



Introduction

Plan

1 Vue générale

- La technologie Java
- Le langage Java
- La machine virtuelle Java
- Résumé

2 Hello World

- Prérequis
- Premier programme : 3 étapes
- Résumé

3 HelloWorld en détail

- Les commentaires
- Définition d'une classe
- La méthode main
- Le code
- Résumé

La technologie Java

Origine

- Sun microsystems, \approx 1995

► [Java sur Wikipédia](#)

“Java”, c’est à la fois :

- un **langage de programmation** orienté objet
- une **plate-forme** : un environnement pour l’exécution et le développement des programmes Java :
 - une **machine virtuelle** : la JVM (Java Virtual Machine)
 - des **APIs** (Application Programming Interfaces)
 - ...

Le langage Java

Principales caractéristiques

- *Simple, Object Oriented, and Familiar*
- *Robust and Secure*
- *Architecture Neutral and Portable (JVM)*
- *High Performance*
- *Interpreted, Threaded, and Dynamic*

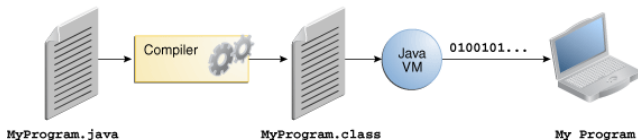
► Design Goals of the Java Programming language

La plate-forme Java

Les implémentations de la plate-forme Java possèdent :

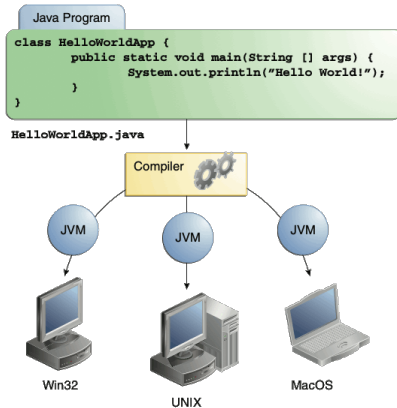
- *Development Tools* : javac , java, javadoc, ...
- *Application Programming Interface (API)*
- *Deployment Technologies*
- *User Interface Toolkits* : JavaFX, Swing, ...
- *Integration Libraries*, API spécialisées : JDBC, JNDI, IDL, ...

Le langage Java



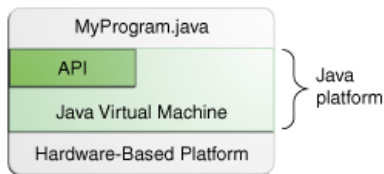
La machine virtuelle Java - JVM

“Write once, run anywhere”

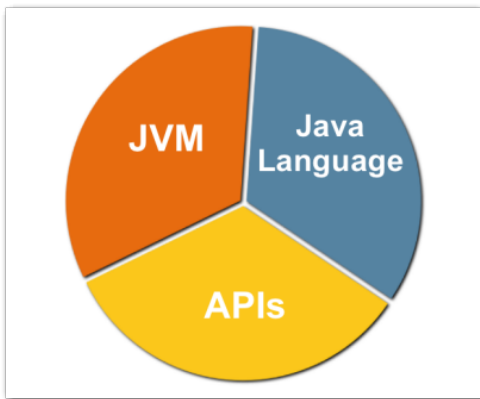


API : Application Programming Interface

L'API Java est formée par une collection de composants logiciels fournissant de nombreuses fonctionnalités (> 4000 classes)



Pour résumer



Prérequis

Pour pouvoir écrire un programme Java il faut au moins :

- The Java SE Development Kit : 8 (1.8) ou 11 / 17 (LTS)
 - Oracle : [▶ téléchargement](#) / [▶ par dépôt ppa](#)
 - OpenJDK : [▶ openjdk.java.net/](#)
- un éditeur de texte

Remarque : Un plugin Java pour navigateur ne suffit pas : il ne contient que le JRE (Java Runtime Environment)

Premier programme : 3 étapes

1. Codage

Création du fichier source **HelloWorldApp.java**

2. Compilation

Compilation du fichier source HelloWorldApp.java en **HelloWorldApp.class**

3. Exécution

Utilisation de la **JVM** locale pour exécuter le fichier HelloWorldApp.class

1. Codage du fichier source HelloWorldApp.java

Par exemple dans /home/joe/ :

HelloWorldApp.java

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
public class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

Java fait la différence majuscule/minuscules

HelloWorldApp \neq helloworldapp

2. Compilation du fichier source en HelloWorldApp.class

Dans /home/joe/, utilisation du programme **javac** (java compiler) :

Dans un terminal

```
joe@iut:~$ javac HelloWorldApp.java
joe@iut:~$ ls
HelloWorldApp.class  HelloWorldApp.java
joe@iut:~$
```

le programme javac doit être dans le PATH de l'OS !

il est aussi bien sûr possible de taper le chemin complet :

```
/usr/lib/jvm/java-11-oracle/bin/javac HelloWorldApp.java
```

2. Compilation du fichier source en HelloWorldApp.class

Pour inclure javac dans le *path* de l'OS au démarrage, il faut modifier la variable d'environnement PATH pour y inclure le répertoire où se trouve javac :

Linux (ajouter dans le fichier `/.bashrc`)

```
PATH=$PATH : JDKInstalDirectory/bin  
export PATH  
(sous Debian/Ubuntu : page Java )
```

Windows

propriétés du poste de travail → avancé → variables d'environnement (nouveau)
PATH= blabla ; c : \JDKInstalDirectory\bin

3. Exécution de HelloWorldApp.class dans la JVM

Dans /home/joe/, le programme **java** lance la JVM sur le fichier .class passé en paramètre (sans l'extension .class) :

Dans un terminal

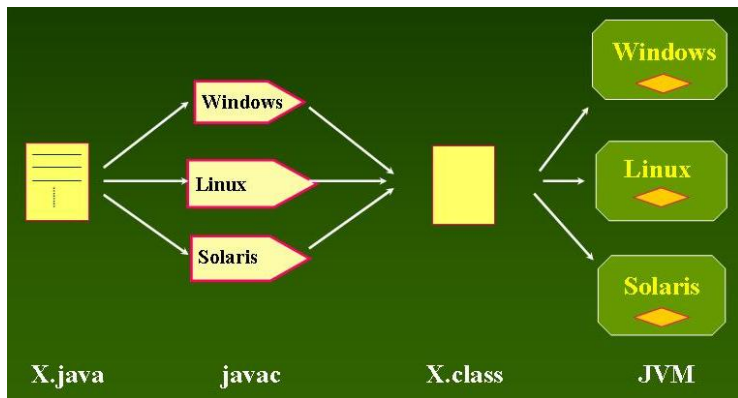
```
joe@iut:~$ java HelloWorldApp
Hello World!
joe@iut:~$
```

le programme java doit lui aussi être dans le PATH de l'OS !

sinon...

```
/usr/lib/jvm/java-11-jdk/bin/java HelloWorldApp
```

Pour résumer



HelloWorldApp.java en détail

Les commentaires

```
/**  
 * The HelloWorldApp class implements an application that  
 * simply prints "Hello World!" to standard output.  
 */
```

Les 3 types de commentaires en Java :

```
/* text */  
/** documentation */  
// text
```

HelloWorldApp.java en détail

La définition de la classe HelloWorldApp

```
public class HelloWorldApp {  
    // code de la classe  
}
```

- la définition d'une classe s'effectue avec le mot clé *class*.
- le mot clé *public* définit la visibilité de la classe.

HelloWorldApp.java en détail

La méthode statique *main*

```
public static void main(String[] args) {  
    System.out.println("Hello World!"); // Display the string.  
}
```

- le mot clé *public* définit la visibilité de la méthode.
- le mot clé *static* signifie que *main* est une méthode de classe.

Une application Java doit contenir au moins une méthode *main*

- C'est cette méthode qui est exécutée par la JVM
- Si elle n'est pas présente :

```
joe@iut:~$ java HelloWorldApp  
Exception in thread "main" java.lang.NoSuchMethodError: main  
joe@iut:~$
```

HelloWorldApp.java en détail

Le code

```
System.out.println("Hello World!"); // Display the string.
```

Utilisation de la classe *System* incluse dans l'API standard pour l'affichage.

Pour résumer

- Il faut au moins définir une classe.
- La méthode main est le point d'entrée du programme.
- l'API fournit en standard de très nombreuses fonctionnalités.

HelloWorldApp.java

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
public class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

Résumé global

- La technologie Java :
 - un langage de programmation orienté objet
 - [The Java language Specification](#)
 - une machine virtuelle : JVM.
 - des APIs, notamment l'API standard
 - [Java Platform Standard Ed. 17 API](#)
- Pour programmer en java, il faut :
 - créer au moins une classe incluant la méthode main.
 - les programmes javac et java pour compiler et exécuter.

Ce cours reprend largement les tutoriaux en ligne proposés par Sun :

▸ [The Java Tutorials](#)