

# 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 compare deux expressions.

Elle utilise un opérateur de comparaison, comme `<` ou `>`

Opérateurs de comparaison du langage Java:

```
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

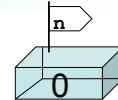
$(a < b) \ \&\& \ (c < d)$

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();
```

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



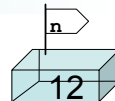
Supposons que la valeur  
entrée pour `n` soit 0

incorrect est affiché

## Exemple avec l'opérateur logique &&

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

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



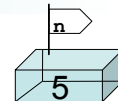
Supposons que la valeur  
entrée pour `n` soit 12

incorrect est affiché

## Exemple avec l'opérateur logique &&

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

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



Supposons que la valeur  
entrée pour `n` soit 12

correct est affiché

## Les opérateurs logiques

L'opérateur logique || (OU):

par exemple, la condition

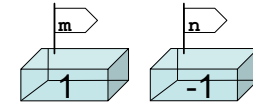
$(a < b) \text{ || } (c < d)$

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");
}
```



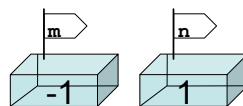
Supposons que la valeur entrée pour m soit +1, et la valeur entrée pour n soit -1

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");
}
```



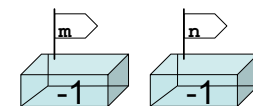
Supposons que la valeur entrée pour m soit -1, et la valeur entrée pour n soit +1

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.