

LES BOUCLES DANS LE LANGAGE DE PROGRAMMATION GO

Description

Une boucle est un moyen d'**exécuter un bloc de code plusieurs fois**, suivant le résultat d'une condition ou selon un nombre d'**itérations** connu à l'avance.

Information

Une itération correspond à un tour de boucle

Boucle for

Pour déclarer une boucle il suffit d'utiliser le mot-clé **for**.

Il existe deux types de boucle **for**, soit vous connaissez au préalable le nombre d'itérations (par exemple vous savez dès le début qu'un bloc de code va s'exécuter x fois) ou inversement vous n'avez aucune idée du nombre d'itérations de votre bloc de code.

Vous connaissez déjà le nombre d'itérations

Voici comment on déclare une boucle quand vous connaissez antérieurement le nombre d'itérations :

```
for initialisation ; condition ; itération {  
    /* le code qui sera répété */  
}
```

Où **initialisation** est une instruction exécutée avant le premier parcours de la boucle **for** c'est là-bas qu'on initialise le **compteur** de la boucle (le compteur est une variable temporaire qui est détruite à la sortie de la boucle) suivit d'une **condition** du compteur dont la valeur déterminera la fin de la boucle, l' **itération** est l'opération qui déterminera le nombre de répétitions de votre bloc de code.

Imaginez le scénario suivant : "Vous êtes jeune, fainéant et mauvais élève (pas bien !). Votre professeur vous demande alors d'écrire 100 fois la phrase suivante *"Je ne dois frapper mes camarades de classe"*. ça tombe vous êtes en train de suivre un cours sur les boucles !" !"

```
package main

import (
    "fmt"
)

func main() {
    for compteur := 0; compteur < 100; compteur++ {
        fmt.Println(compteur + 1, " ) Je ne dois frapper mes camarades de classe")
    }
}
```

Résultat :

```
1 ) Je ne dois taper mes camarades de classe
2 ) Je ne dois taper mes camarades de classe
...
99 ) Je ne dois taper mes camarades de classe
100 ) Je ne dois taper mes camarades de classe
```

[Vous ne connaissez pas le nombre d'itérations](#)

Voici comment on déclare une boucle quand vous ne connaissez pas à l'avance le nombre d'itérations :

```
for condition {
    /* le code qui sera répété */
}
```

Si la **condition** est vraie alors on reste dans la boucle sinon on quitte la boucle.

Imaginez le scénario suivant : "Un videur de boîte de nuit refuse l'entrée des mineurs"

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
)

func main() {
    scanner := bufio.NewScanner(os.Stdin)
    var age int

    for age < 18 { // on quitte la boucle s'il est majeur
        fmt.Print("Entrez votre age : ")
        scanner.Scan()
        age, _ = strconv.Atoi(scanner.Text())
    }

    fmt.Println("Bienvenue en boite de nuit !")
}
```

Résultat :

```
Entrez votre age : 17
Entrez votre age : 19
Bienvenue en boite de nuit
```

break et continue

Pour manipuler vos boucles infinies il peut être intéressant d'utiliser les deux mots-clés suivants :

- **break** : Interrompre une boucle
- **continue** : Revenir au début de la boucle

Imaginons le scénario suivant : "Vous avez bien envie de jouer à un jeu hasard mais le problème c'est que vous n'avez pas assez d'argent par contre vous avez votre compilateur go ."

Vous n'avez pas d'argent ? Pas de panique les boucles infinies vont vous aider à sortir de cette crise !

```
package main

import (
    "bufio"
    "fmt"
    "math/rand"
    "os"
    "strconv"
    "time"
)

func main() {

    rand.Seed(time.Now().UnixNano())
    randomInt := rand.Intn(10)

    scanner := bufio.NewScanner(os.Stdin)

    max := 20

    for true { // boucle infinie
        fmt.Print("Entrez votre nombre : ")
        scanner.Scan()
        nbr, err := strconv.Atoi(scanner.Text())

        if err != nil {
            fmt.Println("Entrez un nombre !")
            continue // on revient au début de la boucle
        }
        if nbr > max || nbr < 0 {
            fmt.Println("Votre nombre doit être compris entre 0 et ", max, " !")
            continue
        } else if nbr == randomInt {
            fmt.Println("Bien joué !")
            break // on quitte la boucle
        }
    }
}
```

```
    } else {  
        fmt.Println("Dommage !")  
    }  
}  
}
```

Résultat :

```
Entrez votre nombre : 50  
Votre nombre doit être compris entre 0 et 20 !  
Entrez votre nombre : arnaque  
Entrez un nombre !  
Entrez votre nombre : 6  
Dommage !  
Entrez votre nombre : 15  
Dommage !  
Entrez votre nombre : 3  
Bien joué !
```