



Chapter 9 if문

📅 날짜	@November 10, 2021
👤 발표자	장에서

9.1 if문 기본 사용법

if문은 조건에 따라 분기하는 구문.

```
if 조건문 {  
    문장  
} else if 조건문 {  
    문장  
} else {  
    문장  
}
```

만족하는 조건문의 {} 안에 있는 문장을 실행, 없으면 else 구문 {} 안에 있는 문장 실행. else if 와 else 구문은 생략 가능

```
package main  
  
import "fmt"  
  
func main() {  
    light := "red"  
  
    if light == "green" {  
        fmt.Println("길을 건넌다")  
    } else {  
        fmt.Println("대기한다")  
    }  
}
```

출력문

대기한다

```
package main

import "fmt"

func main() {
    temp := 33

    if temp > 28 {
        fmt.Println("에어컨을 켜다")
    } else if temp <= 3 {
        fmt.Println("히터를 켜다")
    } else {
        fmt.Println("대기한다")
    }
}
```

출력문

에어컨을 켜다

9.2 그리고 &&, 또는 ||

논리연산자 &&, ||

```
package main

import "fmt"

func main() {
    var age = 22

    if age >= 10 && age <= 15 {
        // age가 10 이상 15 이하인 경우
        fmt.Println("You are young")
    } else if age > 30 || age < 20 {
        // age가 30보다 크거나 20보다 작은 경우. 즉 20대가 아닌 경우
        fmt.Println("You are not 20s")
    }
}
```

```

    } else {
        fmt.Println("Best age of your life")
    }
}

```

출력문

```
Best age of your life
```

9.2.1 쇼트서킷

- 쇼트서킷: && 연산은 좌변이 false이면 우변을 검사하지 않고 false 처리. || 연산도 좌변이 true이면 우변을 검사하지 않고 true 처리.

뜻하지 않은 결과를 얻는 예제

```

package main

import "fmt"

var cnt int = 0

func IncreaseAndReturn() int {
    fmt.Println("IncreaseAndReturn()", cnt)
    cnt++
    return cnt
}

func main() {
    if false && IncreaseAndReturn() < 5 { //함수 호출 X
        fmt.Println("1 increase")
    }
    if true || IncreaseAndReturn() < 5 { //함수 호출 X
        fmt.Println("2 increase")
    }

    fmt.Println("cnt:", cnt)
}

```

출력문

```
2 increase
cnt: 0
```

- 쇼트서킷에 의해서 IncreaseAndReturn() 함수가 호출되지 않음
- && 좌변 false → 우변 함수 호출 x → "1 증가" 출력 x
- || 좌변 true → 우변 함수 호출 x → "2 증가" 출력
- 함수는 한 번도 호출 x → cnt은 0으로 지속
- &&, ||의 특징 때문에 if문 조건에 들어가는 함수는 조건 검사만 하고 다른 로직은 실행하지 않는 게 안전.

9.2.2 소괄호 () 활용

```
if hasBooked() || (money > 200 && hasEmptySeat())
```

- hasBooked() 결과가 true 이거나 hasBooked() 결과가 false여도 money가 200보다 크고 hasEmptySeat() 결과가 true이면 조건은 true.

9.3 중첩 if

```
package main

import "fmt"

// 친구 중 부자가 있는가 반환 - 무조건 true 반환
func HasRichFriend() bool {
    return true
}

// 같이 간 친구 숫자를 반환 - 무조건 3을 반환
func GetFriendsCount() int {
    return 3
}

func main() {
    price := 35000

    if price > 50000 {
        if HasRichFriend() {
            fmt.Println("앗 신발끈이 풀렸네")
        } else {
            fmt.Println("나눠내자")
        }
    }
    } else if price >= 30000 && price <= 50000 {
        if GetFriendsCount() > 3 {
            fmt.Println("어이쿠 신발끈이..")
        } else {
```

```

        fmt.Println("사람도 얼마 없는데 나눠내자")
    }
} else {
    fmt.Println("오늘은 내가 쓴다")
}
}

```

출력문

```

사람도 얼마 없는데 나눠내자

```

- 오만원이 넘을 때 또 다시 판단해야하는 조건 → if문 중첩
- if문은 계속 중첩할 수 있지만 중첩이 심할 경우 코드를 이해하기 힘들기 때문에 **3중첩 이상은 하지 않도록 권장!**

9.4 if 초기문; 조건문

if문 조건을 검사하기 전에 초기문을 넣을 수 있음. 주로 검사에 사용할 변수를 초기화할 때 사용.

```

if 초기문; 조건문 {
    문장
}

```

```

if filename, success := UploadFile(); success {
    fmt.Println("Upload success", filename)
} else {
    fmt.Println("Failed to upload")
}

```

먼저 UploadFile() 함수 실행, filename과 success 변수에 반환값 저장. 성공 여부에 따라 다른 메세지 출력. if 초기문은 어떤 함수 실행 후 결과 검사할 때 주로 사용.

초기문에서 선언한 변수의 범위는 if문 안으로 한정된다는 사실에 주의해야함.

```

package main

import "fmt"

```

```

func getMyAge() (int, bool) {
    return 33, true
}

func main() {

    if age, ok := getMyAge(); ok && age < 20 {
        fmt.Println("You are young", age)
    } else if ok && age < 30 {
        fmt.Println("Nice age", age) // age 값에 접근가능
    } else if ok {
        fmt.Println("You are beautiful", age)
    } else {
        fmt.Println("Error")
    }

    fmt.Println("Your age is", age) // Error - age는 소멸됨.
}

```

출력문

```

.\ex9.6.go:21:29: undefined: age

```

- 일반적으로 변수는 변수가 선언된 중괄호를 벗어나면 소멸되지만 if 초기문에 선언된 변수는 if문 종료되기 전까지 유지됨.

연습문제

1.

```

package main

import "fmt"

func main() {
    age := 22

    if age < 10 {
        fmt.Println("You are a child")
    } else if age >= 20 && age < 30 {
        fmt.Println("Best age of your life")
    } else {
        fmt.Println("You are beautiful")
    }
}

```

2.

신중히 생각해보세요

3.

```
package main

import "fmt"

func main() {
    temp := 30
    rain := 40

    if temp >= 25 {
        if rain >= 80 {
            fmt.Println("덥고 비가 옵니다.")
        } else if rain >= 20 {
            fmt.Println("덥고 습합니다.")
        } else {
            fmt.Println("야외 활동하기 좋습니다.")
        }
    } else if temp < 10 || rain >= 80 {
        fmt.Println("야외 활동하기 좋지 않습니다.")
    } else {
        fmt.Println("좋은 날씨입니다.")
    }
}
```