

Conditions (if, else, switch)

```
System.out.print("Sélectionnez un nombre: ");
int first = io.Console.lireInt();

if (first == 42) {
    System.out.println("Félicitations, vous avez trouvé la réponse à la vie");
} else if (first > 40 && first < 50) {
    System.out.println("Vous y êtes presque");
} else {
    System.out.println("Vous n'avez rien trouvé du tout");
}

// Comparer des Strings ou autres objets ne se fait pas de la même manière
System.out.print("Quel est votre nom ? ");
String nom = io.Console.lireString();
if (nom.equals("Roger")) {
    // faire quelque chose
}

// Donne le jour en texte à partir d'un numéro, cas par cas
int jour = 4;
switch (jour) {
    case 1:
        System.out.println("Lundi");
        break;
    case 2:
        System.out.println("Mardi");
        break;
    case 3:
        System.out.println("Mercredi");
        break;
    case 4:
        System.out.println("Jeudi");
        break;
    case 5:
        System.out.println("Vendredi");
}
```

```

    break;
case 6:
    System.out.println("Samedi");
    break;
case 7:
    System.out.println("Dimanche");
    break;
} // "Jeudi"

switch (jour) {
    case 6:
        System.out.println("Samedi");
        break;
    case 7:
        System.out.println("Dimanche");
        break;
    default:
        System.out.println("En attente du weekend");
} // "En attente du weekend"

```

Il y a deux types de conditions en java, les `if else` et les `switch`.

If, Else if, Else

Les `if` permettent de mettre une condition pour l'exécution d'un code.

Il existe plusieurs mots clés comme vu dans l'exemple du début

- `if (...) {` : si *condition* alors ...
- `else if (...) {` : si la contion précédente du bloc `if` n'a pas réussi alors si *condition*, ...
- `else` : si aucune des conditions précédentes du bloc `if` n'a réussi alors ...

Ces conditions fonctionnent en utilisant un "opérateur conditionnel" entre deux nombres. En voici un exemple :

Opérateur contionnel	Description
<code>a == b</code>	A est égal à B
<code>a != b</code>	A est différent de B
<code>a > b</code>	A est plus grand que B
<code>a < b</code>	A est plus petit que B

Opérateur contionnel	Description
<code>a >= b</code>	A est plus grand ou égal à B
<code>a <= b</code>	A est plus petit ou égal à B
<code>a</code>	A est vrai
<code>!a</code>	A est faux (en vérité ceci est un opérateur binaire et non un opérateur conditionnel)

On peut aussi mettre plusieurs conditions ensemble avec des [opérateurs logiques](#) (ce sont les même qu'au cours de math et d'archi) :

Opérateur logique	Description
<code>&&</code>	Les deux conditions doivent être remplies
<code> </code>	Au moins une des deux conditions doit être remplie
<code>^</code>	Une seule des deux conditions doit être remplie mais pas plus

Cas spéciaux avec les Strings

On ne peut pas utiliser un opérateur tel que `==` sur un String car un String est un objet de la classe String. Si on utilise l'opérateur logique, on va comparer la référence mémoire de la variable et non la valeur en elle même.

Il faut donc utiliser la méthode `.equals(String autreString)` à la place comme vu dans l'exemple.

switch

Si on veut limiter l'usage des `if else` et que l'on veut tester si une valeur correspond à x, y ou z valeur, on peut utiliser `switch`

On donne à switch la valeur à tester et plusieurs cas (`case`) de valeur possible. Pour chacune d'entre elle, quelque chose à faire avec.

On peut utiliser le mot clé `break` pour arrêter le switch sans tester le reste.

Enfin il y a le `default` qui fonctionne comme le `else`, le default, c'est quand aucun autre cas n'est passé.

En savoir plus

- [Wikiversity FR - Boucle et structures conditionnelles](#)
- [Wikiversity FR - Opérations](#)

Revision #1

Created 27 April 2023 04:31:28 by SnowCode

Updated 27 April 2023 04:35:12 by SnowCode