

# JAVA

# LES FONDAMENTAUX

---

Présenté par :  
**Xavier TABUTEAU**

## 06. Les méthodes

# Les méthodes

Les méthodes sont des fonctions situées dans une classe. Or, Java ne fonctionne qu'avec des classes, donc avec des méthodes. Elles représentent une sorte de programme dans le programme, car elles sont la première forme d'organisation du code. On utilise des méthodes pour regrouper des instructions et les appeler sur demande : chaque fois qu'on a besoin de ces instructions, il suffira d'appeler la méthode au lieu de répéter toutes les instructions. Pour accomplir ce rôle, le cycle de vie d'une méthode se divise en deux phases :

1. Une phase unique dans laquelle la méthode est définie.
2. On définit à ce stade toutes les instructions qui doivent être groupées pour obtenir le résultat souhaité.
3. Une phase d'appel de cette méthode qui peut être répétée autant de fois que désiré.
4. On demande à la méthode de mener à bien toutes les instructions dont elle se compose à un moment donnée dans la logique de notre programme.

## 06. Les méthodes

# Les méthodes

Une méthode peut optionnellement retourner une valeur.

Si elle ne retourne pas de valeur, elle doit être du type « void ».

Si elle retourne une valeur, la dernière instruction dans la méthode devra être « return » suivi de la variable ou valeur à retourner qui doit correspondre au type de la méthode.

Elle peut avoir des paramètres (variables) de façon optionnelle.

S'il y a des paramètres alors l'appel de cette méthode doit contenir le même nombre et type de paramètres.

Exemple de méthode Java très utilisée :

`System.out.println(x);` → x représente ce qu'il faut afficher dans la console java.

## 06. Les méthodes

Syntaxe de la définition d'une méthode :

```
« type » nomMethode(type variable1, ...) {  
    instructions;  
    ...  
    return valeurOuVariable;  
}
```

Syntaxe de l'appel d'une méthode :

```
resultat = NomMethode(variable1, ...);
```

Remarque : resultat doit être du même type que la méthode.

Methodes.java

# JAVA

# LES FONDAMENTAUX

---

Présenté par :  
**Xavier TABUTEAU**