**Windows — Projet Kotlin + Gradle avec VS Code (pas d’Android Studio)**

**0) Pré-requis à installer (une seule fois)**

**Option A — avec winget (recommandé, PowerShell en admin)**

# 1) JDK 17+

winget install --id Microsoft.OpenJDK.17 -e

# 2) Gradle (facultatif si tu utilises le wrapper après init)

winget install --id Gradle.Gradle -e

# 3) Kotlin (compilateur CLI ; facultatif si tu compiles via Gradle)

winget install --id JetBrains.Kotlin -e

# 4) Visual Studio Code

winget install --id Microsoft.VisualStudioCode -e

**Option B — sans winget**

* JDK 17+ : télécharge/install (Adoptium/OpenJDK), vérifie java -version.
* Gradle : possible d’installer ensuite, **mais** on va s’appuyer surtout sur le **Gradle Wrapper** (gradlew.bat) après l’initialisation.
* VS Code : installe depuis le site officiel.

**Extensions VS Code utiles**

* **Kotlin Language**
* **Gradle for Java** (pour lancer des tâches Gradle)
* **Code Runner** (optionnel)

**Vérifications rapides (dans PowerShell)**

java -version

gradle -v # si installé

kotlinc -version # si installé

✅ Pas obligatoire d’avoir gradle/kotlinc globalement installés si tu utilises Gradle Wrapper après l’initialisation.

**1) Créer le projet (init Gradle)**

Ouvre **PowerShell** (à l’emplacement où tu veux créer ton projet) :

mkdir MonPremierProjet

cd MonPremierProjet

gradle init --type basic

Si tu n’as pas gradle installé et tu préfères 100% sans Gradle global :  
crée d’abord les fichiers Gradle manuellement (je te mets tout ci‑dessous), puis lance directement le wrapper qu’on ajoute au projet (via un zip Gradle wrapper ou en copiant des fichiers du wrapper).  
Le plus simple reste d’avoir **gradle** pour faire gradle wrapper une fois, puis tu n’en as plus besoin globalement.

Ajoute ensuite le **wrapper** pour être autonome :

gradle wrapper

Ça crée gradlew.bat + le dossier gradle/ (wrapper). À partir de maintenant tu peux utiliser **.\gradlew.bat** partout, même sans Gradle global.

**2) Déposer les fichiers de build (Kotlin JVM + application)**

Crée/édite **settings.gradle.kts** à la racine :

rootProject.name = "MonPremierProjet"

Crée/édite **build.gradle.kts** à la racine :

plugins {

kotlin("jvm") version "1.9.10"

application

}

repositories {

mavenCentral()

}

dependencies {

implementation(kotlin("stdlib"))

}

application {

// Le point d'entrée (fichier Main.kt => classe MainKt)

mainClass.set("MainKt")

}

tasks.withType<org.jetbrains.kotlin.gradle.tasks.KotlinCompile> {

kotlinOptions.jvmTarget = "17"

}

Si le dossier build.gradle.kts existant vient de gradle init, **remplace** son contenu par celui‑ci.

**3) Arborescence du code source**

Crée la structure suivante :

MonPremierProjet/

│ build.gradle.kts

│ settings.gradle.kts

│ gradlew.bat

│ gradlew

│ (dossier) gradle/

└─ src/

└─ main/

└─ kotlin/

└─ Main.kt

Crée **src/main/kotlin/Main.kt** avec ce contenu (exemple « 2 TextView » **simulés** en console) :

class TextView(val text: String) {

fun render() = println("📱 [TextView] -> $text")

}

fun main() {

val tv1 = TextView("Bonjour depuis TextView 1 👋")

val tv2 = TextView("Bonjour depuis TextView 2 🚀")

tv1.render()

tv2.render()

}

Ici on **simule** des TextView (car on n’est pas dans Android, mais en console).  
L’objectif est de tester Kotlin + Gradle + VS Code sans Android Studio.

**4) Lancer le projet (build & run)**

Dans le dossier du projet (où se trouve gradlew.bat) :

# 1) Résoudre les dépendances et compiler

.\gradlew.bat build

# 2) Exécuter l'app console

.\gradlew.bat run

**Résultat attendu :**

📱 [TextView] -> Bonjour depuis TextView 1 👋

📱 [TextView] -> Bonjour depuis TextView 2 🚀

**5) (Optionnel) Configuration VS Code**

**tasks.json (pour lancer Gradle facilement)**

Dans .vscode/tasks.json :

{

"version": "2.0.0",

"tasks": [

{

"label": "Gradle: build",

"type": "shell",

"command": ".\\gradlew.bat build",

"group": "build",

"problemMatcher": []

},

{

"label": "Gradle: run",

"type": "shell",

"command": ".\\gradlew.bat run",

"group": "test",

"problemMatcher": []

}

]

}

**launch.json (déboguer via tâche « run »)**

Dans .vscode/launch.json :

{

"version": "0.2.0",

"configurations": [

{

"name": "Run (Gradle)",

"type": "cppdbg",

"request": "launch",

"preLaunchTask": "Gradle: run",

"program": "${workspaceFolder}/build/tmp/does-not-exist.exe",

"stopAtEntry": false

}

]

}

(Astuce : beaucoup lancent simplement la **tâche** Gradle: run via CTRL+SHIFT+P → “Run Task”, sans launch.json.)

**6) Troubleshooting (Windows)**

* **« ‘java’ n’est pas reconnu… »**  
  → installe JDK, ferme/réouvre PowerShell. Vérifie java -version.  
  → si besoin, définis JAVA\_HOME vers ton JDK et ajoute %JAVA\_HOME%\bin au PATH.
* **Gradle introuvable lors du gradle init**  
  → installe Gradle (winget/scoop/choco) — **ou** initialise le projet avec des fichiers fournis, puis utilise **gradlew.bat**.
* **Caractères Unicode (émojis) mal affichés**  
  → dans le terminal VS Code, définis l’encodage en UTF‑8.