

## Les traitements conditionnels en Java

`http://alexandre-mesle.com`

22 novembre 2022

## Définition

On appelle traitement conditionnel une portion de code qui n'est pas exécutée systématiquement.

- 1 Le bloc **if**
  - Principe
  - Opérateurs de comparaison
- 2 Si ... Alors ... Sinon
  - Connecteurs logiques
  - Accolades superflues
- 3 Blocs
  - Opérateur ternaire
- 4 Switch
  - Utilisation dans des **if**
- 5 Booléens
  - Les priorités

En pseudo-code un traitement conditionnel se rédige de la sorte :

### Définition

```
Si < condition > alors  
| < instructions >  
Fin si
```

## Définition

En java un traitement conditionnel se formule de la sorte :

```
if (<condition>
{
    <instructions>
}
```

Notez bien qu'il n'y a pas de point-virgule après la parenthèse du `if`.

## Définition

En java un traitement conditionnel se formule de la sorte :

```
if (<condition>
{
    <instructions>
}
```

Notez bien qu'il n'y a pas de point-virgule après la parenthèse du `if`.

- 1 Le bloc **if**
  - Principe
  - Opérateurs de comparaison
- 2 Si ... Alors ... Sinon
  - Connecteurs logiques
  - Accolades superflues
- 3 Blocs
  - Opérateur ternaire
- 4 Switch
  - Utilisation dans des **if**
- 5 Booléens
  - Les priorités

## Définition

- `==` : égalité
- `!=` : non-égalité
- `<`, `<=` : inférieur à, respectivement strict et large
- `>`, `>=` : supérieur à, respectivement strict et large

## Exemple

```
System.out.println("Saisissez une valeur");  
int i = scanner.nextInt();  
if (i == 0)  
{  
    System.out.println("Vous avez saisi une valeur nulle.");  
}  
System.out.println("Au revoir !");
```

- 1 Le bloc **if**
  - Principe
  - Opérateurs de comparaison
- 2 Si ... Alors ... Sinon
  - Connecteurs logiques
  - Accolades superflues
- Blocs
- Opérateur ternaire
- 3 Switch
- 4 Booléens
  - Utilisation dans des **if**
- 5 Les priorités

Il existe une forme étendue de traitement conditionnel, on la note en pseudo-code de la façon suivante :

### Exemple

**Si** *condition* **alors**

| instructions

**Sinon**

| autresinstructions

**Fin si**

## Définition

En java un traitement conditionnel étendu se formule de la sorte :

```
if (<condition>
{
    <instructions1>;
}
else
{
    <instructions2>;
}
```

## Exemple

```
System.out.println("Saisissez une valeur");  
int i = scanner.nextInt();  
if (i == 0)  
{  
    System.out.println("Vous avez saisi une valeur nulle.");  
}  
else  
{  
    System.out.println("La valeur que vous avez saisi, " +  
        i +  
        ", n'est pas nulle.");  
}
```

- 1 Le bloc **if**
  - Principe
  - Opérateurs de comparaison
- 2 Si ... Alors ... Sinon
  - Connecteurs logiques
  - Accolades superflues
- 3 Blocs
  - Opérateur ternaire
- 4 Switch
- 5 Booléens
  - Utilisation dans des **if**
- 6 Les priorités

## Définition

- et : `&&`
- ou : `||`

## Exemple

```
System.out.println("Saisissez deux valeurs numériques : ");  
float i = scanner.nextFloat();  
float j = scanner.nextFloat();  
System.out.print("Le produit de " + i + " par " + j + " est ")  
    ;  
if ((i >= 0 && j >= 0) || (i < 0 && j < 0))  
{  
    System.out.println("positif.");  
}  
else  
{  
    System.out.println("négatif.");  
}
```

- 1 Le bloc **if**
  - Principe
  - Opérateurs de comparaison
- 2 Si ... Alors ... Sinon
  - Connecteurs logiques
  - Accolades superflues
- Blocs
- Opérateur ternaire
- 3 Switch
- 4 Booléens
  - Utilisation dans des **if**
- 5 Les priorités

Lorsqu'une seule instruction d'un bloc **if** doit être exécutée, les accolades ne sont plus nécessaires.

### Exemple

```
System.out.println("Saisissez deux valeurs numériques : ");  
float i = scanner.nextFloat();  
float j = scanner.nextFloat();  
System.out.print("Le produit de " + i + " et " + j + " est ");  
if ((i >= 0 && j >= 0) || (i < 0 && j < 0))  
    System.out.println("positif.");  
else  
    System.out.println("négatif.");
```

Lorsqu'une seule instruction d'un bloc **if** doit être exécutée, les accolades ne sont plus nécessaires.

### Exemple

```
System.out.println("Saisissez deux valeurs numériques : ");  
float i = scanner.nextFloat();  
float j = scanner.nextFloat();  
System.out.print("Le produit de " + i + " et " + j + " est ");  
if ((i >= 0 && j >= 0) || (i < 0 && j < 0))  
    System.out.println("positif.");  
else  
    System.out.println("négatif.");
```

## Définition

Un **bloc** est un ensemble d'instructions délimité par des accolades.

## Exemple

- `public static void main(String[] args)`
- `if.`

Achtung!

## Exemple

```
int i = 4;  
if (i == 4)  
{  
    int j = i + 1;  
}  
System.out.println("j = " + j + ".");
```

- 1 Le bloc **if**
  - Principe
  - Opérateurs de comparaison
- 2 Si ... Alors ... Sinon
  - Connecteurs logiques
  - Accolades superflues
- Blocs
- Opérateur ternaire
- 3 Switch
- 4 Booléens
  - Utilisation dans des **if**
- 5 Les priorités

## Définition

```
<variable> = (<condition>) ? <valeur1> : <valeur2> ;
```

## Exemple

```
max = (i > j) ? i : j ;
```

## Exemple

```
int i = 4;  
int j = 2;  
int k = 7;  
int l;  
System.out.println((i > (l = (j > k) ? j : k)) ? i : l);
```

- 1 Le bloc **if**
  - Principe
  - Opérateurs de comparaison
- 2 Si ... Alors ... Sinon
  - Connecteurs logiques
  - Accolades superflues
- 3 **Switch**
- 4 Booléens
  - Utilisation dans des **if**
- 5 Les priorités
  - Blocs
  - Opérateur ternaire

## Définition

```
switch(<nomvariable>
{
  case <valeur_1> : <instructions_1> ; break ;
  case <valeur_2> : <instructions_2> ; break ;
  /* ... */
  case <valeur_n> : <instructions_n> ; break ;
  default : <instructionspardefaut> ; break ;
}
```

## Exemple

```
switch(numeroMois)
{
    case 1 : System.out.print("janvier"); break ;
    case 2 : System.out.print("fevrier"); break ;
    case 3 : System.out.print("mars"); break ;
    case 4 : System.out.print("avril"); break ;
    case 5 : System.out.print("mai"); break ;
    case 6 : System.out.print("juin"); break ;
    case 7 : System.out.print("juillet"); break ;
    case 8 : System.out.print("aout"); break ;
    case 9 : System.out.print("septembre"); break ;
    case 10 : System.out.print("octobre"); break ;
    case 11 : System.out.print("novembre"); break ;
    case 12 : System.out.print("decembre"); break ;
    default : System.out.print("Je connais pas ce mois...");break;
}
```

- 1 Le bloc **if**
  - Principe
  - Opérateurs de comparaison
- 2 Si ... Alors ... Sinon
  - Connecteurs logiques
  - Accolades superflues
- 3 Blocs
  - Opérateur ternaire
- 3 Switch
- 4 Booléens
  - Utilisation dans des **if**
- 5 Les priorités

## Définition

Le type booléen en Java est **boolean** et une variable de ce type peut prendre soit la valeur **true**, soit la valeur **false**.

## Exemple

```
System.out.println("Saisissez un booléen : ");  
boolean b = scanner.nextBoolean();  
if (b)  
    System.out.println("b is true.");  
else  
    System.out.println("b is false.");
```

## Exemple

```
boolean x = (3>2);
```

## Exemple

```
boolean x = (true && false) || (true);
```

## Définition

Parmi les connecteurs logiques se trouve **!**, dit opérateur de **négation**. La négation d'une expression est vraie si l'expression est fausse, fausse si l'expression est vraie.

## Exemple

```
boolean x = !(3==2);
```

- 1 Le bloc **if**
  - Principe
  - Opérateurs de comparaison
- 2 Si ... Alors ... Sinon
  - Connecteurs logiques
  - Accolades superflues
- 3 Blocs
  - Opérateur ternaire
- 4 Switch
- 4 Booléens
  - Utilisation dans des **if**
- 5 Les priorités

noms	opérateurs
opérateurs unaires	cast, -, ~, !, ++, --
produit	*, /, %
somme	+, -
décalage binaire	>>, <<
comparaison	>, <, >=, <=
égalité	==, !=
ET binaire	&
OU Exclusif binaire	^
OU binaire	
connecteurs logiques	&&,
if ternaire	()?:
affectations	=, +=, -=, ...