

Строки, руны, мапы шпаргалка

Строки

Байты

Байт - это минимальная адресуемая единица памяти в компьютере. Байты используются для хранения информации, такой как числа, символы и другие данные.

Массив байтов

Массив байтов - это последовательность байтов фиксированного размера.

```
var buffer [256]byte
```

Строки

Строка - это неизменяемая последовательность байтов переменной длины. Строки в Go представляются типом `string`.

```
var s string = "hello"
```

Руны и преобразования

В Go, строки представляются в формате UTF-8. Если вы хотите перевести строку в байты и обратно, используйте функции `[]byte` и `string`.

```
s := "Привет, мир!"  
b := []byte(s) // Преобразование строки в байты  
s2 := string(b) // Преобразование байтов в строку
```

Когда вы работаете со строками, часто нужно обрабатывать символы Unicode. Для этого в Go есть тип `rune`, который представляет собой Unicode-символ.

```
s := "Привет, мир!"  
for _, r := range s {  
    fmt.Printf("%c", r)
```

```
}  
// Вывод: Привет, мир!
```

Мар в Golang

Мар - это структура данных, которая хранит отображение между ключами и значениями. Ключи и значения могут быть любого типа, при условии, что ключи можно сравнивать оператором `==`.

Создание Мар

Для создания мар используется функция `make`, которая возвращает ссылку на новый мар. Пример:

```
var m map[string]int  
m = make(map[string]int)
```

Также можно создать мар в одну строку:

```
m2 := map[string]int{"foo": 42, "bar": 123}
```

Добавление и удаление элементов

Для добавления элементов в мар используется оператор `[]`. Пример:

```
m["foo"] = 42
```

Для удаления элемента из мар используется функция `delete`. Пример:

```
delete(m, "foo")
```

Получение значения по ключу

Для получения значения по ключу используется оператор `[]`, а также второе значение, которое указывает, найден ли элемент в мар. Пример:

```
value, ok := m["foo"]  
if ok {  
    fmt.Println(value)  
}
```

Итерация по Map

Для итерации по map используется оператор `range`. Пример:

```
for key, value := range m2 {  
    fmt.Println(key, value)  
}
```