

## Q) Is Package and Module being same term in Kotlin?

Ans- In Kotlin, **packages** and **modules** serve different purposes, although they are related:

### 1. Package:

- A **package** is a way to organize code files into namespaces. It allows you to group related classes, functions, and other code elements logically.
- Kotlin files typically start with a `package` declaration, like `package com.example.app`. This helps prevent naming conflicts between classes or functions.
- Packages are mainly about code organization and have no impact on the build or project structure.
- Packages are declared at the beginning of a Kotlin file using the `package` keyword.
- `package com.example.myapplication`

### 2. Module:

- A **module** refers to a set of Kotlin (or Java) code that is compiled together. It can be a project, a library, or a part of a project that can be compiled and run independently.
- Modules can include multiple packages.
- They represent units of code with defined boundaries, dependencies, and build configurations.
- In build systems like **Gradle** or **Maven**, modules are defined in the project's configuration files.
- Multiple modules in a project: In large projects, you might have separate modules for different parts of the app, like app, networking, or database.

Example of a module setup in Gradle:

groovy

Copy code

```
// File: settings.gradle

include ':app', ':network', ':database'
```