Leçon 12
Les Conditions avec when

Objectif

Dans cette leçon, vous apprendrez à utiliser l'instruction when pour gérer des conditions plus complexes et faire des choix dans votre code. when est une alternative pratique à l'instruction if pour comparer des valeurs et exécuter du code en fonction de ces valeurs.

Qu'est-ce que l'instruction when ?

L'instruction when est comme une boîte à choix. Vous lui donnez une valeur, et elle vérifie cette valeur contre plusieurs options possibles. C'est comme un jeu où l'on choisit une action en fonction d'une situation donnée.

La Structure de when

Voici comment utiliser when dans votre code :

```
when (valeur) {
    option1 -> {
        // Code à exécuter si valeur est égale à option1
    }
    option2 -> {
        // Code à exécuter si valeur est égale à option2
    }
    else -> {
        // Code à exécuter si valeur ne correspond à aucune option
    }
}
```

Exemple simple

```
val jour = "Lundi"

setContent {
    when (jour) {
        "Lundi" -> Text( text: "C'est le début de la semaine !")
        "Vendredi" -> Text( text: "C'est presque le week-end !")
        "Dimanche" -> Text( text: "Profitez de votre dimanche !")
        else -> Text( text: "Un jour comme les autres.")
    }
}
```

Dans cet exemple, le programme affiche un message différent selon le jour de la semaine.

Explication du else

Le mot-clé else est utilisé comme une option de secours dans un when. Il s'exécute si aucune des autres conditions spécifiées n'est remplie.

Quand ne pas utiliser else

Vous pouvez omettre else si vous êtes certain que toutes les possibilités sont couvertes par les conditions spécifiées.

Exemple

Imaginons que vous avez une variable jourDeSemaine qui ne peut contenir que les valeurs de 1 à 7, représentant les jours de la semaine, et vous souhaitez afficher le nom du jour correspondant.

```
val jourDeSemaine = 3
setContent {
    when (jourDeSemaine) {
        1 -> Text( text: "Lundi")
        2 -> Text( text: "Mardi")
        3 -> Text( text: "Mercredi")
        4 -> Text( text: "Jeudi")
        5 -> Text( text: "Vendredi")
        6 -> Text( text: "Samedi")
        7 -> Text( text: "Dimanche")
        // Pas de 'else' ici car nous savons que jourDeSemaine est toujours entre 1 et 7
    }
}
```

Utilisation de when pour les Variables Numériques

L'instruction when peut aussi être utilisée avec des variables numériques comme les Int ou Float. Cela permet de vérifier des plages de valeurs ou des conditions spécifiques. Voyons comment cela fonctionne.

Plages de Valeurs avec in et ..

- L'opérateur .. permet de définir une plage de valeurs. Par exemple, 1..5 représente tous les nombres de 1 à 5 inclus.
- L'opérateur in est utilisé pour vérifier si une valeur appartient à une plage. Par exemple, `value in 1..5` vérifie si value est compris entre 1 et 5.

Voici comment utiliser when avec ces concepts :

```
val temperature = 22

setContent {
    when {
        temperature < 10 -> Text( text: "Il fait très froid !")
        temperature in 10 ≤ .. ≤ 20 -> Text( text: "Il fait frais.")
        temperature in 21 ≤ .. ≤ 30 -> Text( text: "Il fait chaud.")
        else -> Text( text: "Il fait très chaud !")
    }
}
```

Combinaison de Conditions avec des Virgules

Vous pouvez combiner plusieurs conditions qui affichent le même résultat en utilisant des virgules. Cela permet d'exécuter le même bloc de code pour plusieurs valeurs différentes.
Voici un exemple :

```
val note = 85
setContent {
    when (note) {
        90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100 -> Text( text: "Excellent !")
        in 70 ≤ .. ≤ 89 -> Text( text: "Bien fait !")
        in 50 ≤ .. ≤ 69 -> Text( text: "Passable.")
        else -> Text( text: "À améliorer.")
    }
}
```

Dans cet exemple, toutes les notes de 90 à 100 affichent "Excellent !". On utilise les virgules pour combiner plusieurs valeurs dans une seule condition.

Exercices

Exercice 1

Créez une variable jour avec le nom d'un jour de la semaine (par exemple, "Lundi"). Utilisez when pour afficher un message spécifique pour chaque jour de la semaine, comme :

- "C'est lundi, début de la semaine !" pour le lundi
- "C'est vendredi, presque le week-end !" pour le vendredi
- "C'est dimanche, jour de repos !" pour le dimanche
- Sinon, affichez "C'est un jour de semaine."

Exercice 2

Créez une variable temperature de type Int (par exemple, 18). Utilisez when pour afficher un message en fonction de la température :

- "Il fait très froid !" si la température est inférieure à 10°C
- "Il fait frais." si la température est entre 10°C et 20°C
- "Il fait chaud." si la température est entre 21°C et 30°C
- "Il fait très chaud !" si la température est supérieure à 30°C

Exercice 3

Créez une variable typeProduit de type String (par exemple, "Smartphone"). Utilisez when pour afficher un message en fonction du type de produit :

- "Appareil mobile." pour "Smartphone", "Tablette", "Montre connectée"
- "Appareil informatique." pour "Ordinateur portable", "Moniteur", "Clavier"
- "Appareil audio." pour "Casque", "Enceinte", "Microphone"
- "Autre appareil." pour tout autre produit

Exercice 4

Créez une variable prix de type Float (par exemple, 45.75). Utilisez when pour afficher un message en fonction du prix de l'article :

- "Pas cher." si le prix est inférieur à 20€
- "Modéré." si le prix est entre 20€ et 50€
- "Cher." si le prix est supérieur à 50€