

# Leçon 11

## Les Conditions avec if

---



### Objectif

Dans cette leçon, vous apprendrez à utiliser les conditions avec l'instruction `if` pour faire des choix dans votre code. Les conditions permettent de déterminer quelles instructions doivent être exécutées en fonction de certaines vérifications.

### Qu'est-ce qu'une Condition ?

Les conditions sont comme des questions que l'on pose au programme. Par exemple, vous pouvez demander : "Est-ce que l'âge d'une personne est supérieur à 18 ans ?" En fonction de la réponse (vraie ou fausse), vous pouvez décider de faire quelque chose ou non.

### L'instruction `if`

L'instruction `if` permet d'exécuter du code uniquement si une condition est vraie. Voici la structure de base :

```
if (condition) {  
    // Code à exécuter si la condition est vraie  
}
```

### Exemple Simple

Imaginons que vous voulez afficher un message si une personne est majeure (c'est-à-dire si son âge est supérieur ou égal à 18 ans). Voici comment vous pourriez le faire :

```
val age = 20  
  
setContent {  
    if (age >= 18) {  
        Text( text: "Vous êtes majeur !")  
    }  
}
```

Dans cet exemple, si age est supérieur ou égal à 18, le message "Vous êtes majeur !" sera affiché sur l'écran.

## L'Instruction if-else

Vous pouvez aussi ajouter une alternative avec `else`. Cela permet de définir ce qui doit se passer si la condition est fausse.

```
val age = 16

setContent {
    if (age >= 18) {
        Text(text: "Vous êtes majeur !")
    } else {
        Text(text: "Vous êtes mineur.")
    }
}
```

Dans cet exemple, si age est inférieur à 18, le message "Vous êtes mineur." sera affiché.

## L'Instruction if-else if-else

Pour tester plusieurs conditions, vous pouvez utiliser `else if`. Cela permet de vérifier plusieurs possibilités.

```
val note = 85

setContent {
    if (note >= 90) {
        Text(text: "Excellent")
    } else if (note >= 70) {
        Text(text: "Bien")
    } else if (note >= 50) {
        Text(text: "Passable")
    } else {
        Text(text: "Échec")
    }
}
```

Dans cet exemple, le programme vérifie plusieurs plages de valeurs et affiche un message approprié en fonction de la note.

## Manipulation des Conditions avec des Variables

Vous pouvez combiner les variables que vous avez apprises précédemment avec des conditions pour effectuer des vérifications plus complexes.

### Exemple combiné

```
val temperature = 18.5
val estWeekend = true

setContent {
    if (temperature >= 20.0 && estWeekend) {
        Text(text: "C'est un bon jour pour faire une promenade.")
    } else {
        Text(text: "Ce n'est pas le meilleur jour pour sortir.")
    }
}
```

Dans cet exemple, la promenade est recommandée seulement si la température est supérieure ou égale à 20°C et c'est le week-end.

## Exercices

### Exercice 1

- Créez une variable `age` et affectez-lui une valeur.
- Affichez un message indiquant si la personne a droit à un tarif réduit (si elle a moins de 12 ans ou plus de 65 ans).

### Exercice 2

- Créez une variable `quantiteEnStock` et affectez-lui une valeur.
- Affichez un message indiquant si le produit est en stock ou s'il est épuisé (si la quantité est supérieure à 0).

### Exercice 3

- Créez une variable `montantAchat` de type `Float` et une variable `reduction` de type `Boolean`.
- Si `reduction` est `true`, appliquez une réduction de 10% au montant. Affichez le montant final après réduction.

## Exercice 4

- Créez une variable `joursRestants` de type `Int`.
- Si le nombre de jours restants est inférieur ou égal à 7, affichez un message indiquant "L'événement approche !". Sinon, affichez "Il reste encore du temps."

## Exercice 5

- Créez une variable `score` de type `Int`.
- Affichez "Excellent" si le score est supérieur ou égal à 90, "Bien" si le score est entre 70 et 89, et "À améliorer" si le score est inférieur à 70.