Solution Review: Season of a Month

This lesson discusses the solution to the challenge given in the previous lesson.

WE'LL COVER THE FOLLOWING

- Using `switch-case` construct
- Using `if-else` construct

There are two methods to solve this problem:

- One is using the `switch-case` construct, which is the requirement of the problem statement
- The other is using the `if-else` construct.

We'll first discuss the `switch-case` solution.

Using `switch-case` construct

```
package main

import "fmt"

func main() {
        fmt.Printf(Season(3))     // calling function to find the season
}

func Season(month int) string {
        switch month {  // switch on the basis of value of the months(1-12)
                case 12,1,2:  return "Winter"  // Jan, Feb and Dec have winter
                case 3,4,5:       return "Spring"  // March, Apr and May have spring
                case 6,7,8:   return "Summer"  // June, July and Aug have summer
                case 9,10,11: return "Autumn"  // Sept, Oct and Nov have autumn

                default: return "Season unknown"         //value outside [1,12], then season
        }
}
```

Season of a Month(switch-case)

As you can see, we declare a function `Season` at **line 9**, which takes an integer value that represents a month as an input parameter. That parameter is the value of `month`. As per requirement, we add `switch month` to switch on the basis of the value of the month. There are **five** cases in total along with the default case.

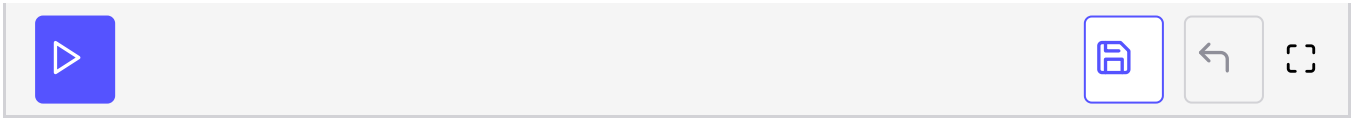
- We make the first case at **line 11** for the season of **Winter**. We return `Winter` for months of **Jan, Feb, and Dec** by using values **1**,**2**, and **12** respectively.
- We make the second case at **line 12** for the season of **Spring**. We return `Spring` for months of **Mar, Apr, and May** by using values **3**,**4**, and **5**, respectively.
- We make the third case at **line 13** for the season of **Summer**. We return `Summer` for months of **June, July, and Aug** by using values **6**,7, and **8**, respectively.
- We make the fourth case at **line 14** for the season of **Autumn**. We return `Autumn` for months of **Sept, Oct, and Nov** by using values **9**,**10**, and **11** respectively.
- Up till now, we have written 4 cases that cover all the seasons. But still, we need one case more. What if the user enters the wrong value for the month, less than 1 or greater than 12? In such a scenario, we have to have a *default* case: `return "Season Unknown"`. If the user inputs an invalid value for the month, the season should be unknown.

The `main` function is at **line 5**, in which we have called the `Season` function to view the results. We pass **3** as a parameter to the function, which means the *second* case will be executed and **Spring** will be printed on the screen.

Using `if-else` construct

```
package main

import "fmt"

func main() {
        fmt.Printf(Season(3))     // calling function to find the season
```

```
}

func Season(month int) string {

                if (month == 12) || (month == 1) || (month ==2) {  // Jan, Feb and Dec have w
                        return "Winter"
                }else if (month == 3) || (month == 4) || (month ==5) { // March, Apr and May
                        return "Spring"
                }else if (month == 6) || (month == 7) || (month ==8) { // June, July and Aug
                        return "Summer"
                }else if (month == 9) || (month == 10)|| (month ==11){ // Sept, Oct and Nov h
                        return "Autumn"
                }else{ //value outside [1,12], then season is unkown
                        return "Season unknown"  }

}
```



Season of a Month (if-else)

As you can see, we declare a function `Season` at **line 9**, which takes an integer value that represents a month as an input parameter. That parameter is the value of `month`. There are **five** conditions in total.

- We make the first condition at **line 11** for the season of **Winter**. We return `Winter` for months of **Jan, Feb, and Dec** by using values **1**,**2**, and **12** respectively.
- We make the second condition at **line 13** for the season of **Spring**. We return `Spring` for months of **Mar, Apr, and May** by using values **3**,**4**, and **5**, respectively.
- We make the third condition at **line 15** for the season of **Summer**. We return `Summer` for months of **June, July, and Aug** by using values **6**,**7**, and **8**, respectively.
- We make the fourth condition at **line 17** for the season of **Autumn**. We return `Autumn` for months of **Sept, Oct, and Nov** by using values **9**,**10**, and **11** respectively.
- Up till now, we have written 4 conditions that cover all the seasons, but there is still, an edge case. What if the user enters the wrong value for the month, less than 1 or greater than 12? In such a scenario, we have to have an `else` part: `return "Season Unknown"`. If the user inputs an invalid value for the month, the season should be unknown.

value for the month, the season should be unknown.

The `main` function is at **line 5**, in which we have called the `Season` function to view the results. We pass **3** as a parameter to the function, which means the *second* condition is true and **Spring** will be printed on the screen.

That's it about the solution. In the next lesson, you'll study another control construct in Go called the *for* construct.