



# Tutoriel 2 - compiler et exécuter un programme Java

☰ Category

Envoi 1

## Pré-requis

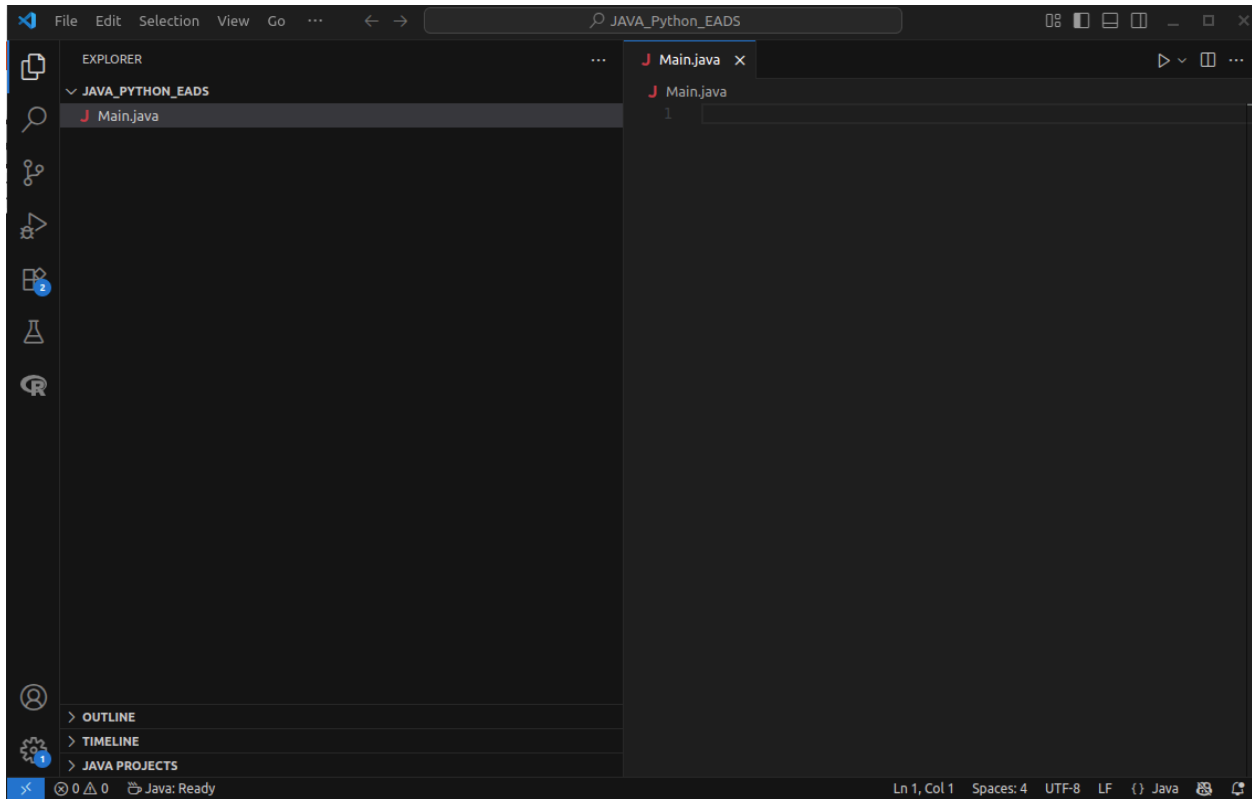
- Avoir installé Java (voir tutoriel)
- Avoir installé VS Code et les extensions nécessaires (voir tutoriel)

## Créez votre premier fichier .java

Créez un dossier dédié à vos travaux Java.

Ouvrez VS Code, et dans le menu (tout en haut) cliquez sur *File > Open Folder*. Sélectionnez votre dossier de travail.

Puis cliquez sur *File > New File*. Dans la pop-up qui s'ouvre, tapez *HelloWorld.java*, puis entrez. Cela crée un nouveau fichier vide. Vous devriez le voir apparaître dans l'onglet "Explorer", qui est dédié à afficher l'arborescence des fichiers dans votre dossier de travail. Le fichier devrait aussi s'ouvrir sur la partie droite de VS Code. C'est ici que vous pourrez l'éditer.



Entrez les lignes de code suivantes et enregistrez le fichier.

```
public Class HellWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello world!");  
    }  
}
```

Pour l'instant il faudra accepter et reproduire cette syntaxe, avant qu'elle soit expliquée. Seuls changent le nom de la classe, qui doit être identique à celle du fichier, et le corps de la fonction, ici la ligne `System.out.println("Hello world!");`.

Survolez le "println" de cette commande pour voir apparaître des indications sur cette fonction Java.

Grâce au menu, vous pouvez ouvrir un Terminal directement positionné dans votre dossier de travail. Affichez les fichiers présents dans ce terminal avec la

commande `ls` . Vous devriez voir votre fichier `HelloWorld.java` . Inspectez-le avec la commande `head HelloWorld.java` .

```
$ ls
# Affiche HelloWorld.java

head HelloWorld.java
# Affiche
# public Class HelloWorld {
#   public static void main(String[] args) {
#     System.out.println("Hello world!");
#   }
# }
```

Nous allons maintenant compiler notre programme java.

Pour cela, tapez la commande `javac HelloWorld.java` . Constatez avec un `ls` que cette commande crée un fichier `HelloWorld.class`. Essayez d'inspecter le fichier avec la commande `head HelloWorld.class` , et constatez que ce fichier n'est pas lisible par un humain. Il s'agit de **bytecode**, c'est-à-dire une version compilée du programme que la **machine virtuelle Java (JVM)** peut comprendre et exécuter. Le code source que vous avez écrit en Java ( `HelloWorld.java` ) est donc traduit par le compilateur ( `javac` ) en un langage intermédiaire ( `HelloWorld.class` ) , indépendant du système d'exploitation.

```
# Compiler le programme.
javac HelloWorld.java

# Vérifier que le fichier .class (bytecode) a été créé
ls
# Affiche HelloWorld.class HelloWorld.java

# Regarder à quoi ressemble le bytecode (non lisible par l'humain)
$ head HelloWorld.class
# Affiche
```

```
#     =  
# java/lang/Object<init>()V  
# java/lang/SystemoutLjava/io/PrintStream;  
#  
# Hello world!  
# java/io/PrintStreamprintln(Ljava/lang/String;)VMainCodeLineNumberTablem  
ain([Ljava/lang/String;)V  
#   urceFile Main.java!*   % 
```

Exécutons maintenant notre programme compilé !

```
java HelloWorld
```

La JVM interprète le bytecode et exécute les instructions qu'il contient, ce qui affiche le message :

Hello world!

Depuis VS Code, vous pouvez facilement compiler et exécuter votre code en cliquant sur le triangle en haut à droite, ou avec le raccourci clavier : **Ctrl + Alt + n** .

Attention, pensez à bien sauvegarder vos modifications ( **Ctrl + S** ) avant de recompiler/exécuter.

Depuis Java SE 11, il est également possible de compiler et d'exécuter en une seule commande. Cependant aucun fichier `.class` n'est créé (seulement mis en cache dans la mémoire):

```
java HelloWorld.java
```