



# 1. Guide d'Installation de Java

## 1.1 Télécharger le JDK

- Allez sur le site officiel d'Oracle :  
<https://www.oracle.com/java/technologies/javase-downloads.html>
- Choisissez la version du JDK que vous souhaitez installer (par exemple, JDK 17 ou JDK 20).
- Sélectionnez votre système d'exploitation (Windows, macOS, Linux).
- Téléchargez le fichier d'installation approprié (par exemple, .exe pour Windows, .tar.gz pour Linux).

## 1.2 Installer Java sur Windows

### 1.2.1 Exécuter l'installateur

- Une fois téléchargé, double-cliquez sur le fichier d'installation.
- Suivez les instructions de l'installateur, acceptez le contrat de licence et choisissez l'emplacement d'installation (le répertoire par défaut est recommandé).

### 1.2.2 Configurer les variables d'environnement

Définir JAVA\_HOME :

1. Allez dans Panneau de configuration → Système → Paramètres système avancés.
2. Cliquez sur Variables d'environnement.
3. Sous Variables système, cliquez sur Nouveau.

4. Dans Nom de la variable, tapez JAVA\_HOME.
5. Dans Valeur de la variable, entrez le chemin d'installation du JDK (par exemple, C:\Program Files\Java\jdk-17).
6. Cliquez sur OK.

#### Ajouter Java au PATH :

1. Sous Variables système, sélectionnez la variable Path et cliquez sur Modifier.
2. Cliquez sur Nouveau et ajoutez le chemin vers le répertoire bin de votre installation JDK (par exemple, C:\Program Files\Java\jdk-17\bin).
3. Cliquez sur OK.

### 1.2.3 Vérifier l'installation

Ouvrez l'Invite de commande et exécutez :

```
1 java -version
2 javac -version
```

Ces commandes doivent afficher la version de Java installée et la version du compilateur, respectivement.

## 1.3 Installer Java sur macOS

### 1.3.1 Utiliser Homebrew (Recommandé)

Si vous avez installé Homebrew, vous pouvez installer le JDK en utilisant :

```
1 brew install openjdk@17
```

### 1.3.2 Définir JAVA\_HOME :

Ouvrez ~/.bash\_profile ou ~/.zshrc et ajoutez les lignes suivantes :

```
1 export JAVA_HOME=$(/usr/libexec/java_home)
2 export PATH=$JAVA_HOME/bin:$PATH
```

Ensuite, exécutez :

```
1 source ~/.bash_profile
2 source ~/.zshrc
```

### 1.3.3 Vérifier l'installation

Vérifiez la version de Java avec :

```
1 java -version
2 javac -version
```

## 1.4 Installer Java sur Linux (Ubuntu/Debian)

### 1.4.1 Utiliser APT

Installez le paquet OpenJDK avec la commande suivante :

```
1 sudo apt update
2 sudo apt install openjdk-17-jdk
```

### 1.4.2 Définir JAVA\_HOME :

Ouvrez ~/.bashrc ou ~/.zshrc et ajoutez :

```
1 export JAVA_HOME=$(dirname $(dirname $(readlink -f $(which
   java))))
2 export PATH=$JAVA_HOME/bin:$PATH
```

Puis exécutez :

```
1 source ~/.bashrc
```

### 1.4.3 Vérifier l'installation

Vérifiez la version de Java avec :

```
1 java -version
2 javac -version
```

## 1.5 Générer la documentation Java avec javadoc

### 1.5.1 Exemple de code avec des commentaires JavaDoc

Voici un exemple d'un programme Java simple avec des commentaires JavaDoc :

```
1 /**
2  * Classe principale de l'application.
3  */
4 public class HelloWorld {
5     /**
6      * Méthode principale de l'application.
7      *
8      * @param args Les arguments de la ligne de commande
9      */
10    public static void main(String[] args) {
11        System.out.println("Hello, World!");
12    }
13 }
```

### 1.5.2 Générer la documentation

Utilisez la commande suivante pour générer la documentation HTML :

```
1 javadoc -d docs HelloWorld.java
```

Cela va créer la documentation dans le répertoire docs.

## 1.6 Désinstaller Java

### 1.6.1 Sur Windows

Allez dans Panneau de configuration → Programmes et fonctionnalités, trouvez Java Development Kit, et cliquez sur Désinstaller.

### 1.6.2 Sur macOS

Exécutez la commande suivante :

```
1 sudo rm -rf /Library/Java/JavaVirtualMachines/jdk-17.jdk
```

### 1.6.3 Sur Linux

Désinstallez avec APT :

```
1 sudo apt remove openjdk-17-jdk
```

## 1.7 Exemple de programme Java

Voici un programme Java simple que vous pouvez utiliser pour tester votre installation :

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hello, World!");  
4     }  
5 }
```

### 1.7.1 Compilation et exécution

1. Sauvegardez le fichier sous le nom HelloWorld.java.
2. Compilez le programme avec :

```
1 javac HelloWorld.java  
2
```

3. Exécutez le programme avec :

```
1 java HelloWorld  
2
```

## 1.8 Vérification de la configuration

Assurez-vous que `JAVA_HOME` est défini correctement et que le chemin `Path` inclut le répertoire `bin` du JDK. Exécutez :

```
1 echo $JAVA_HOME
```



## 2. Guide d'Installation d'Eclipse

### 2.1 Télécharger et Installer Eclipse

- Rendez-vous sur le site officiel d'Eclipse : <https://www.eclipse.org/downloads/>.
- Cliquez sur le bouton Download pour obtenir l'installateur d'Eclipse.
- Choisissez la version d'Eclipse correspondant à votre système d'exploitation (Windows, macOS, Linux).
- Téléchargez le fichier d'installation approprié (par exemple, .exe pour Windows, .dmg pour macOS ou .tar.gz pour Linux).

### 2.2 Installation d'Eclipse sur Windows

#### 2.2.1 Exécution de l'installateur

- Une fois téléchargé, double-cliquez sur le fichier d'installation eclipse-inst-win64.exe.
- L'installateur d'Eclipse propose plusieurs paquets. Sélectionnez Eclipse IDE for Java Developers.
- Choisissez le chemin d'installation recommandé, puis cliquez sur Install.
- Acceptez les termes du contrat de licence et attendez la fin de l'installation.

#### 2.2.2 Lancer Eclipse

- Une fois l'installation terminée, cliquez sur Launch pour ouvrir Eclipse.
- Sélectionnez l'espace de travail (dossier où seront stockés vos projets Java).
- L'environnement Eclipse s'ouvre et vous êtes prêt à commencer à coder en Java.

## 2.3 Installation d'Eclipse sur macOS

### 2.3.1 Installation

- Téléchargez le fichier `.dmg` pour macOS depuis le site d'Eclipse.
- Ouvrez le fichier `.dmg` et faites glisser l'icône Eclipse dans le dossier Applications.
- Allez dans le dossier Applications et double-cliquez sur l'icône Eclipse pour lancer l'application.

### 2.3.2 Configurer Eclipse

- Choisissez un espace de travail lorsque vous lancez Eclipse.
- Eclipse s'ouvrira avec l'interface principale, prête pour le développement Java.

## 2.4 Installation d'Eclipse sur Linux

### 2.4.1 Télécharger et Installer

- Téléchargez le fichier `.tar.gz` pour Linux.
- Ouvrez un terminal et extrayez le fichier téléchargé avec la commande suivante :

```
1 tar -xvzf eclipse-inst-linux64.tar.gz
```

- Naviguez dans le dossier extrait et lancez l'installateur avec la commande suivante :

```
1 ./eclipse-inst
```

- Suivez les instructions à l'écran pour installer Eclipse (choisissez Eclipse IDE for Java Developers).

### 2.4.2 Lancer Eclipse

- Une fois l'installation terminée, vous pouvez lancer Eclipse depuis le terminal en tapant `eclipse`.
- Sélectionnez un espace de travail lorsque vous y êtes invité, puis commencez à coder en Java.

## 2.5 Créer un Premier Projet Java dans Eclipse

### 2.5.1 Étapes pour créer un projet Java

1. Lancez Eclipse et sélectionnez un espace de travail.
2. Dans le menu principal, cliquez sur `File` → `New` → `Java Project`.
3. Donnez un nom à votre projet, par exemple `MonPremierProjet`.
4. Cliquez sur `Finish` pour créer le projet.



### 2.5.2 Créer une classe Java

1. Dans le panneau de gauche, faites un clic droit sur le dossier src de votre projet, puis sélectionnez New → Class.
2. Donnez un nom à votre classe, par exemple HelloWorld, et cochez l'option public static void main(String[] args) pour ajouter la méthode principale.
3. Cliquez sur Finish.

### 2.5.3 Exemple de code Java dans Eclipse

```
1  /**
2   * Classe principale de l'application.
3   */
4  public class HelloWorld {
5      /**
6       * Méthode principale de l'application.
7       *
8       * @param args Les arguments de la ligne de commande
9       */
10     public static void main(String[] args) {
11         System.out.println("Hello, World!");
12     }
13 }
```

### 2.5.4 Exécuter le programme

1. Faites un clic droit sur la classe HelloWorld.java dans l'Explorateur de projet.
2. Sélectionnez Run As → Java Application.
3. Le programme sera exécuté et la sortie Hello, World! apparaîtra dans la console.



### 3. TP 1 : Introduction à Java

## Objectifs

- Comprendre les types de données en Java.
- Manipuler les fonctions d’affichage (print et println).
- Découvrir les structures de contrôle (if/else, switch).
- Utiliser les boucles (for, while).

## Partie 1 : Questions guidées avec réponses

### 1.1 Types de données et affichage

#### Question 1 :

Comment déclarer une variable de type entier (int), et afficher sa valeur dans la console ?

**Réponse :**

```
1 public class Main {
2     public static void main(String[] args) {
3         int nombre = 42;
4         System.out.println("La valeur de nombre est : " +
5             nombre);
6     }
7 }
```

**Question 2 :**

Déclarez une variable de type double et affichez-la avec une description.

**Réponse :**

```
1 public class Main {
2     public static void main(String[] args) {
3         double pi = 3.14159;
4         System.out.println("La valeur de pi est : " + pi);
5     }
6 }
```

**Question 3 :**

Comment concaténer du texte avec des variables pour les afficher ?

**Réponse :**

```
1 public class Main {
2     public static void main(String[] args) {
3         String prenom = "Amina";
4         int age = 25;
5         System.out.println("Bonjour, je m'appelle " + prenom +
6             " et j'ai " + age + " ans.");
7     }
8 }
```

## 1.2 Structures de contrôle (if/else, switch)

**Question 4 :**

Comment utiliser une condition simple avec if pour vérifier si un nombre est positif ou négatif ?

**Réponse :**

```
1 public class Main {
2     public static void main(String[] args) {
3         int nombre = -5;
4         if (nombre > 0) {
5             System.out.println("Le nombre est positif.");
6         } else {
7             System.out.println("Le nombre est négatif.");
8         }
9     }
10 }
```

**Question 5 :**

Utilisez une structure switch pour afficher un message selon une note (A, B, C, D, F).

**Réponse :**

```
1 public class Main {
```

```
2 public static void main(String[] args) {
3     char grade = 'B';
4     switch (grade) {
5         case 'A':
6             System.out.println("Excellent !");
7             break;
8         case 'B':
9             System.out.println("Très bien !");
10            break;
11         case 'C':
12             System.out.println("Bien.");
13             break;
14         case 'D':
15             System.out.println("Passable.");
16             break;
17         case 'F':
18             System.out.println("Échec.");
19             break;
20         default:
21             System.out.println("Note inconnue.");
22             break;
23     }
24 }
25 }
```

### 1.3 Boucles (for, while)

#### Question 6 :

Comment afficher les nombres de 1 à 5 avec une boucle for ?

#### Réponse :

```
1 public class Main {
2     public static void main(String[] args) {
3         for (int i = 1; i <= 5; i++) {
4             System.out.println(i);
5         }
6     }
7 }
```

#### Question 7 :

Utilisez une boucle while pour afficher les nombres de 10 à 1 en ordre décroissant.

#### Réponse :

```
1 public class Main {
2     public static void main(String[] args) {
3         int i = 10;
4         while (i > 0) {
5             System.out.println(i);
```

```
6         i--;  
7     }  
8 }  
9 }
```

## Partie 2 : Exercices sans réponses

### 2.1 Types de données et affichage

#### Exercice 1 :

Déclarez une variable `float` pour stocker une température, puis affichez-la avec un message expliquant s'il fait chaud ou froid (si la température est supérieure à 25°C, il fait chaud).

#### Exercice 2 :

Déclarez une variable `boolean` pour indiquer si une porte est ouverte (`true`) ou fermée (`false`), puis affichez le message correspondant.

### 2.2 Structures de contrôle

#### Exercice 3 :

Écrivez un programme qui demande à l'utilisateur d'entrer un nombre, et affiche un message si le nombre est pair ou impair.

#### Exercice 4 :

Utilisez une structure `switch` pour afficher un message en fonction du jour de la semaine (Lundi, Mardi, Mercredi, etc.).

### 2.3 Boucles

#### Exercice 5 :

Écrivez un programme qui affiche la table de multiplication de 5 (de 1 à 10).

#### Exercice 6 :

Utilisez une boucle `while` pour calculer et afficher la somme des nombres de 1 à 50.

#### Exercice 7 :

Créez une boucle qui demande à l'utilisateur d'entrer un nombre, et continue jusqu'à ce que l'utilisateur entre un nombre négatif.