

Loops and range

For loops

The only looping construct in Go is the for loop. There is no while loop, but it is actually just merged into the for loop.

Loop on condition

Basically while loop except you replace the while with for

```
i := 1

for i <= 3 {
    fmt.Println(i)
    i = i + 1
}
```

Normal for loops

The same for loop that you see in C, Java, or JavaScript without the parenthesis.

```
for j := 0; j < 5; j ++ {
    fmt.Println(j)
}
```

You can break on condition as well.

Range

The range keyword is used to iterate through an iterable object such as an Array or Map.

```
nums := []int{1, 2, 3, 4}
for i, num := range nums {
    fmt.Println(i)
    fmt.Println(num)
}
```

If you don't want the index then you can replace it with `_` to ignore that variable.

The `range` keyword can also be applied on Maps to iterate through the key-value pairs.

Revision #2

Created 3 September 2023 01:00:22 by Tamarine

Updated 3 September 2023 01:22:27 by Tamarine