

# Tests & Boucles

## Les combinaisons et opérateurs booléens

### Comparaisons

- `==`
- `!=`
- `<`
- `>`
- `>=`
- `<=`

### Opérateurs booléens

- `&&`
- `||`
- `!`

## If

```
age := 10
if age > 10 {
  // something
}
```

## Conditions alternatives

```
age := 10
if age > 10 {
  // something
}
```

```
} else if a > 5 {  
    // something else  
} else {  
    // something else else  
}
```

# Switch

Hé il n'y a pas de break dans les switch en go

```
age := 10  
switch age {  
case 10:  
    // code cas 10  
case 5 :  
    // code cas 5  
default :  
    // code par défaut (facultatif)  
}
```

On peut cumuler les valeurs au sein d'un même `case` !

```
age := 10  
switch age {  
    case 10,8,6:  
        // code cas 10 ou 8 ou 6  
    case 5 :  
        // code cas 5  
    default :  
        // code par défaut (facultatif)  
}
```

Et même mieux encore...

```
hour := 10  
fmt.Printf("Il est %dh\n", hour)  
switch {  
case hour < 12:
```

```
fmt.Println("C'est le matin !")
case hour > 12 && hour < 18:
fmt.Println("C'est l'après midi")
default:
fmt.Println("On est le soir")
}
```

# While

Attention !

Le `while` n'existe pas en Go ! Haha !

# For

Le `for` en Go est très évolué et permet de remplacer l'utilisation du `while`.

```
sum := 0
for i := 0; i < 10; i++ {
sum += i
}
fmt.Println(sum)
```

# Remplacement du while

```
n := 1
for n < 5 {
n *= 2
}
fmt.Println(n)
```

Les instructions `continue` et `break` sont aussi utilisables en Go

# Boucle infinie

```
for {
fmt.Println("Boucle infinie")
}
```

}

---

Revision #2

Created 2022-05-09 19:04:13 UTC by Noé Larrieu-Lacoste

Updated 2022-05-09 19:06:49 UTC by Noé Larrieu-Lacoste