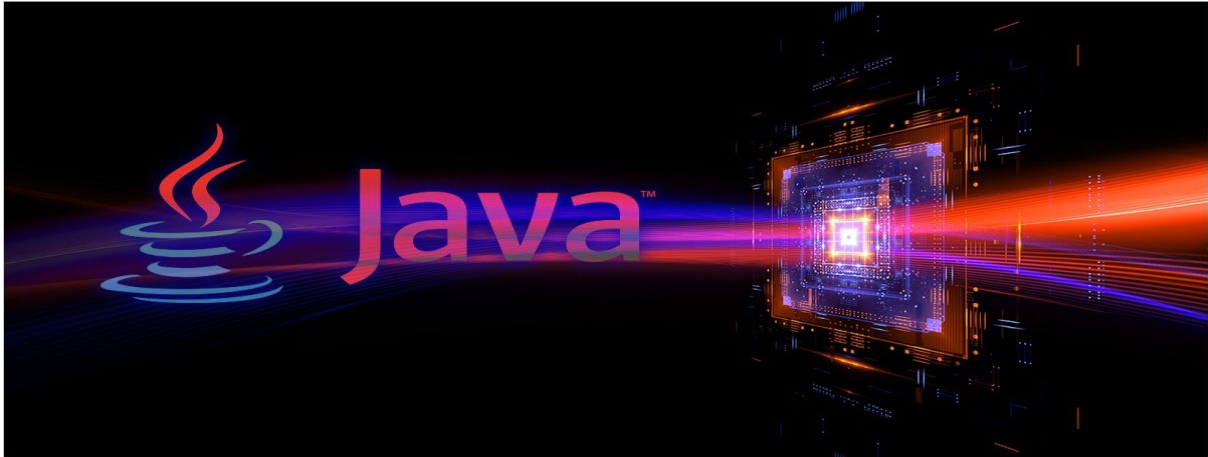


# Tutorials Java Pour Débutants

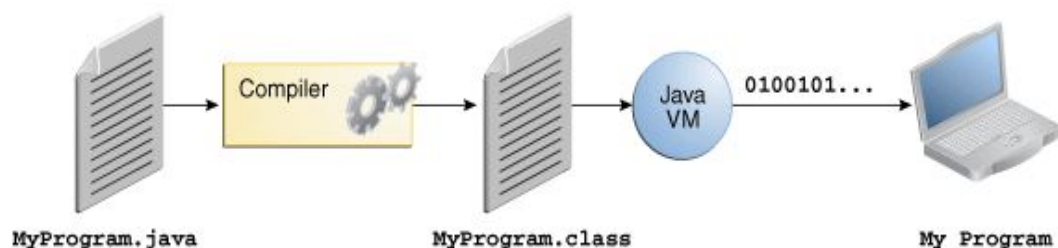


## La technologie JAVA

La technologie Java est à la fois un langage de programmation et une plate-forme.

### Le langage de programmation Java

- ❖ Le langage de programmation Java est un langage de haut niveau qui peut être caractérisé ces mots suivantes:
  - Facile
  - Orienté objet
  - Dynamique
  - Haute performance
  - Sécurisé
- ❖ En travaillant à la langage Java, tout le code source écrit dans des fichiers se terminant par l' `.java` extension.
- ❖ Ces fichiers sont ensuite compilés en des fichiers de l'extension `.class` par le `javac` compilateur.



- ❖ Un `.class` fichier ne contient pas de code natif .

- ❖ Il contient seulement des *bytecodes* - le langage machine de la machine virtuelle Java (Java VM).
- ❖ Grâce à la machine virtuelle Java(JVM), la même application est capable de s'exécuter sur plusieurs plates-formes.

## La plate-forme Java

Une *plate - forme* est l'environnement matériel ou logiciel dans lequel un programme s'exécute.

La plate-forme Java comprend deux composants:

- La *machine virtuelle Java*
- L' *interface de programmation d'application Java* (API)

L'API est une bibliothèque de classes qui a plus de fonctionnalités utiles. sont appelées *packages* .

Il contient tous les objets de base, à la mise en réseau et à la sécurité, à la génération XML et à l'accès aux bases de données, et plus encore.

## Les boucles

L' instruction boucle fournit un moyen pour itérer sur un espace de valeurs . la façon dont elle boucle à plusieurs reprises jusqu'à ce qu'une condition particulière soit satisfaite.

## Syntax generale :

```
for (initialization; termination; increment){
    instructions ..;
}
```

## Les variables :

- La déclaration des variables dans java ce fait comme ceci :
  - <Type de la variable> <Nom de la variable> ;
- les noms des variables ne doivent contenir ni caractère accentués ni espaces et doivent respecter la convention de nommage java.

- Pour les opérations sur les variables on n'a besoin de garder le même type de variable.
- La conversion des types des variables se fait en ajoutant un type devant (int), (double).

## Les Types des variables:

### Primitive types :

**boolean** : True/False

**byte** : num occupe 8 bits ( -128=<byte=<127 )

**short** : num occupe 16 bits ( -32768=<short=<32767 )

**int** : num occupés 32bits ( -2\_147\_483\_648=<int=<2\_147\_483\_647) il faut qu'il soit réel

**long (ex: long my Long = 100L)** : num occupés 64bits

( -9\_223\_372\_036\_854\_775\_808=<long=<9\_223\_372\_036\_854\_775\_807 )

**float** : occupés 32 bits utilisé pour les nombres avec une virgule

**double (float plus précis)**: occupés 64 bits utilisé pour les nombres avec une virgule

**char** : ex : ('A') un Caractère occupés 16 bits -unicode- (ex : char copyright Symbol= '\u00A9' )

**String** : (Normalement c'est Class)

Une chaîne de caractères, Donc il peut contenir plusieurs caractères (techniquement limité par la mémoire ou integer MAX VALUE(2.14B))

## Les Conditions:

L'un des fondamentaux de la programmation c'est les conditions.

Normalement le code JAVA s'exécute d'une façon séquentielle, alors le rôle des conditions ici c'est de gérer plusieurs cas de figure sans pour autant lire tout le code.

## **Les Types des conditions:**

### **If/else :**

```
if(//condition)
{
    //Exécution des instructions si la condition est remplie
}else{
    //Exécution des instructions si la condition n'est pas remplie}
```

### **Switch :**

```
switch (*Variable*/)
{
    case /*Argument*/:
        /*Action*/;
        break;
    default:
        /*Action*/;
```

### **Condition ternaire :**

```
int x = 10, y = 20;
int max = (x < y) ? y : x ;
```

## **Les Collections:**

Les collection c'est un groupe où on peut gérer un ensemble des objets d'un seul type des données.

## **Les Types des Collections:**

## Arrays :

Des tableaux sont utilisés pour stocker plusieurs valeurs dans une seule variable, au lieu de déclarer des variables séparées pour chaque valeur.

```
ex: String[] list= {"Zoro", "Luffy", "Nami", "sanji"};
    int [] numbers = {1, 2, 3, 4};
```

## ArrayList :

La classe **ArrayList** est un tableau redimensionnable, qui se trouve dans le package **java.util**.

parmi ses methodes: add/get/set/clear/remove...

```
ex: ArrayList<String> list= new ArrayList<String>();
```

## HashSet:

Un HashSet est une collection d'éléments où chaqu'un de ces éléments est unique et il se trouve dans le package **java.util**.

parmi ses methodes: add/contain/size/clear/remove...

```
ex: HashSet<String> list= new HashSet<String>();
```

## HashMap :

HashMap stocke les éléments dans des paires

«key/value»(list.put("England", "London");), et vous pouvez y accéder par un index d'un autre type (par exemple, une chaîne). et il se trouve aussi dans le package **java.util**.

parmi ses methodes: put/contain/size/clear/remove...

```
ex: HashMap<String, String> list= new HashMap<String, String>();
```

## Les Fonctions:

Une fonction est un code qui ne s'exécute que lorsqu'elle est appelée.

Il accepte des données, appelées paramètres, dans une fonction qui joue le rôle de variables.

Les fonctions sont utilisées pour effectuer certaines actions et sont également appelées méthodes.

## Les Types des Fonctions:

**void:** void signifie que cette méthode n'a pas de valeur de retour (!return).

```
ex: static void myFunction() {  
    // code to be executed  
}
```

**int :** int retourne une numéro.

```
ex: static int myFunction() {  
    // code to be executed  
}
```

**String:** String retourne une chaîne de caractères.

```
ex: static String myFunction() {  
    // code to be executed  
}
```

**Boolean:** retourne True ou False.

```
ex: static boolean myFunction() {  
    // code to be executed  
}
```

