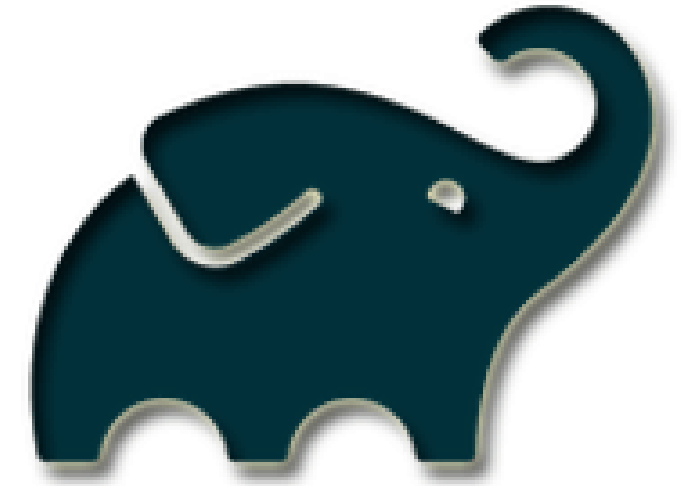




Gradle

Tutorial

# Definition



**Gradle** ( publié en 2007) est un outil open source d'automatisation de construction. Il offre une prise en charge de **la construction**, des **tests** et du **déploiement** de logiciels sur différentes plateformes.



- “ build :
- **compile**
  - **Run tests**
  - **package**
  - **deploy and Run**
- ”

# Why Gradle?

**Multi-Project Build Software**



**incremental builds**



**Build Scans**



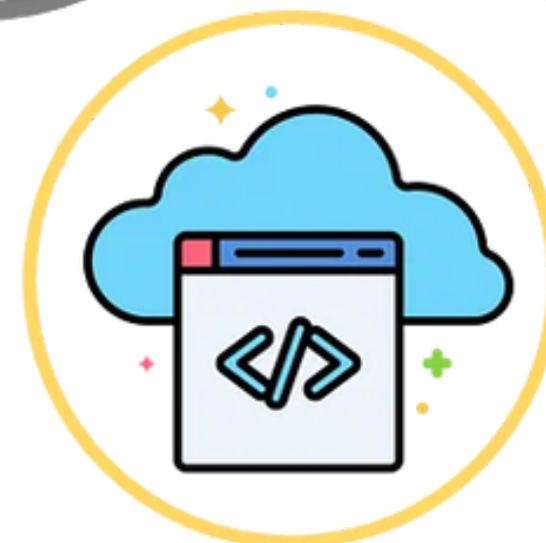
**Provides Support**



**High Performance**



**Familiarity with Java**



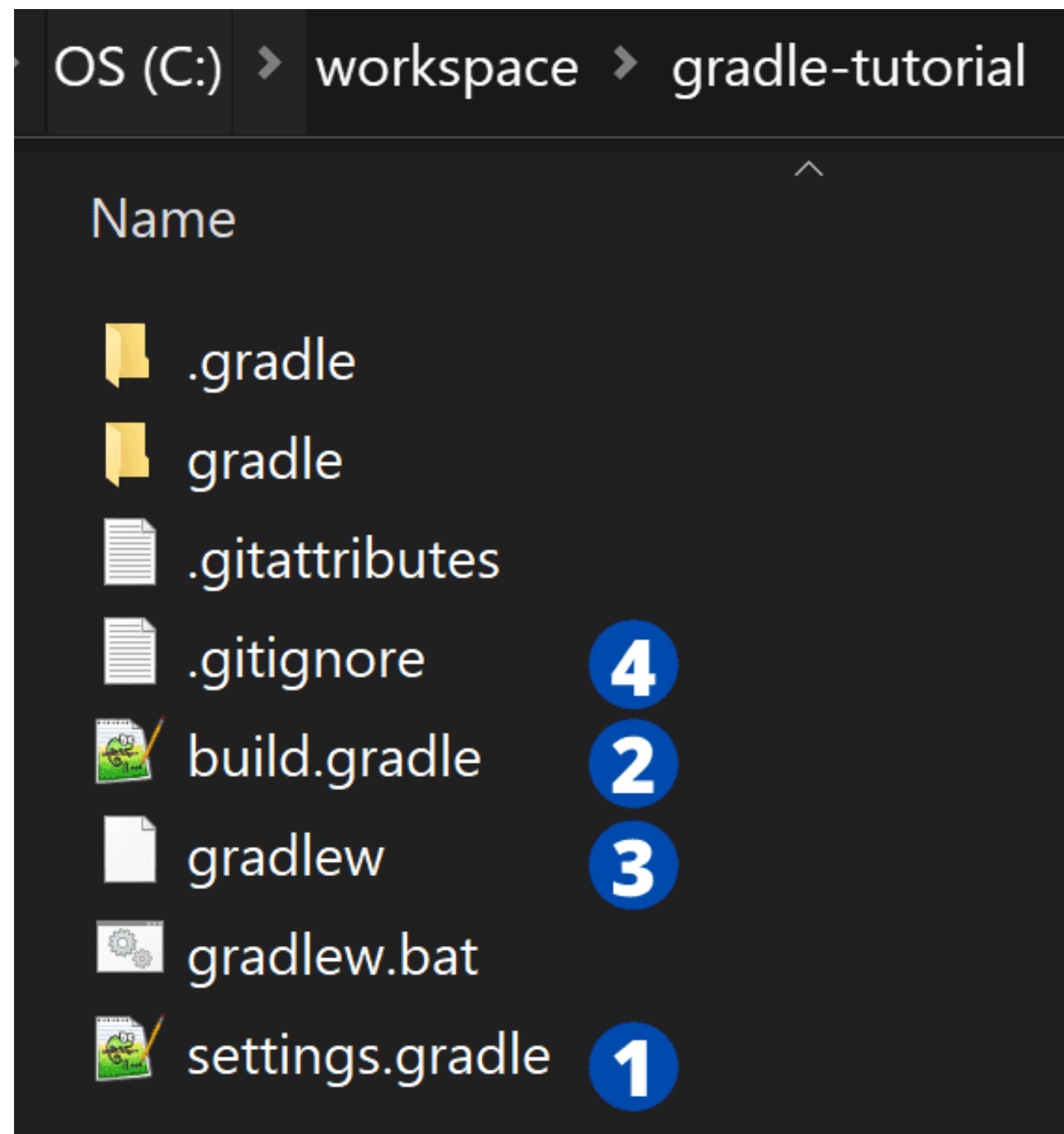
# Getting started with gradle

1

run gradle init

- settings.gradle

sets up some high-level configuration for the project, in our case the project name.



```
/*
 * This file was generated by the Gradle 'init' task.
 *
 * The settings file is used to specify which projects to include in
 *
 * Detailed information about configuring a multi-project build in G
 * in the user manual at https://docs.gradle.org/7.2/userguide/multi
 */

rootProject.name = 'gradle-tutorial'
```

# Getting started with gradle

- build.gradle
- plugins: we can apply the java-library plugin to add support for java library.
- Repositories: we can declare internal and external repository for resolving dependencies. like Maven, Ant, and Ivy.
- Dependencies: dependencies that are necessary for a particular subject.

```
build.gradle  ✕  
1 /*  
2  * This file was generated by the Gradle 'init' task.  
3  *  
4  * This generated file contains a sample Java Library project to get you started.  
5  * For more details take a look at the Java Libraries chapter in the Gradle  
6  * User Manual available at https://docs.gradle.org/6.0.1/userguide/java\_library\_plugin.html  
7  */  
8  
9 plugins {  
10     // Apply the java-library plugin to add support for Java Library  
11     id 'java-library'  
12 }  
13  
14 repositories {  
15     // Use jcenter for resolving dependencies.  
16     // You can declare any Maven/Ivy/file repository here.  
17     jcenter()  
18 }  
19  
20 dependencies {  
21     // This dependency is exported to consumers, that is to say found on their compile classpath.  
22     api 'org.apache.commons:commons-math3:3.6.1'  
23  
24     // This dependency is used internally, and not exposed to consumers on their own compile classpath.  
25     implementation 'com.google.guava:guava:28.0-jre'  
26  
27     // Use JUnit test framework  
28     testImplementation 'junit:junit:4.12'  
29 }  
30
```



# Getting started with gradle

- `gradlew` and `gradlew.bat`

**they are known as the Gradle wrapper scripts, for Linux/Mac and Windows respectively. These let you build an application without having to download and install Gradle.**

# Getting started with gradle

- .gitignore

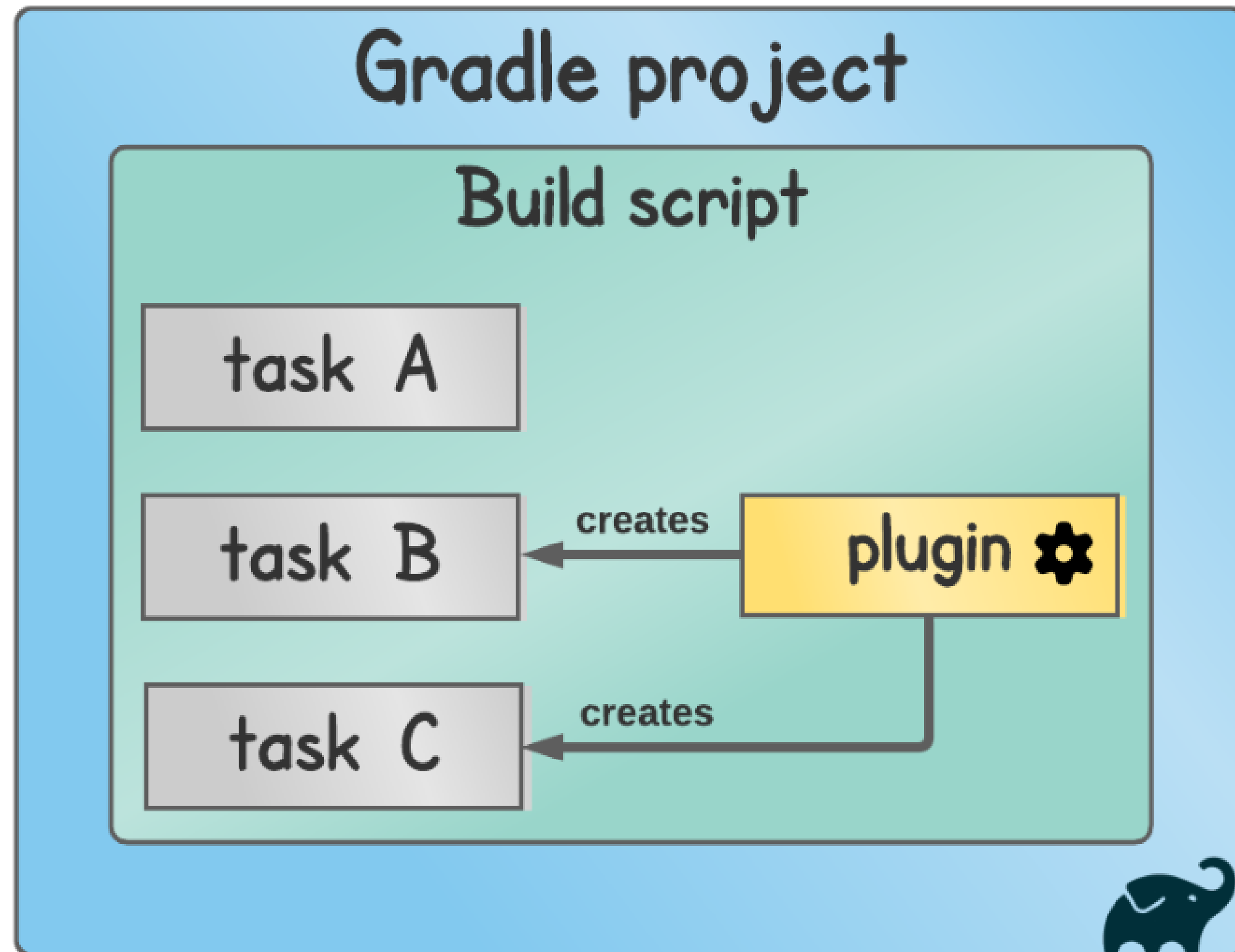
configures Git so that the .gradle and build directories aren't committed into version control. Everything else gets committed though.

```
# Ignore Gradle project-specific cache directory
.gradle

# Ignore Gradle build output directory
build
```



# gradle's components





# Gradle vs Maven

## Gradle

### → Langage de configuration

utilise un langage de script basé sur Groovy pour configurer les projets. Cela permet une configuration plus flexible et expressive.



## Maven

utilise XML pour configurer les projets. La configuration est basée sur des fichiers POM (Project Object Model) XML.

### → Flexibilité et Expressivité

offre une grande flexibilité et expressivité grâce à son langage de script. Il permet de personnaliser plus facilement le processus de construction.

offre une configuration plus rigide et structurée, ce qui peut être limitant dans certaines situations.

# Gradle vs Maven

## Gradle



### Performance

conçu pour offrir de meilleures performances, car il ne suit que les tâches modifiées, ce qui est avantageux pour les projets de grande envergure



### Support des Langages

prend en charge une variété de langages de programmation, dont Java, C, C++, Groovy, et bien d'autres. Il est plus polyvalent en termes de langages pris en charge.

## Maven

peut être moins performant pour les gros projets, car il ne maintient pas de cache entre les builds.

principalement orienté vers Java, bien qu'il prenne en charge d'autres langages. Il est fortement associé à l'écosystème Java.

# Gradle vs Maven

**Gradle**

**Maven**

→ **Communauté et Support**

bénéficie également d'une communauté active, bien que plus récente, et d'une documentation de qualité.

a une communauté bien établie et une documentation abondante en raison de sa longue histoire.

→ **Facilité d'apprentissage**

peut nécessiter un peu plus de temps pour maîtriser en raison de sa flexibilité et de son langage de script.

est généralement considéré comme plus facile à apprendre pour les débutants en raison de sa configuration basée sur XML.



# Installation

- **Téléchargez Gradle : Visitez le site officiel de Gradle à l'adresse: <https://gradle.org/install/> pour télécharger la dernière version de Gradle au format ZIP.**
- **Extrayez l'archive ZIP : Une fois le téléchargement terminé, extrayez le contenu de l'archive ZIP vers un répertoire de votre choix. Par exemple, vous pouvez extraire Gradle vers C:\gradle.**

**Configurez la variable d'environnement GRADLE\_HOME :**

- **Cliquez avec le bouton droit sur "Ce PC" (ou "Poste de travail") et sélectionnez "Propriétés".**
- **Cliquez sur "Paramètres système avancés" (ou "Paramètres système").**
- **Cliquez sur le bouton "Variables d'environnement".**



# Installation

- Dans la section "Variables système", cliquez sur "Nouveau" et ajoutez une variable avec le nom `GRADLE_HOME` et la valeur du chemin vers le dossier Gradle que vous avez extrait.
- Mettez à jour la variable `PATH` :
  - a. Dans la section "Variables système", trouvez la variable `Path`, sélectionnez-la, puis cliquez sur "Modifier".
  - b. Ajoutez `%GRADLE_HOME%\bin` à la fin de la liste des chemins.
  - c. Vérifiez l'installation : Ouvrez un invite de commandes (cmd) et exécutez la commande `gradle -v` pour vérifier que Gradle a été installé correctement.