

# LES SLICES (TABLEAUX DYNAMIQUES) DANS LE LANGAGE DE PROGRAMMATION GO

## Pourquoi les Slices ?

---

Le problème avec un tableau c'est qu'il a une taille fixe, il faut donc absolument connaître sa taille au moment de sa déclaration ce qui n'est pas vraiment évident car on peut très vite s'apercevoir plus tard dans notre code que la taille allouée au départ à notre tableau est insuffisante.

Et c'est là qu'interviennent les Slices dans Go, Ils vont nous permettre d'avoir un **tableau flexible** et le **dimensionner** de façon **dynamique** sans se soucier de sa taille pendant sa déclaration.

## Déclarer une Slice

---

Il existe deux façons pour créer une Slice.

Soit à partir de la même syntaxe qu'un tableau sauf que cette fois-ci on ne spécifie pas la taille du tableau :

```
package main

import (
    "fmt"
)

func main() {
    var nombres = []int{0, 0, 0, 0, 0} // création d'une slice avec 5 éléments
    fmt.Println(nombres)
}
```

## Résultat :

```
[0 0 0 0 0]
```

Soit depuis la fonction `make()`

```
package main

import (
    "fmt"
)

func main() {
    var nombres = make([]int, 5) // création d'une slice avec 5 éléments
    fmt.Println(nombres)
}
```

## Résultat :

```
[0 0 0 0 0]
```

## Rajouter un élément dans une Slice

---

Pour rajouter un élément dans votre slice il faut utiliser la fonction `append()`, qui prend comme paramètres d'abord votre slice et ensuite l'élément que vous voulez rajouter et elle vous retournera une nouvelle Slice avec l'élément rajouté.

```
package main

import "fmt"

func main() {
    var mois []string

    mois = append(mois, "Janvier")
    fmt.Println(mois)

    mois = append(mois, "Février")
    fmt.Println(mois)
}
```

## Résultat :

```
[Janvier]  
[Janvier Février]
```

## Supprimer un élément dans une Slice

---

Il n'existe pas spécialement de fonction dans Go qui permet de supprimer un élément, mais il est possible de réutiliser la fonction `append()` pour supprimer un élément en Go.

Avant de vous montrer comment faire pour supprimer n'importe quel élément d'une slice, laissez-moi avant vous expliquer comment fonctionne la fonction `append()`.

Voici à quoi ressemble le prototype de la fonction `append()` :

```
func append(slice []Type, elems ...Type) []Type
```

La fonction `append()` prend comme premier argument une Slice, le second paramètre est quant à lui est un **paramètre variadique** c'est-à-dire qu'il prend en compte un nombre indéterminé de paramètres.

Comme il est possible de rajouter un nombre indéterminé de paramètres sur le deuxième paramètre de la fonction `append()`, alors pour supprimer un élément il suffit de **concaténer** les éléments qui sont avant l'élément qu'on souhaite supprimer dans notre slice avec les éléments qui sont après l'élément qu'on souhaite supprimer dans notre slice.

Ce qui nous donne :

```
package main

import "fmt"

func main() {
    mois := []string{"Janvier", "Février", "Mars", "Avril", "Juin", "Juillet"}

    fmt.Println(mois)

    // index à supprimer de notre slice
    indexASupprimer := 1

    // Suppression de l'index 1 du tableau soit "Février"
    mois = append(mois[:indexASupprimer], mois[(indexASupprimer+1):]...)

    fmt.Println(mois)
}
```

### Résultat :

```
[Janvier Février Mars Avril Juin Juillet]
[Janvier Mars Avril Juin Juillet]
```

## Copier le contenu d'une Slice

---

Il est possible de copier le contenu d'une slice source vers une slice cible grâce à la fonction `copy()`. Cette fonction prend comme premier paramètre la slice cible et comme deuxième paramètre la slice source.

### Attention

La fonction `copy()` copie le contenu d'une tranche source vers une source cible, il est donc important que la slice cible soit de même taille que la slice source afin de copier le contenu total de la cible.

```
package main
```

```
import "fmt"

func main() {

    animaux1 := []string{"Lion", "Cheval", "Ours"}
    fmt.Println("Contenu du tableau animaux1 :", animaux1)

    // création d'une slice cible avec la même taille que la slice source
    animaux2 := make([]string, len(animaux1))

    // copie du contenu de la slice source vers la slice cible
    copy(animaux2, animaux1)

    fmt.Println("Contenu du tableau animaux2 :", animaux2)
}
```

### Résultat :

```
Contenu du tableau animaux1 : [Lion Cheval Ours]
Contenu du tableau animaux2 : [Lion Cheval Ours]
```