



L'ÉCOLE DES EXPERTS
MÉTIER DE
L'INFORMATIQUE

2JVA

Java Standard Edition



L'ÉCOLE DES EXPERTS
MÉTIER DE
L'INFORMATIQUE

Introduction et concept Java

Introduction.

Histoire de Java

Histoire

1. Le langage Java a été initialement développé dans les années 90 par la société Sun Microsystems.
2. Environnement Indépendant du hardware.
3. Permet d'être exécuter sur divers appareils.





Avantages et caractéristiques

Robuste et sûr

Pas de pointeurs, gestion facile des exceptions.

Gestion de la mémoire

Le Garbage Collector gère la mémoire facilement et de manière transparente.

Facilité

Programmation orientée objet.

Syntaxe proche et inspirée du Language c.

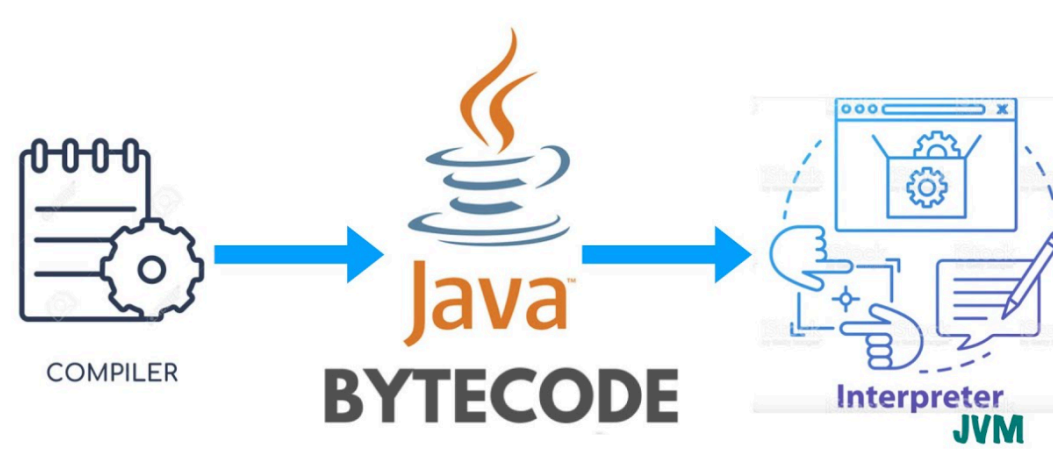
Avantages et caractéristiques

Indépendance de la plateforme

Le compilateur Java génère du bytecode.

Le bytecode est un ensemble d'instructions pour Java Virtual Machine (JVM).

Java Virtual Machine (JVM) est disponible sur la majorité des plateformes (Mac, Windows, Linux etc.).





Différentes éditions de Java

Java standard Edition (JSE)

1. Contient les API nécessaires pour développer des applications de bureau.
2. Connectivité réseau (Sockets).
3. Création d'interface utilisateur graphique (Swing).
4. Connectivité à la base de données (JDBC).
5. Gestion des entrées/sorties (API IO).



Différentes éditions de Java

Java Enterprise Edition (JEE)

1. Contient des API pour développer facilement des applications WEB de qualité

- API servlet et JSP
- EJB et JPA
- JSF
- Prise en charge des services Web
- Outils de déploiement



Différentes éditions de Java

Java Micro Edition (Java ME)

1. Contient des API pour développer des applications pour les appareils mobiles
2. Contient un sous-ensemble de packages Java SE
3. Contient des packages spécifiques à la micro édition

4. Attention ! Java ME \neq Android !

Java Micro Edition





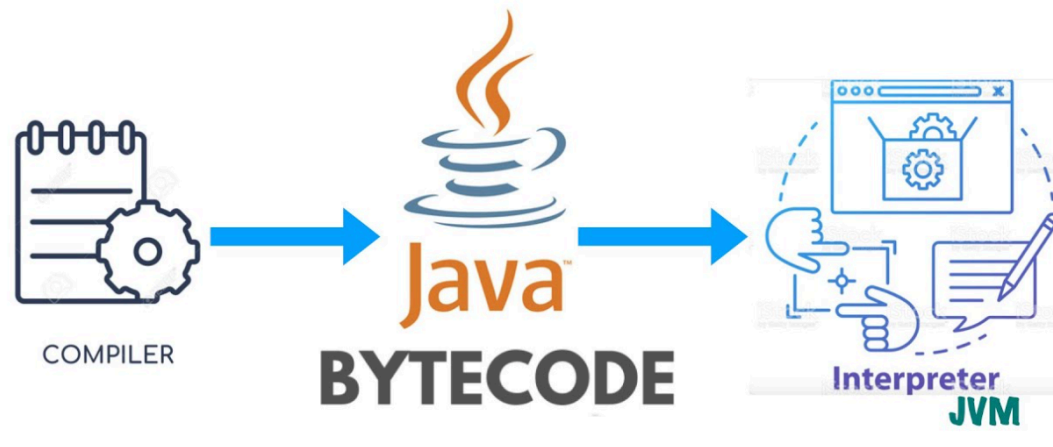
L'ÉCOLE DES EXPERTS
MÉTIER DE
L'INFORMATIQUE

Concepts avancés

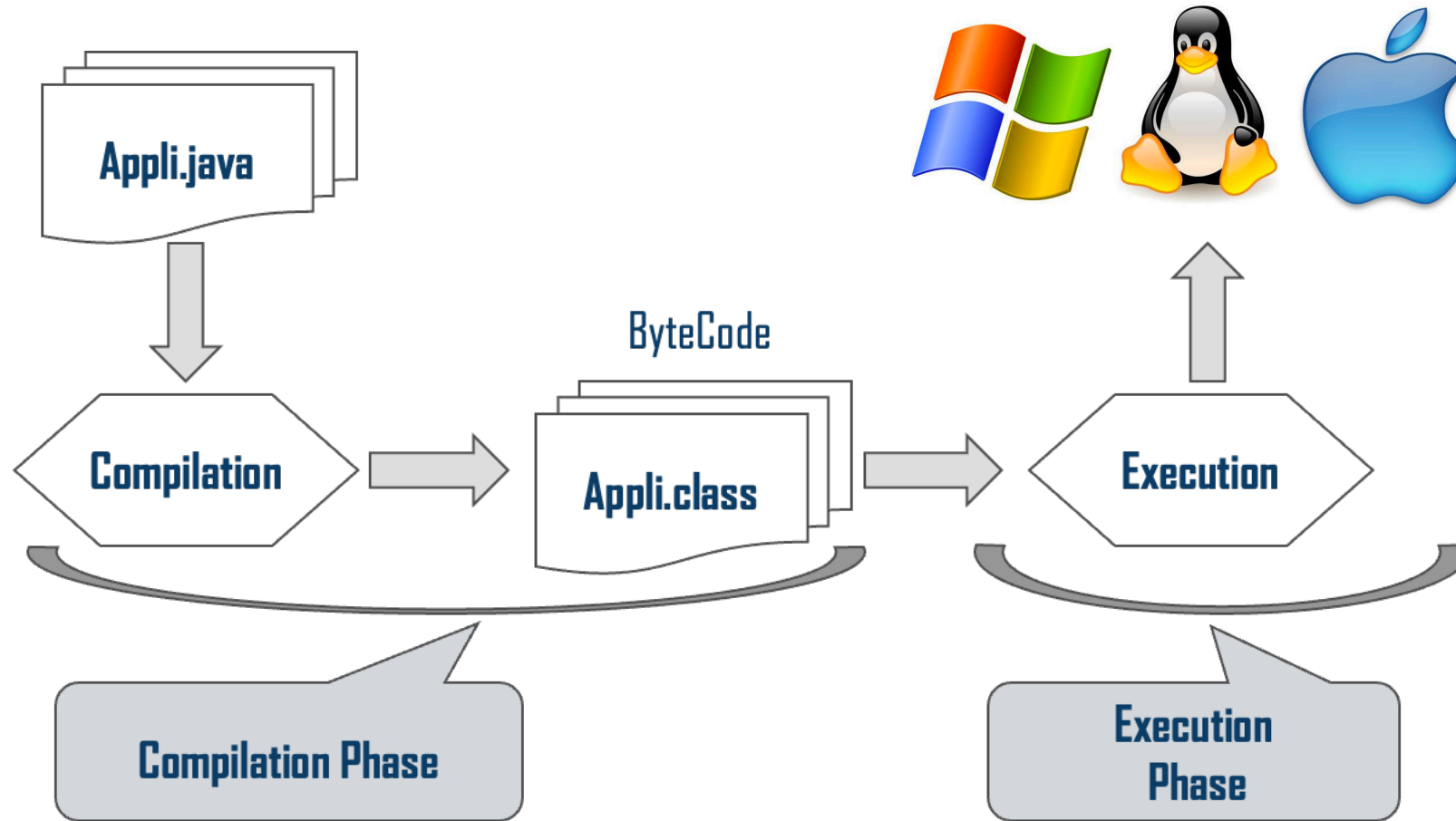
Java Virtual Machine (JVM)

Définition

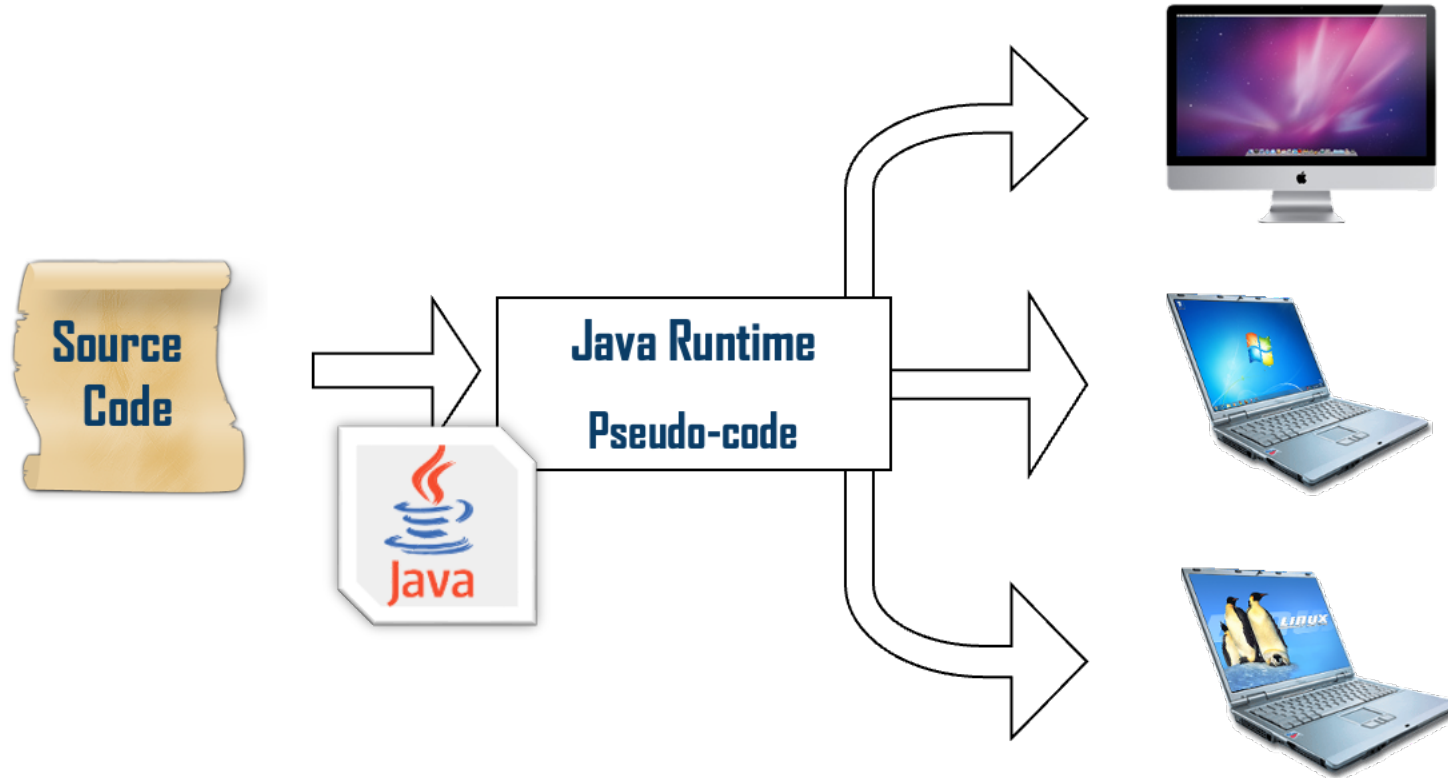
1. La Java Virtual Machine est l'environnement qui sert à l'exécution du code Java compilé en Byte Codes.



Java Virtual Machine (JVM)



Compilation et exécution





JVM, JRE, JDK

JVM

La Java Virtual Machine est l'environnement qui sert à l'exécution du code Java compilé en Byte Codes.

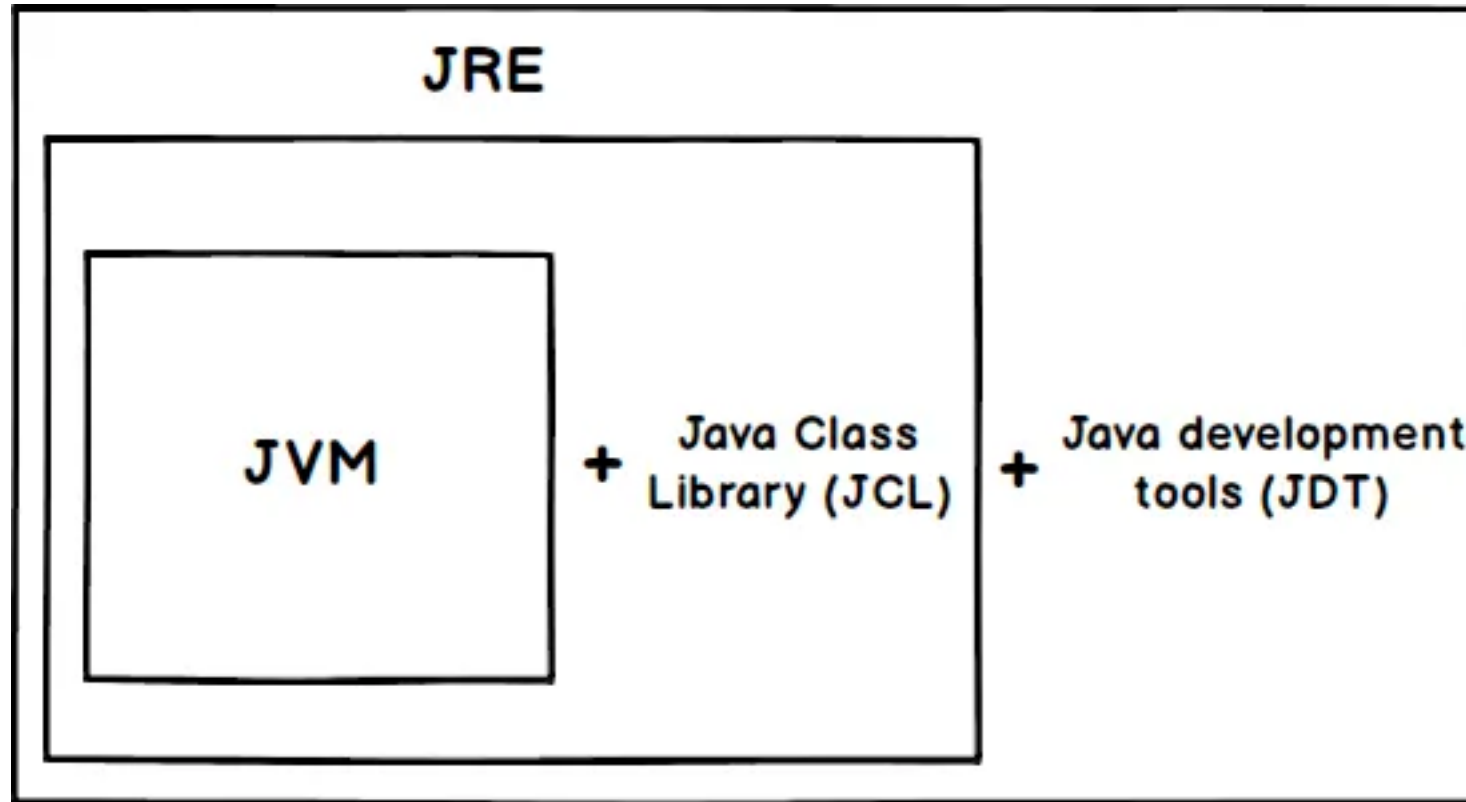
JRE

Environnement nécessaire pour utiliser les applications Java. Il contient un ensemble de bibliothèques et d'autres fichiers que la JVM utilise lors de l'exécution.

JDK

Le kit de développement Java (JDK) est un environnement de développement logiciel permettant de développer des applications et des applets Java. Il contient le JRE + la JVM.

JVM, JRE, JDK



$\text{JRE} = \text{JVM} + \text{Java Class Library (JCL)}$

$\text{JDK} = \text{JRE} + \text{Java Development Tools (JDT)}$



L'ÉCOLE DES EXPERTS
MÉTIER DE
L'INFORMATIQUE

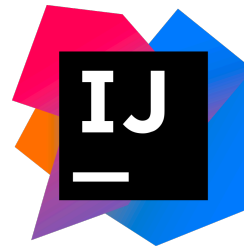
IDE

Integrated Development Environments

Vous pouvez développer avec un simple éditeur de texte et la commande **javac** si vous le souhaitez.

Mais des solutions plus sophistiquées existent : **Environnements de développement intégrés (IDE).**

Il existe plusieurs IDE pour coder en Java comme (Eclipse, NetBeans etc.), mais dans ce cours nous allons nous pencher essentiellement sur l'IDE IntelliJIdea.





IDE

Integrated Development Environments

IntelliJ IDEA est disponible en deux éditions

Community Edition

- Gratuit et Open Source
- Prise en charge de Java SE, Kotlin, Scala, Groovy, Python etc.

Ultimate Edition

- Licence commerciale
- Prise en charge de Java EE, Framework Spring, Ruby, PHP, Go, etc.



IDE

Integrated Development Environments

Les spécifications de IntelliJ IDEA

1. Edité par JetBrains
2. Développé avec Java avec la librairie UI : **SWING**
3. Beaucoup de plugins disponible
4. Multi-Platform

License IntelliJ IDEA Ultimate

Les licences IntelliJ IDEA Ultimate sont disponibles avec nos adresses email SUPINFO :

- <https://www.jetbrains.com/fr-fr/community/education/#students>



L'ÉCOLE DES EXPERTS
MÉTIER DE
L'INFORMATIQUE

Compilation et exécution



Introduction et concept

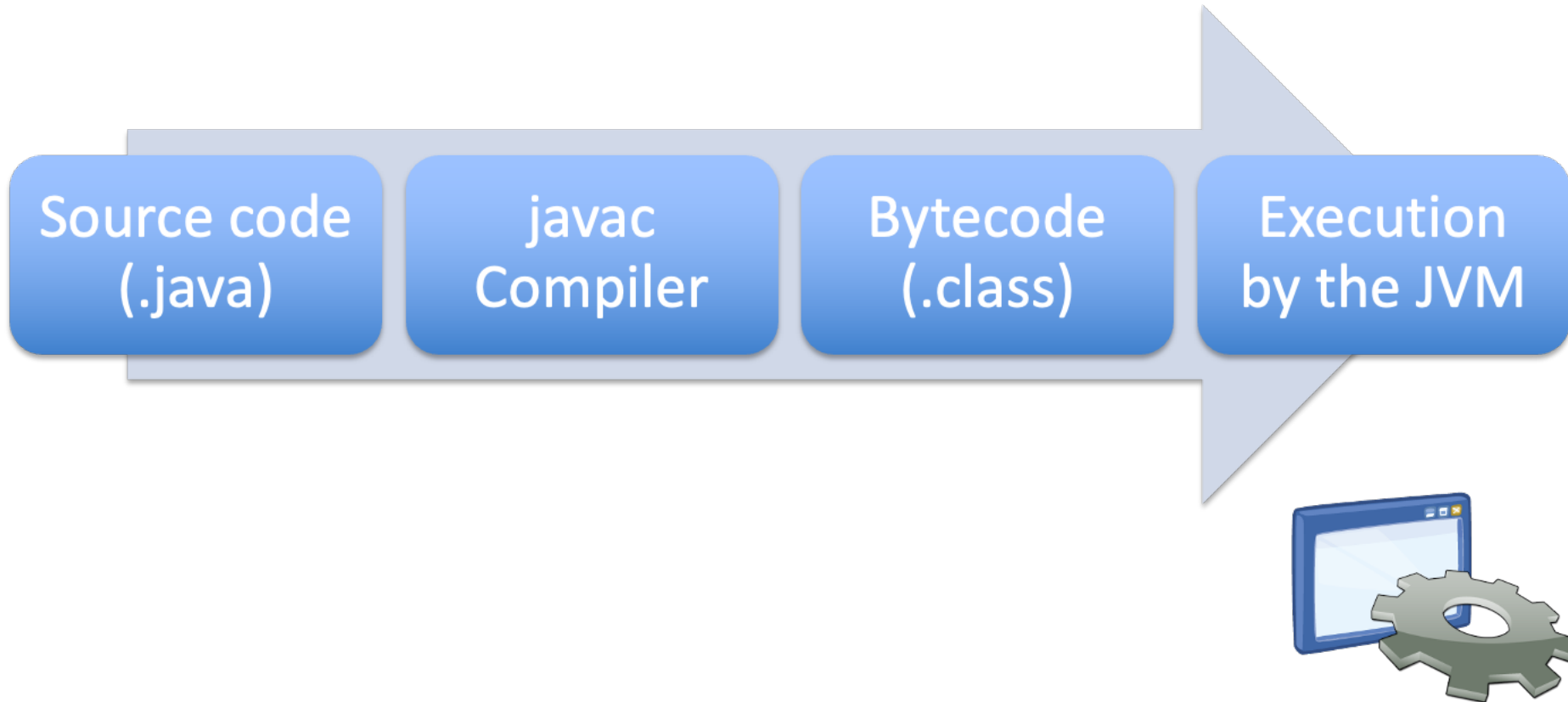
Les fichiers source sont représentés avec l'extension **.java** .

La compilation produit des fichiers compilés avec l'extension **.class** .

Les fichiers de class contiennent le byte code.

Les fichiers de class sont exécutés par la JVM.

Introduction et concept



Entrée clavier

- La classe `Scanner` est utilisée pour récupérer le texte entré d'un utilisateur, et de le donner au programme. (Nous verrons ça plus en détails plus tard)

```
Scanner in = new Scanner(System.in);
```

- **Exemple**

```
// Créer un objet Scanner
Scanner input = new Scanner(System.in);

System.out.print("Entrer votre prénom: ");

// Récupérer le texte du clavier
String name = input.nextLine();

// Afficher le texte dans la console
System.out.println("Je m'appelle " + name);

// Fermer le scanner
input.close();
```



Premier programme Java

Méthode Main

La méthode main est le point d'entrée d'une application Java.

Cette méthode Main est la première méthode appelée au démarrage de l'application.



Exercice : Premier programme Java

Méthode Main

La méthode main est le point d'entrée d'une application Java.
Cette méthode Main est la première méthode appelée au démarrage de l'application.

Exécuter le programme

1. Lancer IntelliJ IDEA
2. Créer un nouveau projet Java
3. Cliquer sur le symbole « play » situé en haut à droite du programme pour l'exécuter.
4. Copier le code pour récupérer les entrées clavier, et les afficher sur la console

Main.java

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