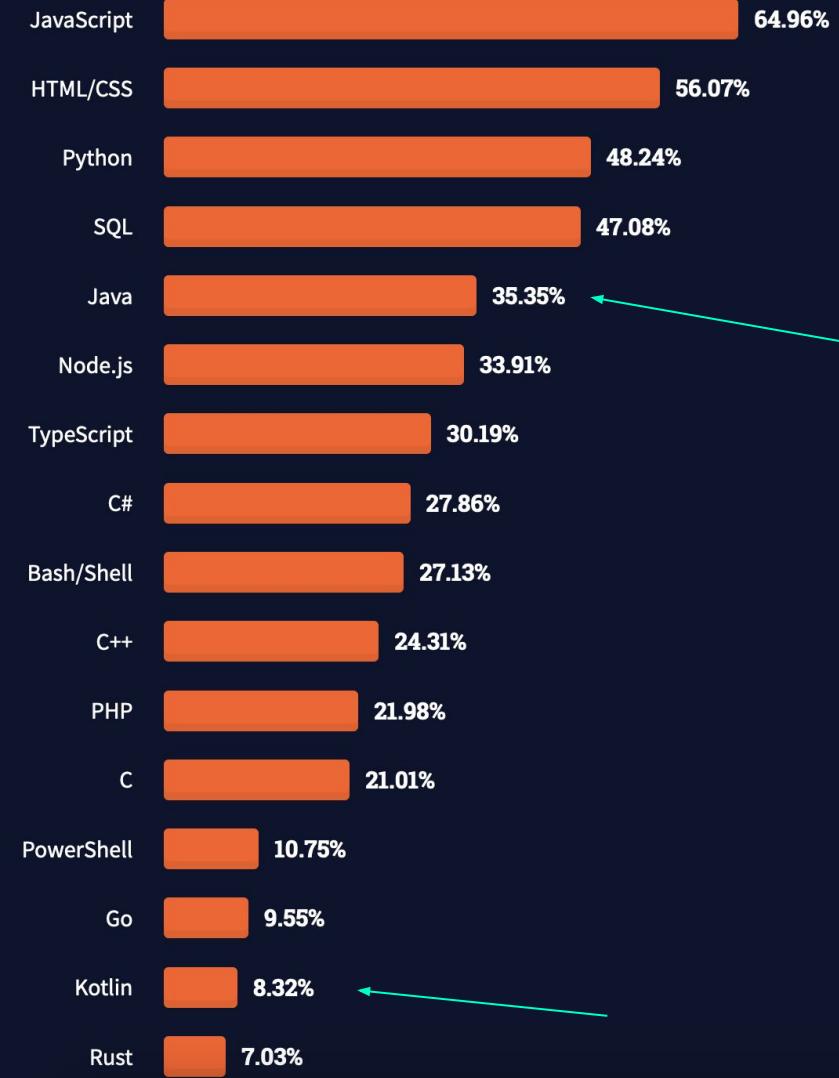
Vad säger Stackoverflow?





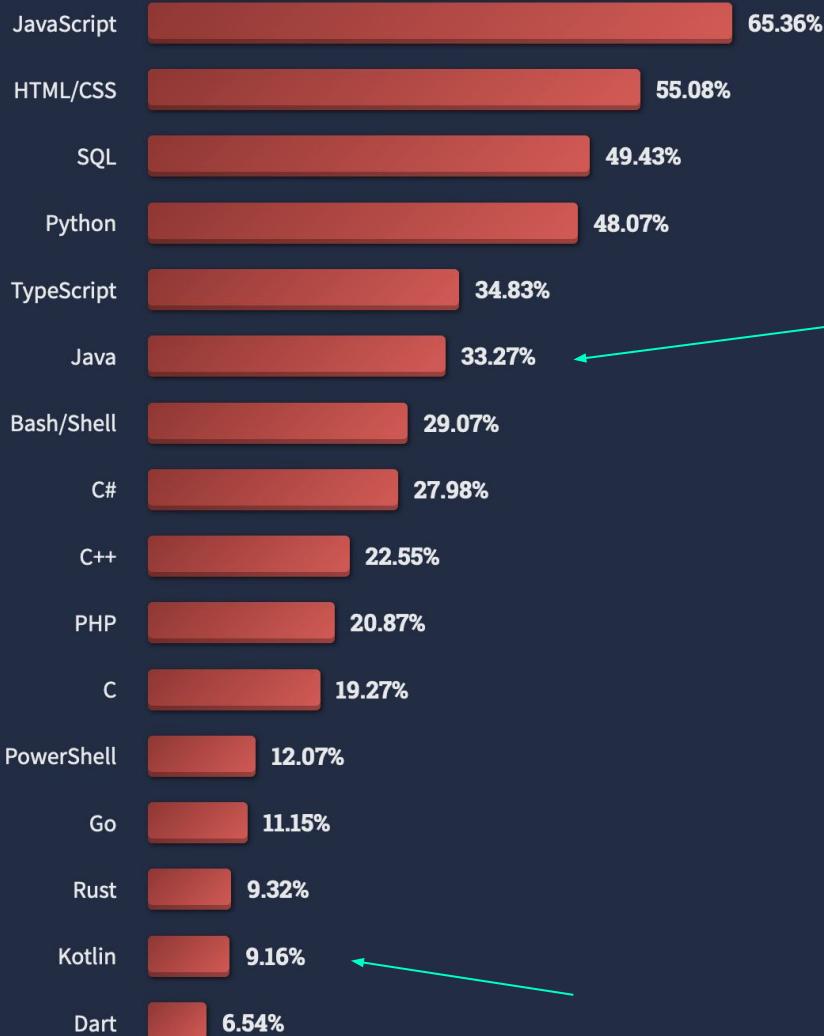
2020

<https://insights.stackoverflow.com/survey/2020>



<https://insights.stackoverflow.com/survey/2021>



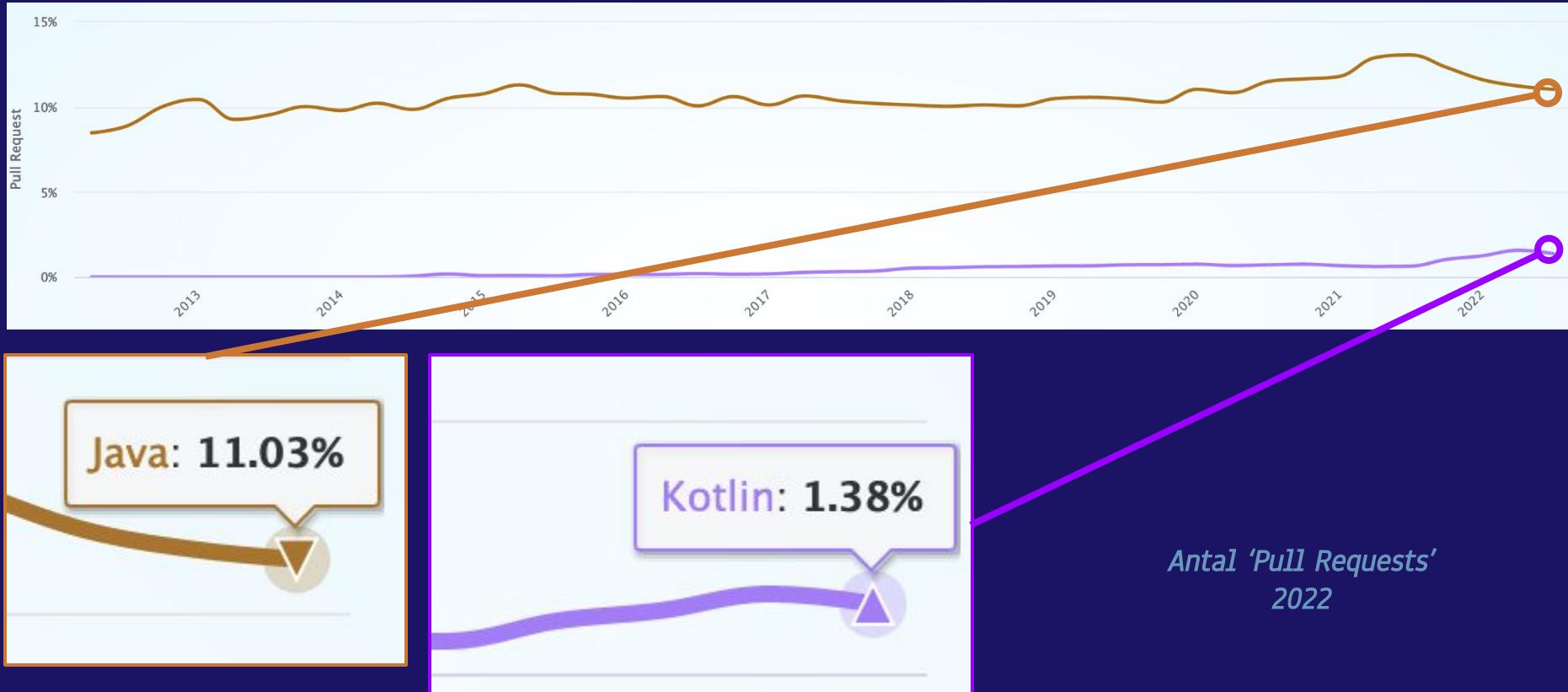


<https://survey.stackoverflow.co/2022/>



Vad säger GitHut?

GitHut 2.0



Out of all languages on projects at GitHub
https://madnight.github.io/githut/#/pull_requests/2022/3

Java

33.27 %

*Java still remains at the top
10 programming languages!*

-7.0 %

*Roughly..
Decrease in 2020 to 2022*

1.0 %

Kotlin grew 'barely' by one
percentage

Java

Java kommer inte att dö ut någon gång snart!

- Stort gemenskap med över 10 Mil. utvecklare
- Utvecklas än idag
- Biljoner enheter använder sig fortfarande utav Java
- Java är fortfarande ett otroligt bra språk för byggnation av bl.a. 'Backends'



Why Kotlin?



Why Kotlin?



Multiplatform Mobile

Share the logic of your Android and iOS apps while keeping UX native



Server-side

Modern development experience with familiar JVM technology



Web Frontend

Extend your projects to web



Android

Recommended by Google for building Android apps

Source: <https://kotlinlang.org/>

Why Kotlin?

“Modern, concise and safe programming language

Easy to pick up, so you can create powerful applications immediately.”

Source: <https://kotlinlang.org/>

```
data class Employee(  
    val name: String,  
    val email: String,  
    val company: String  
) // + automatically generated equals(), hashCode(), toString(), and copy()  
  
object MyCompany {  
    const val name: String = "MyCompany"  
}  
  
fun main() {  
    val employee = Employee("Alice",  
        "alice@mycompany.com", MyCompany.name)  
    println(employee)  
}
```

No more GETTERS & SETTERS

In Kotlin, getters and setters are autogenerated:

```
package com.baeldung

class Person {
    var id: Long = 0
    var name: String? = null
    var brand: String? = null
    var price: Long = 0
}
```



Modification of getter/setter visibility can also be changed, but keep in mind that the getter's visibility must be the same as the property's visibility.

In Kotlin, every class comes with the following methods (can be overridden):

- *toString* (readable string representation for an object)
- *hashCode* (provides a unique identifier for an object)
- *equals* (used to compare two objects from the same class to see if they are the same)

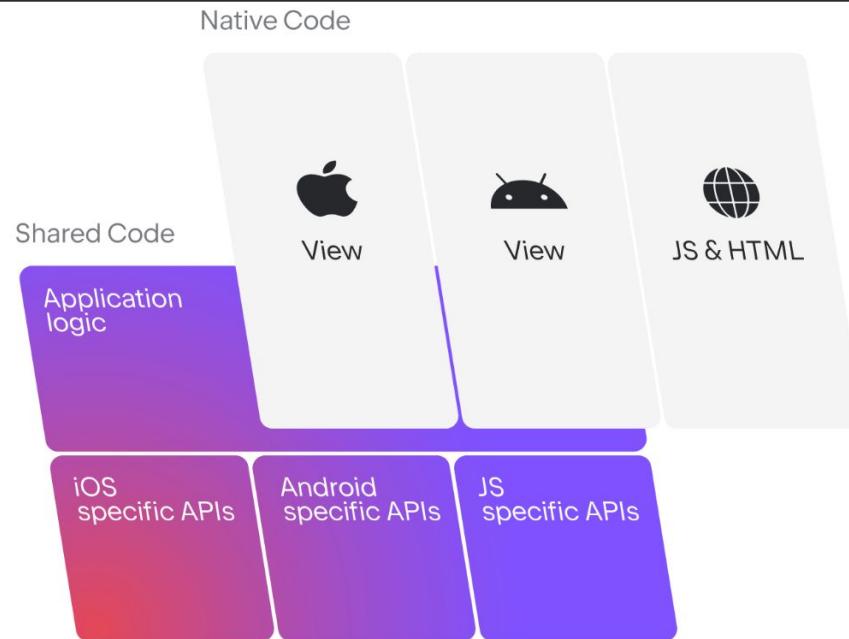
Why Kotlin?

Cross-platform layer for native applications

Share application logic between web, mobile, and desktop platforms while keeping an experience native to users.

Save time and get the benefit of unlimited access to features specific to these platforms.

[Learn about Kotlin Multiplatform →](#)



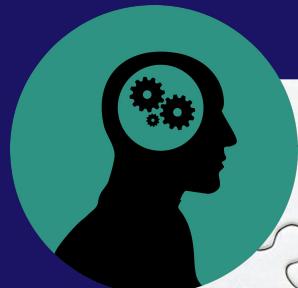
Why Kotlin?

Ramverket *Spring* har även stöd för '*Kotlin*' !

The screenshot shows the Spring Initializr interface. At the top, there's a logo consisting of a green hexagon with white leaf-like patterns and the text "spring® initializr". Below the logo, there are two main sections: "Project" and "Language". In the "Project" section, "Gradle - Groovy" is selected (indicated by a filled green circle). In the "Language" section, "Java" is selected (indicated by a filled green circle). Other options like "Kotlin" and "Groovy" are shown with unfilled circles. A vertical double-headed arrow is positioned between the "Language" section and the bottom right corner of the slide.

Project	Language
<input checked="" type="radio"/> Gradle - Groovy	<input checked="" type="radio"/> Java
<input type="radio"/> Gradle - Kotlin	<input type="radio"/> Kotlin
<input type="radio"/> Maven	<input type="radio"/> Groovy

Frågor?



Talk LAB



3 min

Discuss the following topics:

Vad är Kotlin?

Kommer Kotlin slå ut Java?

Vad tror ni om framtiden för Java?

Vilka tillverkade 'kotlin'?

Why Kotlin?

Source: developer.android

Android's Kotlin-first approach



At Google I/O 2019, we announced that Android development will be increasingly Kotlin-first, and we've stood by that commitment. Kotlin is an expressive and concise programming language that reduces common code errors and easily integrates into existing apps. If you're looking to build an Android app, we recommend starting with Kotlin to take advantage of its best-in-class features.

In an effort to support Android development using Kotlin, we co-founded the [Kotlin Foundation](#) and have ongoing investments in improving compiler performance and build speed. To learn more about Android's commitment to being Kotlin-first, see [Android's commitment to Kotlin](#).



Android Studio Installation



Installation



Instruktioner för installering
<https://developer.android.com/studio/install>

WINDOWS

Windows

To install Android Studio on Windows, follow these steps:

- If you downloaded an `.exe` file (recommended), double-click to launch it.
- If you downloaded a `.zip` file:
 1. Unpack the `.zip`.
 2. Copy the **android-studio** folder into your **Program Files** folder.
 3. Open the **android-studio > bin** folder.
 4. Launch `studio64.exe` (for 64-bit machines) or `studio.exe` (for 32-bit machines).
 5. Follow the **Setup Wizard** in Android Studio and install any recommended SDK packages.

The following video shows each step of the setup procedure for the recommended `.exe` download:

VIDEO: <https://developer.android.com/static/studio/videos/studio-install-windows.mp4>

MAC

Mac

To install Android Studio on your Mac, follow these steps:

1. Launch the Android Studio DMG file.
2. Drag and drop Android Studio into the Applications folder, then launch Android Studio.
3. Choose whether to import previous Android Studio settings, then click **OK**.
4. Complete the Android Studio **Setup Wizard**, which includes downloading the Android SDK components that are required for development.

The following video shows each step of the recommended setup procedure:

VIDEO:

<https://developer.android.com/static/studio/videos/studio-install-mac.mp4>

Installation

Choose the type of setup you want for Android Studio:

Standard

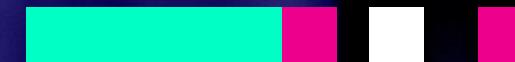
Android Studio will be installed with the most common settings and options.

Recommended for most users.

Custom

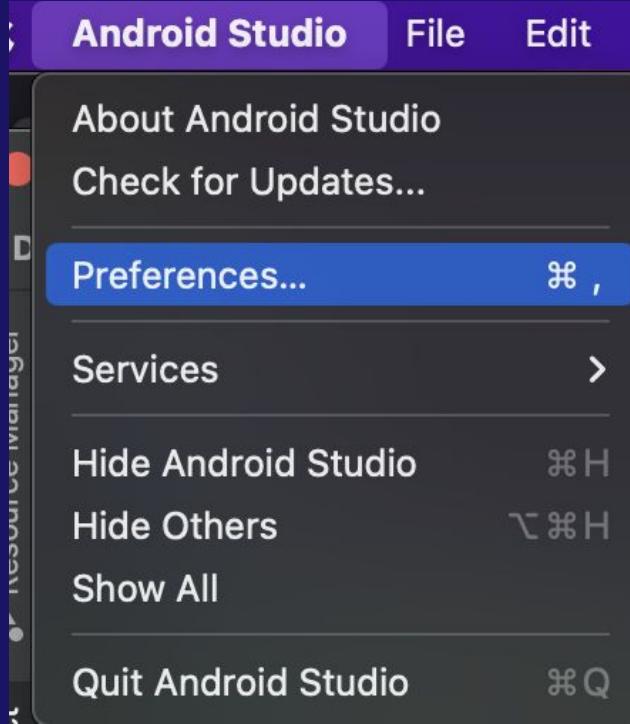
You can customize installation settings and components installed.

Plugins



First Project

Demonstrering sker inom OS: MAC, gränssnitt kan variera!



För att redigera inställningar så
går vi till '**preferences**'

På Windows brukar det ofta vara
inom '**File**' -> '**settings**'

First Project

Plugins jag använder mig av: ‘Rainbow Brackets’, ‘One Dark Theme’

The screenshot shows the Android Studio interface with the 'Plugins' tab selected. A search bar at the top contains the text 'rainbow'. Below it, the 'Search Results (16)' section displays two items:

- Rainbow Brackets**: An icon of blue and yellow brackets. It has 9,2M downloads and a rating of 4.29. An 'Install' button is shown with a green arrow pointing to it.
- One Dark theme**: An icon of a stylized atom or gear. It has 4,8M downloads and a rating of 4.84. An 'Install' button is shown with a green arrow pointing to it.

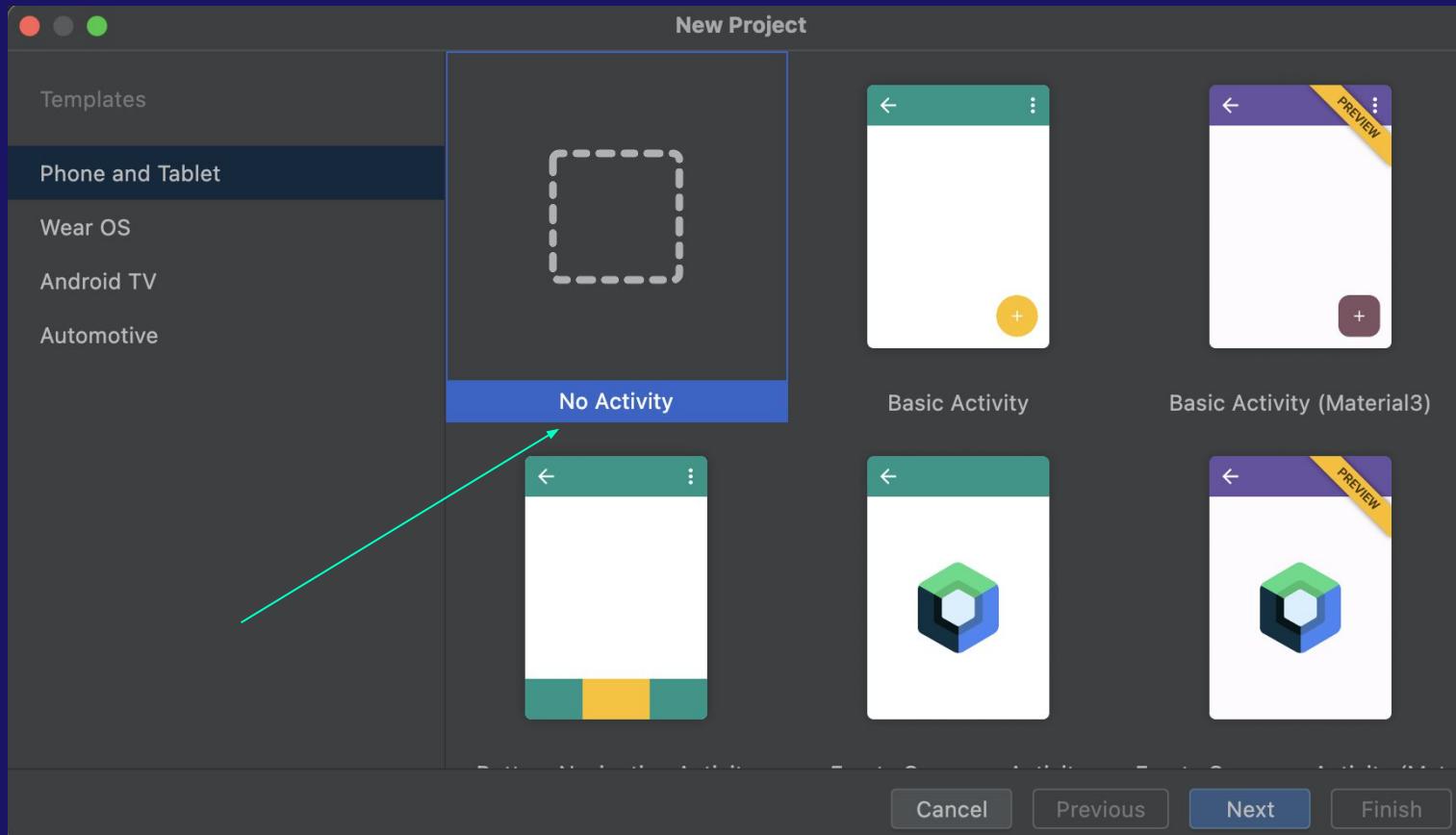
The 'Marketplace' tab is highlighted in blue at the top right of the search bar.

Plugin	Downloads	Rating	Action
Rainbow Brackets	9,2M	4.29	Install
One Dark theme	4,8M	4.84	Install

First Project



New Project



No Activity

Creates a new empty project

Name

Lektion_1_Kotlin

Namnge projektet!

Package name

com.example.lektion_1_kotlin

Save location

/Users/kristofferjohansson/AndroidStudioProjects/Lektion_1_Kotlin



Language

Kotlin

Minimum SDK

API 24: Android 7.0 (Nougat)

Your app will run on approximately **94,4%** of devices.

[Help me choose](#)

Use legacy android.support libraries [?](#)

Using legacy android.support libraries will prevent you from using
the latest Play Services and Jetpack libraries

Verifiera att det
står med Kotlin

Cancel

Previous

Next

Finish

First Project

Första gången ni gör detta så kommer det att ta en stund för projektet att installera moduler och dependencies.

Detta sker enbart en gång!



First Project

TIPS!

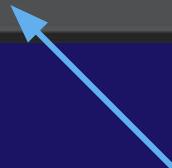


Android Studio performance could be improved

Increasing the maximum heap size from 1280MB...▼

Actions ▾

Don't show again



Om denna ruta visas upp, följ
rekommendationerna och applicera dem - starta
sedan om applikation!

Projekt Struktur

"Varför existerar 3 likadana paket?"

#1 Source Code - Kotlin filer för:

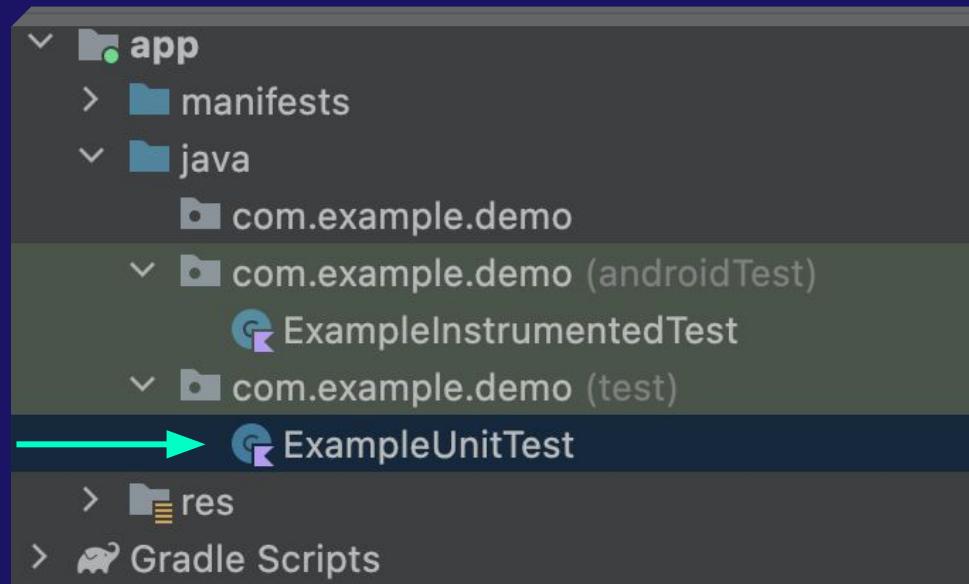
Activities, Services, broadcast receivers,
content providers, models, utilities etc..

#2 AndroidTest - 'instrumented tests'

Source: [What is instrumented tests?](#)

#3 Test - 'test' för enhetstester med JUnit

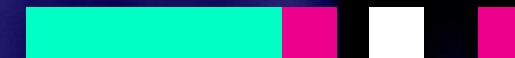
Navigera Hit



03

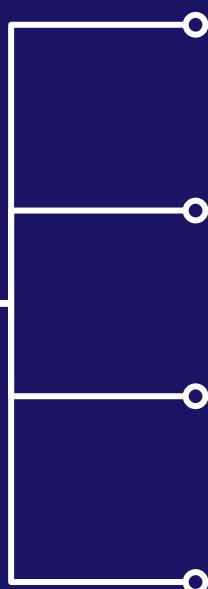
Kotlin Programmering AVD / Emulator

Java & Kotlin Syntax Jämförelser



Intro

**Android
Studio**
Kotlin & Java



Console Log

Hur loggar vi resultat för debugging?

Metod / Function

Vilka skillnader finns på metoder?

Variabler

Hur definierar vi variabler, är detta typsäkert?

Unit Test

Hur kan vi enkelt komma igång med tester?

Print()

```
// KOTLIN SYNTAX  
println("Hello, World!")
```

```
// JAVA SYNTAX  
System.out.println("Hello, World!");
```

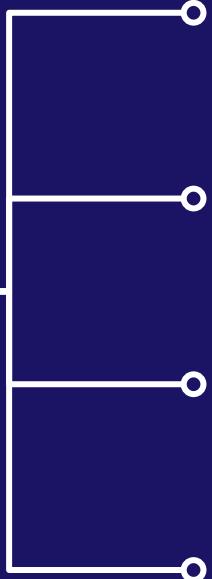
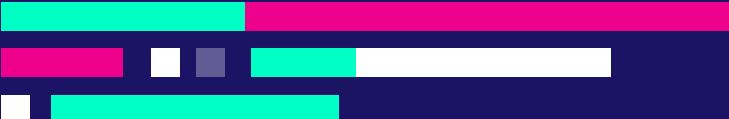
Debugging

Mindre syntax
Inga semikolon

Denna funktion loggar i konsolen "Hello, World!"
detta bör inte vara något som är förvånansvärt, men vi
börjar enkelt!

Variabler

Android
Studio
Kotlin



Int, String

Vi börjar med en versal, till skillnad från Java som kör små bokstäver!

Arrayer

Här kan vi specificera datatypen för en array med IntArray, BooleanArray etc..

Variabler

Vi definierar variabler med nyckelorden:
var & val

Datatyper

Varje gång vi ska definiera en datatyp för variabler, så skriver vi:
`var 'name' : 'datatype' = value`

```
// KOTLIN SYNTAX  
val a: Int  
val b = 21  
  
var c: Int  
var d = 25  
  
d = 23  
c = 21
```

```
// JAVA SYNTAX  
final int a;  
final int b = 21;  
  
int c;  
int d = 25;  
  
d = 23;  
c = 21;
```

Variables

Inom Kotlin, skriver vi datatypen på höger sida av namn deklaration. Märk av att vi har **kolon** som symboliseras datatypen.

Val = final / const

Var = Generell variable

För att tilldela ett nytt värde skriver vi bara:
namnet på variabeln = nytt värde

```
// KOTLIN SYNTAX
val name = "John"
val lastName = "Smith"
val result = "My name is: $name $lastName"
val otherText = "My name is: ${name.substring(2)}"

val text = """
    First Line
    Second Line
    Third Line
"""

print("$name $lastName $result $otherText $text")
```

```
// JAVA SYNTAX
String name = "John";
String lastName = "Smith";
String text = "My name is: " + name + " " +
lastName;
String otherText = "My name is: " +
name.substring(2);

String text = "First Line\n" + "Second Line\n" +
"Third Line";
```

Strings

Concatenate AKA sammanfogning av strängar, är otroligt mycket lättare inom Kotlin, och gör utveckling mycket snabbare!

Märk av att vi också kan kalla på metoder inom strängar!

`$()` ← Indikerar metod anrop!

`name.substring()` ← indikerar startpunkt på strängen

John blir då istället → 'hn'
Vi börjar på index 2 trots allt!

Loops

Vi kan skriva for loops inom en rad.

Vi kan såklart också lägga till en body { }

```
// KOTLIN SYNTAX
for (item in collection) println(item) For Each

for (i in 10 downTo 0) println(i) // Counts down
for (i in 0..10) println(i)    // Counts up
for (i in 10 downTo 0 step 2)      // Every 2:nd
```

Arrays

När vi tittar på 'arrays' så ser vi att det finns alternativ för 'list' och såklart traditionella arrayer!

```
// KOTLIN SYNTAX
val myArray: IntArray = intArrayOf(0, 2, 5)
val myList: ArrayList<Int> = ArrayList(arrayListOf(0, 15, 25))

for (item in myArray) {
    println(item)
}

for (item in myList) {
    println(item)
}
```

Man brukar prata om .indices()
När det kommer till loops + arrayer.

Vi kommer se exempel på detta senare...

```
// KOTLIN SYNTAX

val x = 5 // some value

val xResult = when (x) {
    0, 11      -> "0 or 11"
    in 1..10    -> "from 1 to 10"
    !in 12..14  -> "not from 12 to 14"

    else -> if (isOdd(x)) {
        "is odd"
    } else {
        "otherwise"
    }
}
```

when()

When AKA Switch: är ett alternativ mot fleraltif-satser!

```
// KOTLIN SYNTAX

val colors = setOf("Red", "Green", "Blue")

for (color in colors) {
    when(color) {
        "Red" -> break
        "Green" -> continue
        "Blue" -> println("This is blue")
    }
}
```

when()#2

Vi kan också välja att styra flödet i t.ex en for loop!

‘Break’ avbryter processen,
‘Continue’ fortsätter processen

```
// KOTLIN SYNTAX
fun example() {
}

fun calculate(): Int {
    return 2+2
}

fun calculateSubtract (x: Int, y: Int): Int {
    return x - y
}
```



fun()

Metoder skrivs med nyckelordet 'fun'

Returtyp specificeras efter parametrar AKA ()

Parametrar och argumentets datatyp, specificeras
efter variabel namn följt av kolon!

x : String



Problem

Hur kan vi arbeta med Kotlin
om detta är likt Java?



Solution

Dokumentation och även intellisense
i sig kommer känna av Java och
försöka konvertera om detta till
Kotlin.



Frågor?



Emulator

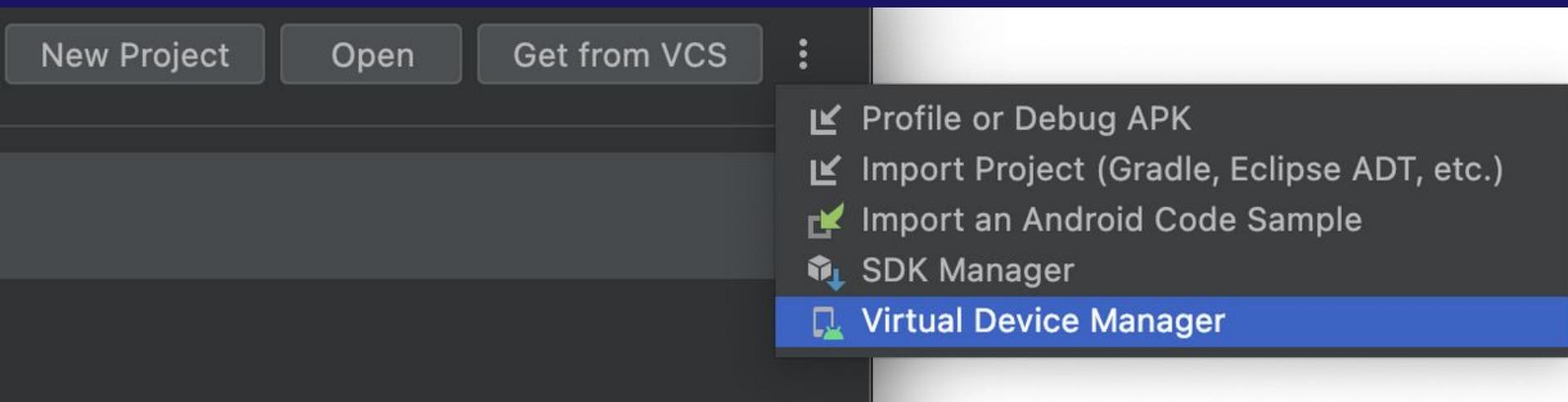


AVD Installation



First Project

Windows:



Welcome to Android Studio

Create a new project to start from scratch.

Open existing project from disk or version control.

MAC



New Project



Open



Get from VCS

More Actions ▾

- Profile or Debug APK
- Import Project (Gradle, Eclipse ADT, etc.)
- Import an Android Code Sample
- SDK Manager
- Virtual Device Manager



Select Hardware

Choose a device definition

Category	Name ▾	Play Store	Size	Resolution	Density
Phone	Pixel XL		5,5"	1440x2...	560dpi
Tablet	Pixel 6 Pro		6,7"	1440x3...	560dpi
Wear OS	Pixel 6		6,4"	1080x2...	420dpi
Desktop	Pixel 5		6,0"	1080x2...	440dpi
TV	Pixel 4a		5,8"	1080x2...	440dpi
Automomo...	Pixel 4 XL		6,3"	1440x3...	560dpi
	Pixel 4	▶	5,7"	1080x2...	440dpi



New Hardware Profile

Import Hardware Profiles



Clone Device...



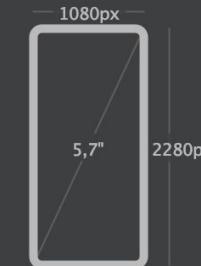
Cancel

Previous

Next

Finish

Pixel 4



Size: large
Ratio: long
Density: 440dpi

Utgå efter 'Play Store' ikoner

AVD Installation

Recommended ARM Images Other Images Tryck på 'ladda ner'

Release Name	API Level ▾	ABI	Target
TiramisuPrivacySandbox ↓	TiramisuPrivacyS	arm64-v8a	Android API T
<i>Tiramisu</i> ↓	33	arm64-v8a	Android 13.0
<i>Sv2</i> ↓	32	arm64-v8a	Android 12L (
<i>S</i> ↓	31	arm64-v8a	Android 12.0
<i>R</i> ↓	30	arm64-v8a	Android 11.0 (
<i>Q</i> ↓	29	arm64-v8a	Android 10.0
<i>Pie</i> ↓	28	arm64-v8a	Android 9.0 (

AVD Installation

Verify Configuration

AVD Name **Pixel 4 API TiramisuPrivacySandbox**

Pixel 4 5.7 1080x2280 440dpi

TiramisuPrivacySandbox Android API TiramisuPrivacySandbox arm64-v8

Startup orientation **Portrait** **Landscape**

Emulated Performance Graphics: Automatic

Show Advanced Settings

?

Cancel Previous Next Finish

AVD Name

The name of this AVD.

Utgå från '**Portrait**' mode, tryck sedan på '**FINISH**'

AVD Installation

Verify Configuration

AVD Name Pixel 4 API TiramisuPrivacySandbox

Pixel 4 5.7 1080x2280 440dpi

TiramisuPrivacySandbox Android API TiramisuPrivacySandbox arm64-v8

Startup orientation

Portrait Landscape

Emulated Performance Graphics: Automatic

Show Advanced Settings

AVD Name

The name of this AVD.

Utgå från '**Portrait**' mode,
tryck sedan på '**FINISH**'

Kan vara bra att kika på HAXM om detta steget klagar!!
<https://github.com/intel/haxm>

Cancel

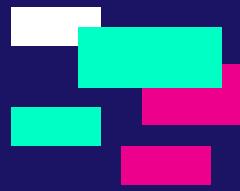
Previous

Next

Finish

AVD Installation

Låt den sedan installera!





Problem

Hur kan vi påbörja Android Studio delen med UI osv..

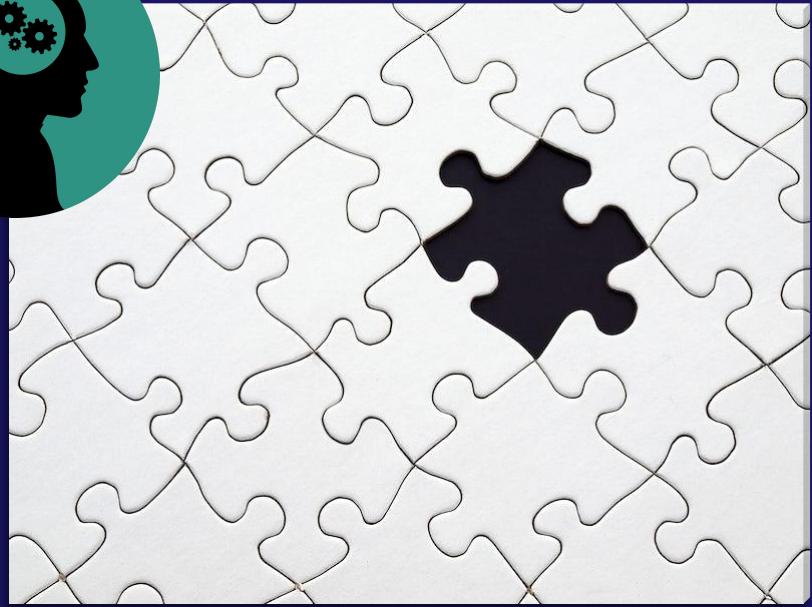


Solution

Android Studio, Kotlin och sist men inte minst : Emulator, krävs för att komma igång propert!



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?

```
// Varför har vi påbörjat programmering  
inom Kotlin men inte Java?
```

```
// Hur kan man skriva en println()  
inom Java?
```

Hur gör vi detta inom **Kotlin**?

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

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

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

```
6
```

```
7     Döp projektet till: Lektion_1_uppgifter
```

```
8
```

```
9
```

```
10    Försök nu att skriva ut: “Hello World”
```

```
11 */
```

```
12
```

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

```
14 hint(" Slide #43 - First Project, går igenom
```

```
15 stegen om ni glömt!
```

```
16
```

```
17 'No Activity', skall skapas
```

```
18
```

```
19 ln inom println() står för 'ny rad' ")
```

```
20
```

```
21
```

```
22
```

```
23
```



Kom igång enkelt med uppgift #1

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

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Skapa en variabel för följande datatyper:
```

```
6     + Int
```

```
7     + String
```

```
8     + Double
```

```
9     + Long
```

```
10    + Boolean
```

```
11
```

```
12     Gör nu om dessa till en 'konstant'.
```

```
13
```

```
14     Glöm inte att tilldela variablerna
```

```
15     ett värde!
```

```
16 */
```

```
17
```

```
18 // HINT & Examples
```

```
19 hint(" Slide #52 går igenom variabler.
```

```
20
```

```
21 Prova gärna att plussa variablerna med varandra
```

```
22 inom en Println() så får du övat på debugging!
```

```
23     ")
```



*Övning ger färdighet, börja enkelt
med variabler!*

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

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Skapa en vanlig Array
```

```
6     Skapa en ArrayList
```

```
7
```

```
8     Tilldela initiala värden av valfri datatyp.
```

```
9     Antalet element bestämmer du själv.
```

```
10
```

```
11    När ni är klara, skriv ut sista elementet.
```

```
12 */
13
```

```
14 // HINT & Examples
```

```
15 hint(" Slide #55 går igenom Arrays.
```

```
16
```

```
17 För att komma åt sista elementet behöver vi komma
```

```
18 åt storleken på arrayen på något vis.
```

```
19 ")
```

```
20
```

```
21
```

```
22
```

```
23
```

0 1 0 1 0
0 1 0 1 0
0 1
0
0 1 0 0 0 0 0 1 0 1
0 1 0 0 0 0 0 1 0 1
0 0
0

Uppgift #3

Arrays är en *datastruktur*, storlek kan variera. Detta är viktigt när vi försöker komma åt 'sista' värdet i datastrukturen.

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

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Skapa en for loop för varje punkt, som kan:
```

```
6         + Räkna upp från 0 till 25
```

```
7         + Räkna ner från 20 till 5
```

```
8         + Räknar upp via vartannat värde till 50
```

```
9         + Skriver ut varje element inom 'array'
```

```
10 */
```

```
11
```

```
12 // HINT & Examples
```

```
13 hint("
```

```
14     Slide #54 går igenom Loops.
```

```
15     Slide #55 går igenom Arrays.
```

```
16
```

```
17 För att komma åt sista elementet behöver vi komma
```

```
18 åt storleken på arrayen på något vis.
```

```
19     ")
```

```
20
```

```
21
```

```
22
```

```
23
```



A binary code graphic consisting of a grid of 0s and 1s. The grid has 10 columns and 10 rows. A cyan rounded rectangle highlights the text "Uppgift #4".

0	1	0	1	0	0	1	0	1	0
0	1	0	1	0	0	0	1	0	1
0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1

*For loops - är ett sätt att upprepa sig själv utan att behöva skriva om kod multipla gånger.
Använd dig av denna metod för att lösa problem som kräver upprepning!*

Minska din börd

```
1 // -Uppgift #5- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Med en vanlig array,
```

```
6     sätt en 'initial' storlek på arrayen.
```

```
7     Därefter, sätt alla dessa värden med en
```

```
8     symbol.
```

```
9 */
```

```
10
```

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

```
12 hint(" Detta har vi ej gått igenom!
```

```
13 Leta efter svaret här: kotlin docs ")
```

```
14
```

```
15
```

```
16
```

```
17
```

```
18
```

```
19
```

```
20
```

```
21
```

```
22
```

```
23
```

0 1 0 1 0
0 1 0 1 0
0 1
0
0 1 0 0 0 0 0 1 0 1
0 1 0 0 0 0 0 1 0 1
0 0
0

Uppgift #5

Likt inom Java, så går det att sätta ett initialt värde på en array vilket sätter storleken på arrayen!

RESURSER

Kotlin dokumentation:

<https://kotlinlang.org/docs/getting-started.html>

Android Studio:

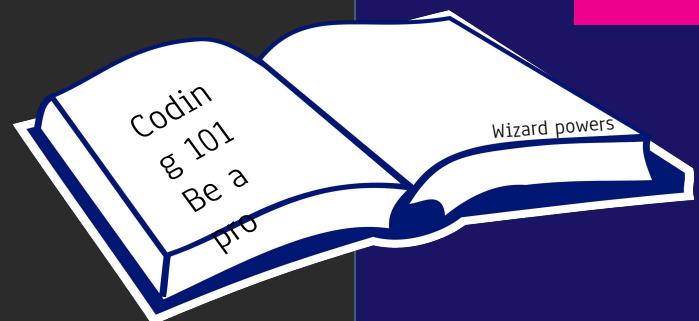
<https://developer.android.com/studio/install>

Kotlin dokumentation Baeldung:

<https://www.baeldung.com/kotlin/>

Keywords

<https://kotlinlang.org/docs/keyword-reference.html#hard-keywords>



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)*