

# Structures de contrôle

## Exercice 1

Commençons simplement. Affichez les résultats attendus en fonction de la variable `age`.

```
package main

func main() {
    // Modifiez ceci en conséquence pour produire les résultats attendus
    // age := 10

    // Saisissez votre instruction if ici.
}
```

### Résultat attendu

```
Si l'âge est supérieur à 60, affichez :
    Getting older
Si l'âge est supérieur à 30, affichez :
    Getting wiser
Si l'âge est supérieur à 20, affichez :
    Adulthood
Si l'âge est supérieur à 10, affichez :
    Young blood
Sinon, affichez :
    Booting up
```

## Exercice 2

Pouvez-vous simplifier l'instruction `if` dans le code ci-dessous ?

Lorsque :

- `isSphere` est vrai (true)
- Le rayon ( `radius` ) est égal ou supérieur à 200

Il devrait afficher "C'est une grosse sphère."

Sinon, il devrait afficher "Je ne sais pas."

```
package main

import "fmt"

func main() {
    // NE MODIFIEZ PAS CE CODE
    isSphere, radius := true, 200

    var big bool

    // Simplifiez cette structure if imbriquée
    if radius >= 200 {
        big = true
    }
```

```

    if !big {
        fmt.Println("Je ne sais pas.")
    } else if isSphere {
        fmt.Println("C'est une grosse sphère.")
    } else {
        fmt.Println("Je ne sais pas.")
    }
}

```

### Résultat attendu

C'est une grosse sphère.

## Exercice 3

Refactorisez le code suivant pour utiliser une instruction `switch` à la place de `if / else`.

"Impression du nombre de jours dans un mois donné."

Indice : Pour utiliser les arguments en Go, consultez les documentations :

- <https://gobyexample.com/command-line-arguments>
- <https://yourbasic.org/golang/command-line-arguments/>

```

package main

import (
    "fmt"
    "os"
    "strings"
    "time"
)

func main() {
    if len(os.Args) != 2 {
        fmt.Println("Donnez-moi un nom de mois")
        return
    }

    year := time.Now().Year()
    leap := year%4 == 0 && (year%100 != 0 || year%400 == 0)

    days, month := 28, os.Args[1]

    if m := strings.ToLower(month); m == "avril" ||
        m == "juin" ||
        m == "septembre" ||
        m == "novembre" {
        days = 30
    } else if m == "janvier" ||
        m == "mars" ||
        m == "mai" ||
        m == "juillet" ||
        m == "août" ||
        m == "octobre" ||
        m == "decembre" {

```

```

        days = 31
    } else if m == "fevrier" {
        if leap {
            days = 29
        }
    } else {
        fmt.Printf("%q n'est pas un mois valide.\n", month)
        return
    }

    fmt.Printf("%q a %d jours.\n", month, days)
}

```

## Exercice 4

1. À l'aide d'une boucle, additionnez les nombres de 1 à 10.
2. Affichez la somme.

```

package main

import "fmt"

func main() {
    // Votre code ici
}

```

## Exercice 5

1. Étendez l'exercice "Somme jusqu'à 10".
2. Affichez les chiffres additionnés.
3. Additionnez uniquement les nombres pairs.
  - Ignorez les nombres impairs en utilisant l'instruction `continue`.

```

package main

import "fmt"

func main() {
    // Votre code ici
}

```

### Résultat attendu

```
2 + 4 + 6 + 8 + 10 = 30
```