Conditionals

Created	@Jan 28, 2021 12:26 AM
Tags	Lesson

The `if` Statement

```
alarmRinging := true
if alarmRinging {
  fmt.Println("Turn off the alarm!!")
}
```

The `else` Statement

```
isHungry := false
if isHungry {
  fmt.Println("Eat the cookie")
} else {
  fmt.Println("Step away from the cookie...")
}
```

Comparison Operators

- `==` Is equal to
- `!=` Is NOT equal to
- `<` Less than
- `>` Greater than
- `<=` Less than or equal to
- `>=` Greater than or equal to

Logical Operators

- `&&` And

- `||` Or
- `!` Not

The `else if` Statement

```
amountStolen := 64650

if amountStolen > 1000000 {
  fmt.Println("We've hit the jackpot!")
} else if amountStolen >= 5000{
  fmt.Println("Think of all the candy we can buy!")
} else {
  fmt.Println("Why did we even do this?")
}
```

The `switch` Statement

```
clothingChoice := "sweater"

switch clothingChoice {
  case "shirt":
    fmt.Println("We have shirts in S and M only.")
  case "polos":
    fmt.Println("We have polos in M, L, and XL.")
  case "sweater":
    fmt.Println("We have sweaters in S, M, L, and XL.")
  case "jackets":
    fmt.Println("We have jackets in all sizes.")
  default:
    fmt.Println("Sorry, we don't carry that")
}
```

Scoped Short Declaration Statement

```
x := 8
y := 9
if product := x * y; product > 60 {
  fmt.Println(product, "  is greater than 60")
}

//product cannot be accessed here
```

*Only scoped within the conditional

Randomizing

- By importing `math/rand` you can have access to the `rand.Intn()` method
- `randIntn(100)` will output an integer from 0 to 100
- Use `rand.Seed()` to generate new random values each time
  - It's common to make seed with the current time: `rand.Seed(time.Now().UnixNano())`