

# LES MAPS DANS LE LANGAGE DE PROGRAMMATION GO

## Pourquoi les Maps ?

---

Go fournit un autre type de données important nommé Map, qui associe des **clés uniques** à des valeurs.

Avant pour récupérer un élément précis dans votre tableau vous utilisiez un index, les Maps possèdent le même processus de récupération d'éléments qu'un tableau cependant pour les Maps votre index est nommé clé et votre clé peut être de n'importe quel type (int, string, float64 ...) à l'instar des tableaux où l'index est obligatoirement de type int.

Par exemple on peut créer une Map avec comme clé les noms des élèves et comme valeur leurs notes et intercepter la note d'un élève précis (la valeur) en utilisant son nom (la clé)

## Déclarer une Map

---

Comme les Slices, il existe deux façons pour créer une Map.

Première méthode :

```
package main

import (
    "fmt"
)

func main() {
    var notes map[string]int
}
```

Deuxième méthode depuis la fonction `make()`

```
package main

import (
    "fmt"
)

func main() {
    var notes = make(map[string]int)
}
```

Comme dans d'autres types de variables il est aussi possible de surcharger les valeurs par défaut de votre Map.

```
package main

import "fmt"

func main() {
    notes := map[string]int{"Hatim": 20, "Alex": 18}
    fmt.Println(notes)
}
```

**Résultat :**

```
map[Alex:18 Hatim:20]
```

## Rajouter un élément dans une Map

---

Voici comment on rajoute un élément dans une Map.

```
package main

import "fmt"

func main() {
    var notes = make(map[string]int)
    notes["Hatim"] = 20
    notes["Alex"] = 18
}
```

```
notes["Kevin"] = 15

fmt.Println(notes)
}
```

**Résultat :**

```
map[Alex:18 Hatim:20 Kevin:15]
```

## Récupérer les éléments d'une Map

---

### Récupérer un élément précis

Comme expliqué sur l'introduction de ce chapitre, on récupère un élément d'une Map grâce à sa clé, donc on utilisera la clé de l'élément de notre Map pour récupérer sa valeur.

```
package main

import "fmt"

func main() {
    notes := map[string]int{"Hatim": 20, "Alex": 18}

    fmt.Println("La note de Hatim est :", notes["Hatim"])
    fmt.Println("La note de Alex est :", notes["Alex"])
}
```

**Résultat :**

```
La note de Hatim est : 20
La note de Alex est : 18
```

### Boucle for dans une map

On peut utiliser la boucle `for` avec le mot-clé `range` pour récupérer la clé de tous les éléments de votre Map.

```
package main

import "fmt"

func main() {
    notes := map[string]int{"Hatim": 20, "Alex": 18, "Kevin": 15, "Robert": 17}

    for eleve := range notes {
        fmt.Println("La note de ", eleve, "est", notes[eleve])
    }
}
```

**Résultat :**

```
La note de  Hatim est 20
La note de  Alex est 18
La note de  Kevin est 15
La note de  Robert est 17
```

## Supprimer un élément dans une Map

---

Pour supprimer un élément de votre Map il faut utiliser la fonction `delete()`, qui prend comme paramètres d'abord votre Map et ensuite la clé de l'élément que vous voulez supprimer.

```
package main

import "fmt"

func main() {
    notes := map[string]int{"Hatim": 20, "Alex": 18, "Kevin": 15, "Robert": 17}
    fmt.Println(notes)

    delete(notes, "Hatim")
    fmt.Println(notes)
}
```

**Résultat :**

```
map[Alex:18 Hatim:20 Kevin:15 Robert:17]  
map[Alex:18 Kevin:15 Robert:17]
```