

Java

Java Notebook Demo

Emmanuel BRUNO
Université de Toulon, LIS UMR CNRS 7020
emmanuel.bruno@univ-tln.fr

2026-10-14

Table des matières

1	Introduction	1
1.1	Practice Exercises	1
1.2	Contexte	1
1.3	Premier programme	1
1.4	Types primitifs et calculs	1
1.5	Opérateurs	2
1.6	Structures de contrôle	2
1.7	Tableaux	2
1.8	Méthodes	3
1.9	Programmation orientée objet	3
1.10	Conclusion	3

1 Introduction

Ce document présente Java dans un environnement notebook interactif (IJava). Il couvre les fondamentaux du langage : types primitifs, structures de contrôle, tableaux, méthodes, et principes de programmation orientée objet avec des exemples concrets et exécutables.

1.1 Practice Exercises

Related practice: Java exercises

1.2 Contexte

Le noyau IJava permet d'exécuter du code Java de manière interactive dans Jupyter, offrant un environnement idéal pour l'apprentissage et l'expérimentation. Contrairement aux scripts

Java traditionnels nécessitant une classe `main`, le notebook permet d'exécuter des instructions individuelles et de voir immédiatement les résultats.

1.3 Premier programme

Voici un exemple minimal — les notes contenant une brève explication avant l'exécution.

```
System.out.println("Hello World !");
```

```
Hello World !
```

1.4 Types primitifs et calculs

Exemples et explications sur les types numériques, booléens et caractères.

```
int entier = 42;
double decimal = 3.14159;
boolean vrai = true;
char lettre = 'A';

System.out.println("Entier:   " +
entier);
System.out.println("Décimal:  " +
decimal);
System.out.println("Boolean:  " +
vrai);
```

```
System.out.println("Caractère: " +  
lettre);
```

```
Entier: 42  
Décimal: 3.14159  
Boolean: true  
Caractère: A
```

1.5 Opérateurs

Opérations arithmétiques de base — montrez les résultats pour expliquer les différences (ex: division entière).

```
int a = 10;  
int b = 3;  
System.out.println("Addition: " + (a  
+ b));  
System.out.println("Soustraction: "  
+ (a - b));  
System.out.println("Multiplication:  
" + (a * b));  
System.out.println("Division  
entière: " + (a / b));  
System.out.println("Modulo: " + (a %  
b));
```

```
Addition: 13  
Soustraction: 7  
Multiplication: 30  
Division entière: 3  
Modulo: 1
```

1.6 Structures de contrôle

Présentez if/else, for et while avec commentaires sur l'usage et les bonnes pratiques.

```
int age = 20;  
if (age >= 18) {  
    System.out.println("Majeur");  
} else {  
    System.out.println("Mineur");  
}
```

```
}  
  
for (int i = 1; i <= 5; i++) {  
    System.out.println(i);  
}  
  
int compteur = 0;  
while (compteur < 3) {  
    System.out.println("Tour " +  
compteur);  
    compteur++;  
}
```

```
Majeur  
1  
2  
3  
4  
5  
Tour 0  
Tour 1  
Tour 2
```

1.7 Tableaux

Déclaration, accès et itération sur tableaux.

```
int[] nombres = {10, 20, 30, 40, 50};  
System.out.println("Premier élément:  
" + nombres[0]);  
System.out.println("Dernier élément:  
" + nombres[nombres.length - 1]);  
for (int n : nombres)  
    System.out.println(n);
```

```
Premier élément: 10  
Dernier élément: 50  
10  
20  
30  
40  
50
```

1.8 Méthodes

Définition et appel de méthodes simples
— montrez la signature et un exemple d'utilisation.

```
int additionner(int x, int y) { return  
x + y; }  
System.out.println(additionner(5,7));
```

12

1.9 Programmation orientée objet

Exemple de classe, constructeur et utilisation
— commentez l'encapsulation et les getters/setters.

```
class Person {  
    private String name; private int  
    age;  
    public Person(String name, int age)  
    { this.name = name; this.age = age; }  
    public void greet()  
    { System.out.println("Bonjour, je  
    suis " + name + ", " + age + " ans."); }  
}  
  
Person    alice    =    new  
Person("Alice",29);  
alice.greet();
```

Bonjour, je suis Alice, 29 ans.

1.10 Conclusion

Cette démonstration couvre les bases de Java et fournit des exemples exécutables directement dans le notebook.

Pour aller plus loin : interfaces, classes abstraites, collections (`ArrayList`, `HashMap`), gestion des exceptions, streams et lambda expressions (Java 8+).

- Types primitifs et opérateurs