

# JAVA FONDAMENTAUX

---

Présenté par :  
**Xavier TABUTEAU**

## Les structures conditionnelles

# Structures conditionnelles

Les structures de contrôle permettent d'exécuter seulement certaines instructions d'un programme selon la vérification d'une ou plusieurs conditions. Il existe 3 structures conditionnelles. Ces structures sont imbriquables entre elles et les autres types de structures.

- if / else
- switch / case
- ( ? : )

## Les structures conditionnelles

### La structure if / else

Cette structure exécute un bloc d'instructions si la condition au niveau du if est vérifiée. Le else est optionnel et permet d'exécuter un autre bloc d'instructions si la condition du if correspondant est fausse. Toute évaluation qui retourne un booléen peut être utilisée avec le if / else.

Syntaxe d'un if / else :

```
If (variable == valeur) {  
    instructions si la condition est vraie;  
}  
else {  
    instructions si la condition est fausse;  
}
```

## Les structures conditionnelles

# La structure switch / case

Le switch apporte de la clarté au code par rapport au « if / else » dans le cas où un traitement différent doit être appliqué selon les différentes valeurs d'une variable. Plutôt que d'imbriquer des « if / else » on préférera utiliser le switch / case. Cette structure peut également être utilisée pour tester des chaînes de caractères.

Important : Il faut stipuler la sortie du bloc switch dès que l'on a exécuté le traitement voulu. Pour cela nous utilisons l'instruction break.

Si aucun cas ne correspond à la valeur testée, alors c'est celui par défaut (default) qui est exécuté.

```
switch (valeur) {  
    case "A" :  instructions;  
                ...;  
                break;  
    case "B" :  instructions;  
                ...;  
                break;  
    ...  
    default :   instructions;  
                ...;  
                break;  
}
```

## Les structures conditionnelles

# La structure switch / case

A partir du JDK 14, la structure switch a changé et n'a plus besoin de break. Un « case » peut accepter plusieurs valeurs et le « : » devient « -> ». De plus, s'il y a plusieurs instructions pour un « case », ces instructions devront être placées entre accolades comme n'importe quel autre bloc d'instructions.

Syntaxe :

```
switch (valeur) {  
    case "A", "B", ...    -> instruction;  
  
    case "C", ...        -> {  
                            instructions;  
                            ...;  
                        }  
  
    ...  
    default              -> instruction;  
}
```

## Les structures conditionnelles

### La structure switch / case

Ce nouveau switch / case permet aussi d'être utilisé sous forme d'expression. Cela permet par exemple, d'initialiser une variable avec une valeur qui dépend de plusieurs conditions. S'il faut plusieurs lignes on utilisera les accolades pour délimiter le bloc d'instructions, et pour retourner la valeur on doit employer « yield » (comme un return).

Remarque : dans ce cas d'utilisation attention au ; à la fin du switch.

Syntaxe :

```
<type> variable = switch (valeur) {  
    case "A", "B", ... -> valeur_de_retour;  
    case "C", ...      -> {  
                        instructions;  
                        ...;  
                        yield valeur_de_retour;  
                    }  
    ...  
    default            -> instruction;  
};
```

## Les structures conditionnelles

### La structure ternaire ( ? : )

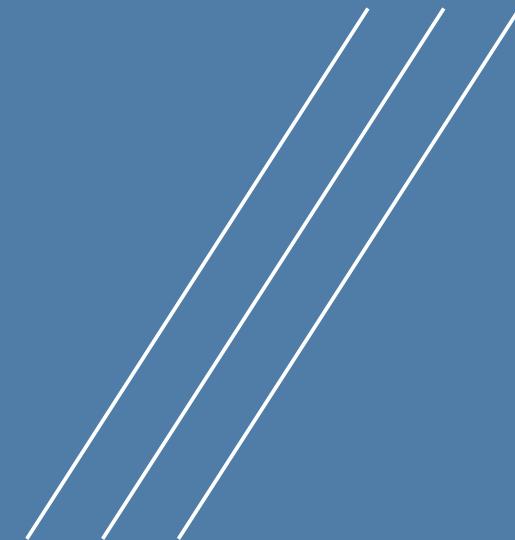
Elle permet de remplacer une structure conditionnelle « if / else » dont le résultat peut être affecté directement à une variable. Le type de variable doit correspondre avec la valeur retournée.

Syntaxe du ternaire :

« type » resultat = (a == b) ? « valeur retournée si vrai » : « valeur retournée si faux »;

Conditions.java

# JAVA FONDAMENTAUX



Présenté par  
**Xavier TABUTEAU**