

# Branchements conditionnels

```
import java.util.Scanner;

class PremierExempleIf
{
    private static Scanner scanner = new Scanner(System.in);
    public static void main(String[] args) {
        System.out.print("Entrez votre nombre:");
        int n = scanner.nextInt();

        if (n < 5) {
            System.out.println("Votre nombre est plus petit que 5.");
        } else {
            System.out.println("Votre nombre est plus grand ou egal a 5.");
        }

        System.out.println("fin du programme");
    }
}
```

Mot-clé if

```
if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
} else {
    System.out.println("Votre nombre est plus grand ou egal a 5.");
}
```

Condition

```
if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
} else {
    System.out.println("Votre nombre est plus grand ou egal a 5.");
}
```

Une accolade ouvrante

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

Cette instruction sera exécutée si la condition est vraie.

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

Une accolade fermante

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
}  
else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

Le mot-clé else

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

Une accolade ouvrante

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

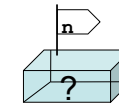
```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

Cette instruction sera exécutée si la condition est *fausse*.

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

Une accolade fermante

```
→ System.out.print("Entrez votre nombre:");  
int n = scanner.nextInt();
```



```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

```
System.out.println("fin du programme");
```

Ce qui s'affiche dans la fenêtre Terminal:

Entrez votre nombre:

|

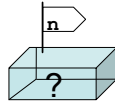
```

System.out.print("Entrez votre nombre:");
→ int n = scanner.nextInt();

if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
} else {
    System.out.println("Votre nombre est plus grand ou egal a 5.");
}

System.out.println("fin du programme");

```



Ce qui s'affiche dans la fenêtre Terminal:

Entrez votre nombre:

|

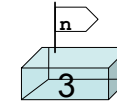
```

System.out.print("Entrez votre nombre:");
→ int n = scanner.nextInt();

if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
} else {
    System.out.println("Votre nombre est plus grand ou egal a 5.");
}

System.out.println("fin du programme");

```



Ce qui s'affiche dans la fenêtre Terminal:

Entrez votre nombre:

3

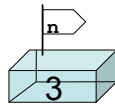
|

```

System.out.print("Entrez votre nombre:");
int n = scanner.nextInt();
→ if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
} else {
    System.out.println("Votre nombre est plus grand ou egal a 5.");
}

System.out.println("fin du programme");

```



Ce qui s'affiche dans la fenêtre Terminal:

Entrez votre nombre:

3

|

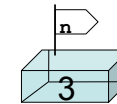
```

System.out.print("Entrez votre nombre:");
int n = scanner.nextInt();

if (n < 5) {
→ System.out.println("Votre nombre est plus petit que 5.");
} else {
    System.out.println("Votre nombre est plus grand ou egal a 5.");
}

System.out.println("fin du programme");

```



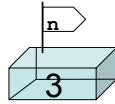
Ce qui s'affiche dans la fenêtre Terminal:

Entrez votre nombre:

3

|

```
System.out.print("Entrez votre nombre:");  
int n = scanner.nextInt();
```



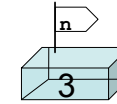
```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

```
System.out.println("fin du programme");
```

Ce qui s'affiche dans la fenêtre Terminal:

```
Entrez votre nombre:  
3  
Votre nombre est plus petit que 5.  
|
```

```
System.out.print("Entrez votre nombre:");  
int n = scanner.nextInt();
```



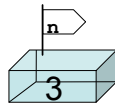
```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

```
System.out.println("fin du programme");
```

Ce qui s'affiche dans la fenêtre Terminal:

```
Entrez votre nombre:  
3  
Votre nombre est plus petit que 5.  
|
```

```
System.out.print("Entrez votre nombre:");  
int n = scanner.nextInt();
```



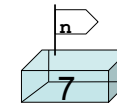
```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

```
System.out.println("fin du programme");
```

Ce qui s'affiche dans la fenêtre Terminal:

```
Entrez votre nombre:  
3  
Votre nombre est plus petit que 5.  
fin du programme  
|
```

```
System.out.print("Entrez votre nombre:");  
int n = scanner.nextInt();
```



```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

```
System.out.println("fin du programme");
```

Ce qui s'affiche dans la fenêtre Terminal:

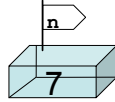
```
Entrez votre nombre:  
7  
|
```

```

System.out.print("Entrez votre nombre:");
int n = scanner.nextInt();
?
→ if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
} else {
    System.out.println("Votre nombre est plus grand ou egal a 5.");
}

System.out.println("fin du programme");

```



Ce qui s'affiche dans la fenêtre Terminal:

```

Entrez votre nombre:
7
|

```

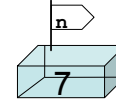
```

System.out.print("Entrez votre nombre:");
int n = scanner.nextInt();

if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
} else {
→ System.out.println("Votre nombre est plus grand ou egal a 5.");
}

System.out.println("fin du programme");

```



Ce qui s'affiche dans la fenêtre Terminal:

```

Entrez votre nombre:
7
Votre nombre est plus grand ou egal a 5.
|

```

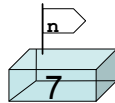
```

System.out.print("Entrez votre nombre:");
int n = scanner.nextInt();

if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
} else {
    System.out.println("Votre nombre est plus grand ou egal a 5.");
}

→ System.out.println("fin du programme");

```



Ce qui s'affiche dans la fenêtre Terminal:

```

Entrez votre nombre:
7
Votre nombre est plus grand ou egal a 5.
fin du programme
|

```

```

if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
} else {
    System.out.println("Votre nombre est plus grand ou egal a 5.");
}

```

Les accolades délimitent un bloc d'instructions

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

Les accolades délimitent un bloc d'instructions

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

On peut mettre autant d'instructions qu'on veut dans un bloc.  
Supposons qu'on veuille aussi afficher la valeur de  $n$  quand  $n$  est plus petit que 5.  
Il suffit d'ajouter une instruction dans le premier bloc:

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
    System.out.println("Votre nombre est " + n);  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

Quand un bloc contient une seule instruction, il n'est pas obligatoire d'utiliser des accolades.  
On aurait pu écrire:

```
if (n < 5)  
    System.out.println("Votre nombre est plus petit que 5.");  
else  
    System.out.println("Votre nombre est plus grand ou egal a 5.");
```

Néanmoins, une bonne pratique est de toujours utiliser des blocs, même quand il n'y a qu'une seule instruction.  
Ca facilite l'ajout d'instructions.

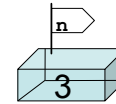
Une instruction `if` peut ne pas avoir de deuxième partie.  
Par exemple, si on veut ne rien afficher si `n` est plus grand ou égal à 5, il suffit d'enlever la deuxième partie, à partir du `else`:

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
}
```

```
System.out.print("Entrez votre nombre: ");  
int n = scanner.nextInt();
```

```
→ if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
}
```

```
System.out.println("Au revoir");
```



Ce qui s'affiche dans la fenêtre Terminal:

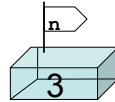
```
Entrez votre nombre: 3
```

```
|
```

```
System.out.print("Entrez votre nombre: ");  
int n = scanner.nextInt();
```

```
→ if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
}
```

```
System.out.println("Au revoir");
```



Ce qui s'affiche dans la fenêtre Terminal:

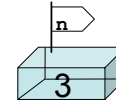
```
Entrez votre nombre: 3  
Votre nombre est plus petit que 5.
```

```
|
```

```
System.out.print("Entrez votre nombre: ");  
int n = scanner.nextInt();
```

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
}
```

```
→ System.out.println("Au revoir");
```



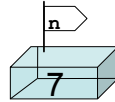
Ce qui s'affiche dans la fenêtre Terminal:

```
Entrez votre nombre: 3  
Votre nombre est plus petit que 5.  
Au revoir
```

```
|
```



```
System.out.print("Entrez votre nombre: ");
int n = scanner.nextInt();
```



```
→ if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
}

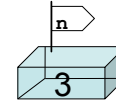
System.out.println("Au revoir");
```

Ce qui s'affiche dans la fenêtre Terminal:

```
Entrez votre nombre: 7
```

```
|
```

```
System.out.print("Entrez votre nombre: ");
int n = scanner.nextInt();
```



```
if (n < 5) {
    System.out.println("Votre nombre est plus petit que 5.");
}
```

```
→ System.out.println("Au revoir");
```

Ce qui s'affiche dans la fenêtre Terminal:

```
Entrez votre nombre: 7
```

```
Au revoir
```

```
|
```

## Les choix imbriqués

L'instruction `if` suit donc le schéma:

```
if (condition1) {
    ...
} else {
    ...
}
```

Les instructions figurant dans les blocs sont absolument quelconques. *Il peut donc s'agir d'autres instructions `if`.*

## Choix imbriqués: exemple

```
if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}
```

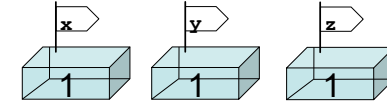
## Choix imbriqués: exemple

```
if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}
```

**Attention** à ne pas abuser de cette solution.

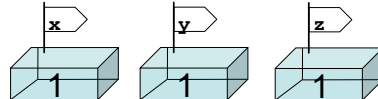
Au-delà de 3 niveaux, le code devient vite illisible!

Supposons:



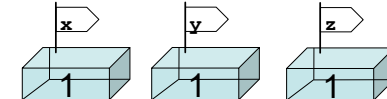
```
→ if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}
```

Supposons:



```
if (x == y) {
    → if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}
```

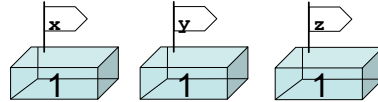
Supposons:



```
if (x == y) {
    if (y == z) {
        → System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}
```

Les trois valeurs sont egales. est affiché

Supposons:

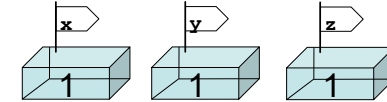


```

if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}

```

Supposons:

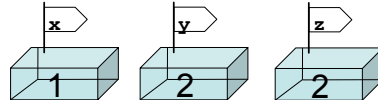


```

if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}

```

Supposons:

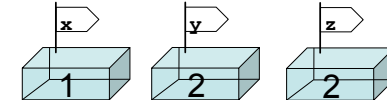


```

→ if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}

```

Supposons:

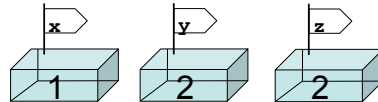


```

if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    → if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}

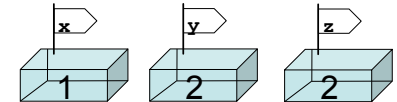
```

Supposons:



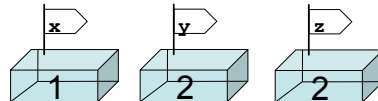
```
if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}
```

Supposons:



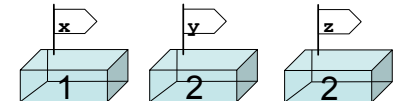
```
if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Seules les deux dernieres valeurs sont egales. est affiche");
        }
    }
}
```

Supposons:



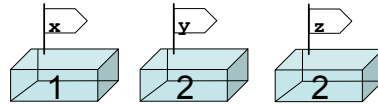
```
if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}
```

Supposons:



```
if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}
```

Supposons:



```
if (x == y) {  
    if (y == z) {  
        System.out.println("Les trois valeurs sont egales.");  
    } else {  
        System.out.println("Seules les deux premieres valeurs sont egales.");  
    }  
} else {  
    if (x == z) {  
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");  
    } else {  
        if (y == z) {  
            System.out.println("Seules les deux dernieres valeurs sont egales.");  
        } else {  
            System.out.println("Les trois valeurs sont differentes.");  
        }  
    }  
}
```

# Conditions

## Les conditions

L'instruction `if` fait apparaître une **condition** entre parenthèses

**Condition**

```
if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou égal a 5.");  
}
```

**Attention**, la condition est toujours entourée de parenthèses.

Pour l'instant, nous n'avons rencontré qu'une condition simple, `n < 5`

Nous allons voir maintenant comment s'écrivent les conditions d'une façon générale.

## Les conditions simples Les opérateurs de comparaison

Une **condition simple** compare deux expressions.  
Elle utilise un **opérateur de comparaison**, comme `<` ou `>`

Opérateurs de comparaison du langage Java:

Opérateur de comparaison	Signification
<code>&lt;</code>	inférieur à
<code>&gt;</code>	supérieur à
<code>==</code>	égal à
<code>&lt;=</code>	inférieur ou égal à
<code>&gt;=</code>	supérieur ou égal à
<code>!=</code>	différent de

## Les conditions Les opérateurs de comparaison

**Attention:**  
L'opérateur pour tester si deux valeurs sont égales s'écrit avec deux signes égal `==`

Une condition simple utilise un opérateur de comparaison

Elle utilise un opérateur de comparaison

Opérateurs de comparaison

`if (n == 5)`

et non pas:

~~`if (n = 5)`~~

`==`

égal à

`<=`

inférieur ou égal à

`>=`

supérieur ou égal à

`!=`

différent de

**Attention:**  
Il n'y a pas d'espaces entre les deux caractères

Opérateur de comparaison



==  
<=  
>=  
!=

Signification  
inférieur à  
supérieur à  
égal à  
inférieur ou égal à  
supérieur ou égal à  
différent de

```
int a = 1;
int b = 2;

if (a == b) {
    System.out.println("Cas 1");
} else {
    System.out.println("Cas 2");
}

if (2 * a == b){
    System.out.println("b est egal au double de a.");
}

affiche

Cas 2
b est egal au double de a.
```

```
int a = 1;
int b = 2;

if (a != b) {
    System.out.println("Cas 2");
} else {
    System.out.println("Cas 1");
}

if (2 * a != b) {
    System.out.println("b est different du double de a.");
}

affiche

Cas 2
```

```
int a = 1;
int b = 2;

if (a <= b) {
    System.out.println("Cas 3");
} else {
    System.out.println("Cas 4");
}

if (2 * a <= b) {
    System.out.println("b est superieur ou egal au double de a.");
}

affiche

Cas 3
b est superieur ou egal au double de a.
```

## Les opérateurs logiques

On peut relier des conditions simples par des opérateurs logiques.

L'opérateur logique **&&** (ET):

par exemple, la condition

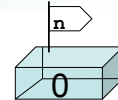
$(a < b) \ \&\& \ (c < d)$

est vraie **uniquement** si les deux conditions  $(a < b)$  et  $(c < d)$  sont toutes les deux vraies.

## Exemple avec l'opérateur logique &&

```
System.out.println("Entrez un nombre entre 1 et 10:");  
int n = scanner.nextInt();
```

```
if ((n >= 1) && (n <= 10)) {  
    System.out.println("correct");  
} else {  
    System.out.println("incorrect");  
}
```



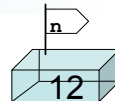
Supposons que la valeur  
entrée pour `n` soit 0

incorrect est affiché

## Exemple avec l'opérateur logique &&

```
System.out.println("Entrez un nombre entre 1 et 10:");  
int n = scanner.nextInt();
```

```
if ((n >= 1) && (n <= 10)) {  
    System.out.println("correct");  
} else {  
    System.out.println("incorrect");  
}
```



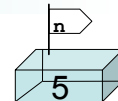
Supposons que la valeur  
entrée pour `n` soit 12

incorrect est affiché

## Exemple avec l'opérateur logique &&

```
System.out.println("Entrez un nombre entre 1 et 10:");  
int n = scanner.nextInt();
```

```
if ((n >= 1) && (n <= 10)) {  
    System.out.println("correct");  
} else {  
    System.out.println("incorrect");  
}
```



Supposons que la valeur  
entrée pour `n` soit 5

correct est affiché



## Les opérateurs logiques

L'opérateur logique || (OU):

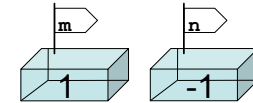
par exemple, la condition

$(a < b) \ || \ (c < d)$

est vraie **si au moins une** des deux conditions  $(a < b)$  ou  $(c < d)$  est vraie.

## Exemple avec l'opérateur logique ||

```
System.out.println("Entrez deux valeurs:");
int m = scanner.nextInt();
int n = scanner.nextInt();
```



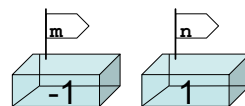
```
if ((m >= 0) || (n >= 0)) {
    System.out.println("au moins une valeur est positive");
} else {
    System.out.println("les deux valeurs sont negatives");
}
```

Supposons que la valeur entrée pour m soit +1, et la valeur entrée pour n soit -1

au moins une valeur est positive est affiché

## Exemple avec l'opérateur logique ||

```
System.out.println("Entrez deux valeurs:");
int m = scanner.nextInt();
int n = scanner.nextInt();
```



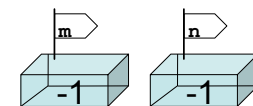
```
if ((m >= 0) || (n >= 0)) {
    System.out.println("au moins une valeur est positive");
} else {
    System.out.println("les deux valeurs sont negatives");
}
```

Supposons que la valeur entrée pour m soit -1, et la valeur entrée pour n soit +1

au moins une valeur est positive est affiché

## Exemple avec l'opérateur logique ||

```
System.out.println("Entrez deux valeurs:");
int m = scanner.nextInt();
int n = scanner.nextInt();
```



```
if ((m >= 0) || (n >= 0)) {
    System.out.println("au moins une valeur est positive");
} else {
    System.out.println("les deux valeurs sont negatives");
}
```

Supposons que la valeur entrée pour m soit -1, et la valeur entrée pour n soit -1

les deux valeurs sont negatives est affiché

## Les opérateurs logiques

**L'opérateur logique ! (NON):**

par exemple, la condition

`!(a < b)`

est vraie si `(a < b)` est fausse, et fausse si `(a < b)` est vraie.

Nous verrons des exemples d'utilisation de cet opérateur plus loin dans la suite du cours.

## Erreurs de débutants

### Erreurs classiques

Le test d'égalité s'écrit `==`, et pas `=`

```
if (a = 1) // !!!
```

n'est pas accepté par le compilateur.

### Erreurs classiques

```
if (a == 1); // !!!  
    System.out.println("a vaut 1");
```

a vaut 1 est toujours affiché quelle que soit la valeur de a!

Le point-virgule est considéré comme une instruction, qui ne fait rien.

Le code précédent est compris par le compilateur comme:

```
if (a == 1)  
;  
System.out.println("a vaut 1");
```

l'instruction `System.out.println("a vaut 1");` est donc située après le `if`.

Si on utilise des accolades même quand il n'y a qu'une instruction dans le bloc, et qu'on écrit le test de la façon suivante:

```
if (a == 1) {  
    System.out.println("a vaut 1");  
}
```

l'erreur précédente a beaucoup moins de chance d'arriver.

## Erreurs classiques

Ne pas oublier les accolades, l'indentation ne suffit pas:

```
if (n < p)
    System.out.println("n est plus petit que p");
    max = p;
else
    System.out.println("n est plus grand ou egal a p");
```

génère à la compilation l'erreur:

**error: 'else' without 'if'**

Voici une meilleure présentation du code précédent:

```
if (n < p)
    System.out.println("n est plus petit que p");

max = p;
else
    System.out.println("n est plus grand ou egal a p");
```

```
System.out.println("Entrez le premier nombre:");
int n = scanner.nextInt();
System.out.println("Entrez le deuxieme nombre:");
int p = scanner.nextInt();
```

```
if ((n < p) && (2 * n >= p)) {
    System.out.print("1");
}
```

```
if ((n < p) || (2 * n >= p)) {
    System.out.print("2");
}
```

```
if (n < p) {
    if (2 * n >= p) {
        System.out.print("3");
    } else {
        System.out.print("4");
    }
}
```

```
System.out.println();
```

Qu'affiche ce programme quand l'utilisateur entre 1 et 2 ?

A: 2  
B: 24  
C: 123  
D: 1234

?

```
System.out.println("Entrez le premier nombre:");
int n = scanner.nextInt();
System.out.println("Entrez le deuxieme nombre:");
int p = scanner.nextInt();
```

```
if ((n < p) && (2 * n >= p)) {
    System.out.print("1");
}
```

```
if ((n < p) || (2 * n >= p)) {
    System.out.print("2");
}
```

```
if (n < p) {
    if (2 * n >= p) {
        System.out.print("3");
    } else {
        System.out.print("4");
    }
}
```

```
System.out.println();
```

Qu'affiche ce programme quand l'utilisateur entre 1 et 3 ?

A: 2  
B: 24  
C: 123  
D: 1234

?

```
System.out.println("Entrez le premier nombre:");
int n = scanner.nextInt();
System.out.println("Entrez le deuxieme nombre:");
int p = scanner.nextInt();
```

```
if ((n < p) && (2 * n >= p)) {
    System.out.print("1");
}
```

```
if ((n < p) || (2 * n >= p)) {
    System.out.print("2");
}
```

```
if (n < p) {
    if (2 * n >= p) {
        System.out.print("3");
    } else {
        System.out.print("4");
    }
}
```

```
System.out.println();
```

Qu'affiche ce programme quand l'utilisateur entre 2 et 1 ?

A: 2  
B: 24  
C: 123  
D: 1234

?

# Le type booléen (boolean)

## Le type boolean

Le type `boolean` (pour booléen) est le type des **conditions**.

Il permet de déclarer des **variables contenant la valeur d'une condition**.

Une variable de type booléen est souvent appelée simplement un **booléen**.

Un booléen ne peut prendre que **deux valeurs possibles**: `true` ou `false`

Exemple:

```
int a = 1, b = 2;
boolean test1 = (a == b);
boolean test2 = (a < b);
```

## Le type bool

Le type `boolean` (pour booléen) est le type des **conditions**.

Il permet de déclarer des **variables contenant la valeur d'une condition**.

Une variable de type booléen est souvent appelée simplement un **booléen**.

Un booléen ne peut prendre que **deux valeurs possibles**: `true` ou `false`

Exemple:

```
int a = 1, b = 2;
→ boolean test1 = (a == b);
boolean test2 = (a < b);
```

## Le type bool

Le type `boolean` (pour booléen) est le type des **conditions**.

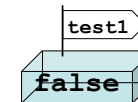
Il permet de déclarer des **variables contenant la valeur d'une condition**.

Une variable de type booléen est souvent appelée simplement un **booléen**.

Un booléen ne peut prendre que **deux valeurs possibles**: `true` ou `false`

Exemple:

```
int a = 1, b = 2;
→ boolean test1 = (a == b);
boolean test2 = (a < b);
```



## Le type bool

Le type `boolean` (pour booléen) est le type des **conditions**.

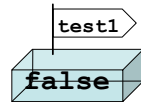
Il permet de déclarer des **variables contenant la valeur d'une condition**.

Une variable de type booléen est souvent appelée simplement un **booléen**.

Un booléen ne peut prendre que **deux valeurs possibles**: `true` ou `false`

Exemple:

```
int a = 1, b = 2;  
boolean test1 = (a == b);  
→ boolean test2 = (a < b);
```



## Le type bool

Le type `boolean` (pour booléen) est le type des **conditions**.

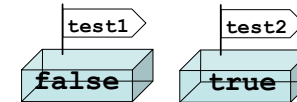
Il permet de déclarer des **variables contenant la valeur d'une condition**.

Une variable de type booléen est souvent appelée simplement un **booléen**.

Un booléen ne peut prendre que **deux valeurs possibles**: `true` ou `false`

Exemple:

```
int a = 1, b = 2;  
boolean test1 = (a == b);  
→ boolean test2 = (a < b);
```



On peut initialiser des booléens à l'aide des constantes `false` et `true`.

On peut utiliser des booléens comme des conditions. Par exemple:

- on peut utiliser des opérateurs logiques (`&&`, `||` et `!`) entre booléens;
- on peut utiliser un booléen comme condition dans un `if`.

```
int a = 1, b = 2;  
  
boolean c = true;  
boolean d = (a == b);  
boolean e = (d || (a < b));
```

```
if (e) {  
    System.out.println("e vaut true");  
}
```

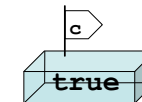
On peut initialiser des booléens à l'aide des constantes `false` et `true`.

On peut utiliser des booléens comme des conditions. Par exemple:

- on peut utiliser des opérateurs logiques (`&&`, `||` et `!`) entre booléens;
- on peut utiliser un booléen comme condition dans un `if`.

```
int a = 1, b = 2;  
  
→ boolean c = true;  
boolean d = (a == b);  
boolean e = (d || (a < b));
```

```
if (e) {  
    System.out.println("e vaut true");  
}
```



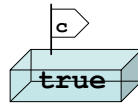
On peut initialiser des booléens à l'aide des constantes `false` et `true`.

On peut utiliser des booléens comme des conditions. Par exemple:

- on peut utiliser des opérateurs logiques (`&&`, `||` et `!`) entre booléens;
- on peut utiliser un booléen comme condition dans un `if`.

```
int a = 1, b = 2;  
  
boolean c = true;  
→ boolean d = (a == b);  
boolean e = (d || (a < b));
```

```
if (e) {  
    System.out.println("e vaut true");  
}
```



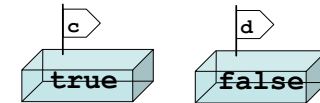
On peut initialiser des booléens à l'aide des constantes `false` et `true`.

On peut utiliser des booléens comme des conditions. Par exemple:

- on peut utiliser des opérateurs logiques (`&&`, `||` et `!`) entre booléens;
- on peut utiliser un booléen comme condition dans un `if`.

```
int a = 1, b = 2;  
  
boolean c = true;  
→ boolean d = (a == b);  
boolean e = (d || (a < b));
```

```
if (e) {  
    System.out.println("e vaut true");  
}
```



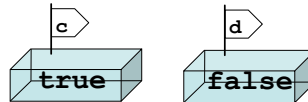
On peut initialiser des booléens à l'aide des constantes `false` et `true`.

On peut utiliser des booléens comme des conditions. Par exemple:

- on peut utiliser des opérateurs logiques (`&&`, `||` et `!`) entre booléens;
- on peut utiliser un booléen comme condition dans un `if`.

```
int a = 1, b = 2;  
  
boolean c = true;  
boolean d = (a == b);  
→ boolean e = (d || (a < b));
```

```
if (e) {  
    System.out.println("e vaut true");  
}
```



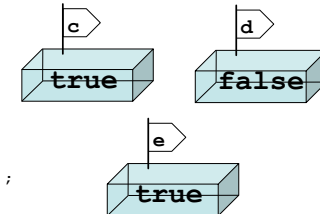
On peut initialiser des booléens à l'aide des constantes `false` et `true`.

On peut utiliser des booléens comme des conditions. Par exemple:

- on peut utiliser des opérateurs logiques (`&&`, `||` et `!`) entre booléens;
- on peut utiliser un booléen comme condition dans un `if`.

```
int a = 1, b = 2;  
  
boolean c = true;  
boolean d = (a == b);  
→ boolean e = (d || (a < b));
```

```
if (e) {  
    System.out.println("e vaut true");  
}
```



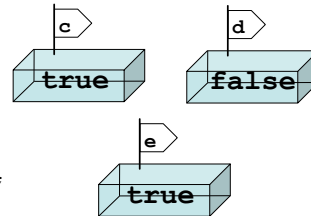
On peut initialiser des booléens à l'aide des constantes `false` et `true`.

On peut utiliser des booléens comme des conditions. Par exemple:

- on peut utiliser des opérateurs logiques (`&&`, `||` et `!`) entre booléens;
- on peut utiliser un booléen comme condition dans un `if`.

```
int a = 1, b = 2;  
  
boolean c = true;  
boolean d = (a == b);  
boolean e = (d || (a < b));
```

```
→ if (e) {  
    System.out.println("e vaut true");  
}
```



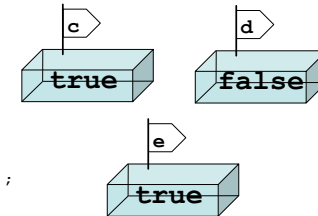
On peut initialiser des booléens à l'aide des constantes `false` et `true`.

On peut utiliser des booléens comme des conditions. Par exemple:

- on peut utiliser des opérateurs logiques (`&&`, `||` et `!`) entre booléens;
- on peut utiliser un booléen comme condition dans un `if`.

```
int a = 1, b = 2;  
  
boolean c = true;  
boolean d = (a == b);  
boolean e = (d || (a < b));
```

```
if (e) {  
→ System.out.println("e vaut true");  
}
```



Les booléens sont utiles pour de nombreux problèmes, nous rencontrerons des exemples concrets dans la suite du cours.