

# Условные конструкции

## шпаргалка

### Условные конструкции

Условия объявляются с помощью ключевого слова `if`.

```
if x < 0 {  
    fmt.Println(sqrt(-x) + "i")  
}
```

### Else if, else

```
if score >= 60 {  
    fmt.Println("Отличная работа!")  
} else {  
    fmt.Println("Нужно усерднее поучиться!")  
}
```

`Else` объявляется сразу после закрывающей скобки условия `if`.

Для обозначения нескольких условий мы можем использовать конструкцию `else if`.

```
if age < 18 {  
    fmt.Println("Вы несовершеннолетний")  
} else if age < 65 {  
    fmt.Println("Вы взрослый")  
} else {  
    fmt.Println("Вы очень взрослый")  
}
```

### Операторы сравнения

<code>==</code>	равно
<code>!=</code>	не равно
<code>&lt;</code>	меньше
<code>&lt;=</code>	меньше или равно
<code>&gt;</code>	больше

>=	больше или равно
----	------------------

## Приоритет операций

Сверху вниз по убыванию приоритета

Конечно! Вот таблица с уровнями приоритетов основных операторов в Go:

Уровень	Операции
1	*, /, %, <<, >>, &, &^
2	+, -,
3	==, !=, <, <=, >, >=
4	&&

## Короткое объявление условных конструкций в Go

Кроме полной записи `if`, в Go есть “короткая запись условных конструкций”.

```
v := math.Sqrt(x)
if v < 0 {
    fmt.Println(sqrt(-x) + "i")
}
```

можно записать как

```
if v := math.Sqrt(x); v < 0 {
    fmt.Println(sqrt(-x) + "i")
}
```

В этом случае переменная `v` объявляется только внутри условия `if` и будет доступна только внутри этого блока.

Можно использовать несколько утверждений до проверки условия.

```
if err := file.Chmod(0664); err != nil && file.Name == "something" {
    log.Print(err)
}
```

## Switch statement

Switch в Go позволяет легко проверять значения переменных и выполнять соответствующие действия в зависимости от этих значений.

Синтаксис switch statement в Go:

```
switch variable {  
    case value1:  
        // выполняемые действия, если variable равна value1  
    case value2:  
        // выполняемые действия, если variable равна value2  
    ...  
    default:  
        // выполняемые действия, если variable не равна ни одному из значений  
}
```

Пример использования:

```
package main  
  
import "fmt"  
  
func main() {  
    i := 2  
    switch i {  
        case 1:  
            fmt.Println("i равно 1")  
        case 2:  
            fmt.Println("i равно 2")  
        default:  
            fmt.Println("i не равно ни 1, ни 2")  
    }  
}
```