



链滴

# Go 学习之路：Println 与 Printf 的区别

作者：[haleyrom](#)

原文链接：<https://ld246.com/article/1566786785120>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

**Println和Printf都是fmt包中公共方法；在需要打印信息时常用的数，那么二函数有什么区别呢？**

## 附上代码

```
package main

import (
    "time"
    "fmt"
)

const (
    Man = 1
    Female = 2
)

func main(){
    timer := time.Now().Unix()
    if(timer % Female == 0){
        fmt.Println("%d is Female", timer)
        fmt.Printf("%d is Female", timer)
    }else{
        fmt.Println("%d is Man", timer)
        fmt.Printf("%d is Man", timer)
    }
}
```

## 运行结果

```
%d is Man 1529049077 // println输出结果
1529049077 is Man // printf输出结果
```

## 结果可知

**Printf** : 可打印出格式化的字符串,**Println**不行;

## 稍做修改下

```
package main

import "fmt"

const (
    StrN = "123"
    IntN = 123
)
```

```
func main(){
    fmt.Println(StrN)
    fmt.Printf("%s\n",StrN)
    fmt.Printf(StrN)
    fmt.Println(IntN)
    fmt.Printf("%d\n",IntN)
    fmt.Printf(IntN)
}
```

## 结果

cannot use IntN(type int) as type string in argument to fmt.Printf

## 原因

```
// Println formats using the default formats for its operands and writes to standard output.
// Spaces are always added between operands and a newline is appended.
// It returns the number of bytes written and any write error encountered.
```

```
func Println(a ...interface{}) (n int, err error) {
    return Fprintln(os.Stdout, a...)
}
```

```
// Printf formats according to a format specifier and writes to standard output.
// It returns the number of bytes written and any write error encountered.
```

```
func Printf(format string, a ...interface{}) (n int, err error) {
    return Fprintf(os.Stdout, format, a...)
}
```

总结一句话： `println`会根据你输入格式原样输出，`printf`需要格式化输出并带输出格式；