

Structure conditionnelle

- SI
- Opérateur de comparaison
- Opérateur d'aggrégation
- Booléen
- SINON
- SINON SI
- Priorité des opérations

Structure conditionnelle

- Jusqu'ici nous faisons des structures impératives. C'est-à-dire que chaque ligne de code était exécuté, une à la suite de l'autre.
- Nous allons maintenant ajouter des structures conditionnelles. C'est-à-dire que certaine ligne de code ne s'exécuteront, que si une condition est respecté.
- Cela ce fait en 2 étapes:
 - Déterminer la conditions
 - Écrire dans un bloc, le code qui sera exécuté si la condition est respecté.

SI

Voyons un exemple du concept en pseudo-code pour un programme qui détermine si une personne est majeur à partir de son âge.

```
entier AGE_OF_MAJORITY = 18
entier age;

ECRIRE "Veuillez écrire votre âge: "
LIRE age

SI age >= AGE_OF_MAJORITY // Déterminer la condition. La condition doit toujours être une valeur booléenne (vrai / faux)
    ECRIRE "Vous êtes un adulte" // Écrire le bloc de code qui s'exécute seulement si la condition est respectée.
FINSI // Fin du bloc de code propre à condition
```

Opérateur de comparaison

- Un opérateur de comparaison sert à comparer deux valeurs et à retourner une valeur booléenne issue du résultat de la comparaison.
- Voici la liste des opérateurs de comparaison que nous pouvons utiliser pour déterminer une valeur booléenne.
 - `==` égalité *Notez ici l'utilisation de 2 `=`, car l'opérateur `=` est utilisé pour l'affectation.*
 - `!=` différent de
 - `>` plus grand que
 - `<` plus petit que
 - `>=` plus grand ou égale à
 - `<=` plus petit ou égale à
 - `!` non

Opérateur d'agrégation

- Il est possible de combiner plusieurs comparaisons dans une seule condition avec les opérateurs d'agrégation ET et OU.
- Par exemple, si je reprend mon exemple précédent, mais que cette fois je veux vérifier si la personne est majeur au Québec, mais ne l'est pas au États-unis.

```
entier AGE_OF_MAJORITY_QC = 18
entier AGE_OF_MAJORITY_USA = 21
entier age;

Ecrire "Veuillez écrire votre âge: "
Lire age

SI age >= AGE_OF_MAJORITY_QC ET age < AGE_OF_MAJORITY_USA
    Ecrire "Vous êtes un adulte au Québec, mais pas au États-unis"
FINSI
```

ET : Les deux conditions doivent être vrai pour entrer dans la condition

OU : Une des deux conditions doit être vrai pour entrer dans la condition

Booléen

- Nous avons vu que les variables de type booléen contiennent une valeur vrai ou faux.
- Ces variables peuvent donc être utilisées directement dans une condition ou même pour stocker une condition.
- Par exemple:

```
entier AGE_OF_MAJORITY = 18
entier age;

Ecrire "Veuillez écrire votre âge: "
Lire age

BOOLEEN isInQc = true
BOOLEEN isMajor = age >= AGE_OF_MAJORITY

SI isMajor ET isInQc
    Ecrire "Vous êtes un adulte"
FINSI
```

SINON

- Chaque bloc de condition peut avoir un second bloc de code qui s'exécute uniquement si la condition n'est pas respecté.
- On appelle ce bloc le SINON.
- Par exemple:

```
entier AGE_OF_MAJORITY = 18
entier age;

Ecrire "Veuillez écrire votre âge: "
Lire age

SI age >= AGE_OF_MAJORITY
    Ecrire "Vous êtes un adulte"
SINON
    Ecrire "Vous n'êtes pas encore un adulte"
FINSI
```

```
SI !isGood  
    ECRIRE "VRAI"  
SINON  
    ECRIRE "FAUX"  
FINSI
```

```
!isGood  
!true  
false  
"FAUX"
```

- Est-ce que le programme suivant affichera "VRAI" ou "FAUX"

```
ETNIER age = 56;  
SI age > 75 || age < 60
```