

1. Instal·lació de Kotlin



Continguts

1	Kotlin	3
1.1	Preparant l'entorn	3
1.1.1	Instal·lació del JDK	3
1.1.2	Instal·lació de Kotlin	5
1.1.3	Ús bàsic de Kotlin	7
1.1.4	Extensions de Kotlin per a VSCode	7

1 Kotlin

Kotlin és un llenguatge de programació multiplataforma, en el sentit més ample del terme, creat per JetBrains (els creadors dels IDEs IntelliJ Idea i GoLand).

El seu compilador suporta la generació de codi per a diverses plataformes, entre les que es troba la màquina virtual de Java (JVM) o fins i tot, javascript. El fet de compilar directament sobre JVM, fa que siga totalment compatible amb Java. Actualment és el llenguatge recomanat per Google per al desenvolupament d'aplicacions per a la plataforma Android.

En general, Kotlin presenta els següents avantatges respecte a Java:

- Es tracta d'un llenguatge més concís, pel que s'evita codi innecessari i repetitiu
- És un llenguatge segur, ja que evita errors a l'hora de programar (com el `NullPointerException`),
- És totalment interoperable amb Java, pel que podem utilitzar llibreries java, Android, etc. fins i tot utilitzar ambdós llenguatges en un mateix projecte.

En aquest document anem a fer un repàs als conceptes de Java vistos en primer, i una introducció a partir d'estos al llenguatge Kotlin.

1.1 Preparant l'entorn

Per tal de crear aplicacions compatibles amb la màquina virtual de java amb Kotlin necessitem principalment dos elements:

- El SDK de Java (o **JDK**)
- El **compilador** de Kotlin

I a més, ens serà de gran utilitat algun **editor de codi**.

1.1.1 Instal·lació del JDK

Kotlin compila directament a bytecode de la màquina virtual de Java, per tant, el primer que necessitem és instal·lar el JDK si no el tenim encara.

Important!

La *màquina virtual de Java (JVM)*, junt amb diverses llibreries conformen el *Java Runtime Environment* o *JRE*, i és el que ens permetrà executar aplicacions compilades a bytecode.

Tot i que amb això seria suficient, anem a preparar l'entorn també per al desenvolupament amb

Java, instal·lant l'SDK (*Software Development Kit*) de Java (*JDK-Java Development Kit*) que a més del JRE inclou eines com el compilador `javac` entre d'altres.

El JDK, per la seua banda té diferents implementacions. La que utilitzarem és la implementació lliure, però oficial, OpenJDK (versió 11 LTS).

Per fer la instal·lació amb Linux:

1. Actualitzem la caché de paquets

```
$ sudo apt update
```

2. Instal·lem la versió més recent del JDK:

```
$ sudo apt install default-jdk
```

...

Se instalarán los siguientes paquetes NUEVOS:

```
ca-certificates-java default-jdk default-jdk-headless default-jre default-jre-headless fonts-dejavu-extra java-common libatk-wrapper-java ibatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
```

Com veiem, descarregarà la versió 11 ((`openjdk-11-jdk`). Fixeu-vos que amb les dependències instal·larà el paquet `openjdk-11-jre`, que és la implementació de l'entorn d'execució de Java (JRE), que com hem dit, forma part del JDK.

Aquesta instal·lació, en Ubuntu es realitza a la carpeta `/usr/lib/jvm/java-11-openjdk-amd64`, i es genera l'enllaç `/usr/bin/java` que apunta a `/etc/alternatives/java` que a la seua vegada apunta a `/usr/lib/jvm/java-11-openjdk-amd64/bin/java`. Aquests alternatives ens serviran per utilitzar diferents entorns Java en el nostre sistema. A més de l'enllaç a l'ordre `java` també s'han generat enllaços a `javac`.

Podem consultar la versió de Java en ús amb: `java --version`.

En cas que haverem d'alternar entre diferents versions de Java (per exemple si instal·lem la versió LTS 17 que s'allibera el 14 de setembre de 2021), utilitzaríem l'ordre:

```
sudo update-alternatives --config java
```

3. Actualització del PATH

Finalment, hem d'actualitzar el path i la variable d'entorn `JAVA_HOME`, amb la carpeta d'instal·lació.

Per a això, com a administrador, el més senzill serà crear un script al directori `/etc/profile.d` amb el següent contingut:

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin
```

Qualsevol nom és vàlid, ja que al carregar el perfil es carreguen tots els fitxers d'aquest directori. Al nostre cas, l'hem anomenat com `jdk_home.sh`.

Amb això, es carregaran aquestes variables junt amb la càrrega del perfil, en cada inici de sessió. Si no desitgem eixir, podem carregar-les des del propi terminal amb `source /etc/profile.d/jdk-home.sh`

Amb açò ja tenim Java completament funcional al nostre equip.

Windows

Per a Windows, caldrà descarregar el fitxer `zip` de l'OpenJDK des de la web: <https://openjdk.java.net>, descomprimir-lo, per exemple en `C:\java\jdk11` i afegir a les variables d'entorn del sistema (des de la configuració avançada) la variable `JAVA_PATH`, amb el valor d'aquest directori, i modificar la variable `PATH` per incloure la carpeta `bin` de dins la carpeta `jdk11`.

1.1.2 Instal·lació de Kotlin

Per tal d'instal·lar el compilador de Kotlin, podem optar bé per Ubuntu Make o bé per SDKMan (opció recomanada).

1.1.2.1 Ubuntu Make En Ubuntu disposem de l'eina *Ubuntu Make* (ordre `umake`) per instal·lar diverses eines de desenvolupament de forma senzilla i ràpida. Per instal·lar Kotlin amb Ubuntu Make farem:

```
$ umake kotlin kotlin-lang
```

Açò ens demanarà el path on volem instal·lar-lo. Per omissió, s'instal·la al nostre directori personal `/home/usuari/.local/share/umake/kotlin/kotlin-lang`.

Una vegada instal·lat **cal que eixim de la sessió d'usuari per poder-lo utilitzar**.

1.1.2.2 SDKMan SDKMAN! és una eina per gestionar diversos kits de desenvolupament de programari a la majoria de sistemes basats en Unix. Proporciona una interfície de línia d'ordres (CLI) i una API còmodes per instal·lar, canviar, eliminar i llistar candidats.

En primer lloc, per instal·lar SDKMAN!, des de la línia d'ordres, necessitarem tindre disponibles al sistema les ordres `curl`, `zip` i `unzip`.

El primer que farem serà des d'una terminal llançar:

```
$ curl -s "https://get.sdkman.io" | bash
...
...
All done!
```

Please open a new terminal, or run the following in the existing one:

```
source "/home/usuari/.sdkman/bin/sdkman-init.sh"
```

Then issue the following command:

```
sdk help
```

Enjoy!!!

Açò ens generarà un script al nostre home, que haurem d'invocar amb `source`, tal i com s'indica al final de la instal·lació, i després llançar `sdk help`, per comprovar que està correctament instal·lat i veure l'ajuda:

```
$ source "/home/usuari/.sdkman/bin/sdkman-init.sh"
$ sdk help
```

Ara per instal·lar Kotlin amb SDKMan, simplement farem:

```
$ sdk install kotlin

Downloading: kotlin 1.5.30

In progress...
```

```
##### 100,0%
```

```
Installing: kotlin 1.5.30  
Done installing!
```

```
Setting kotlin 1.5.30 as default.
```

1.1.3 Ús bàsic de Kotlin

Per utilitzar kotlin, podem fer-ho de dues formes; la primera amb la shell interactiva, llançant des de la terminal:

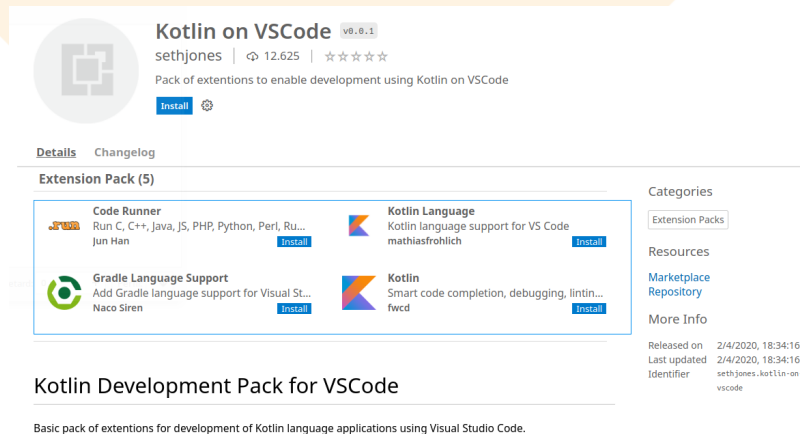
```
$ kotlin  
Welcome to Kotlin version 1.5.30 (JRE 11.0.11+9-Ubuntu-0ubuntu2.20.04)  
Type :help for help, :quit for quit  
>>>
```

Fixeu-vos que us indica la versió de Kotlin i també la del JRE, ja que és l'entorn en què s'executa. En aquesta shell podem realitzar declaracions i llançar ordres de forma interactiva.

I pe altra banda, fent ús del compilador `kotlinc`, per compilar fitxers, i executar-los amb `kotlin`. Ho veurem un poc més avant.

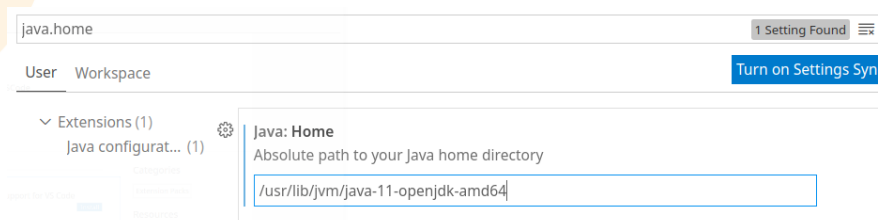
1.1.4 Extensions de Kotlin per a VSCode

Per altra banda, anem a instal·lar les extensions del llenguatge Kotlin per a Visual Studio Code. Per a això anem a la part d'extensions de la barra esquerra, i busquem l'extensió *Kotlin on VSCode*, que és un pack d'extensions amb el suport per al llenguatge:

**Figura 1:** Kotlin on VSCode

Aquest pack inclou les extensions Code Runner, per executar codi, Kotlin i Kotlin Language, per al suport al llenguatge i Gradle Language Support i Gradle Tasks per a la gestió de projectes Gradle.

Si no ho teniu instal·lat prèviament, també ens serà d'utilitat el pack d'extensions *Java Extension Pack*, per treballar també amb Java des de VSCode. A més, una vegada instal·lades les extensions de Java, cal que ajustem la variable `java.home` amb el directori on tenim instal·lat el JDK. Per a això, anem a *File > Preferences > Settings* (o premem `Ctrl+,`), per accedir a la pestanya de preferències, i busquem `java.home` al quadre de cerca.

**Figura 2:** Establir java.home

Fem clic en *Edit in settings.json*, i establim la variable amb el valor que hem obtingut del `$JAVA_HOME` del sistema. Al nostre cas:

```
"java.home": "/usr/lib/jvm/java-11-openjdk-amd64/"
```

Una vegada editat, guardem el fitxer i ja ho tindrem tot configurat.