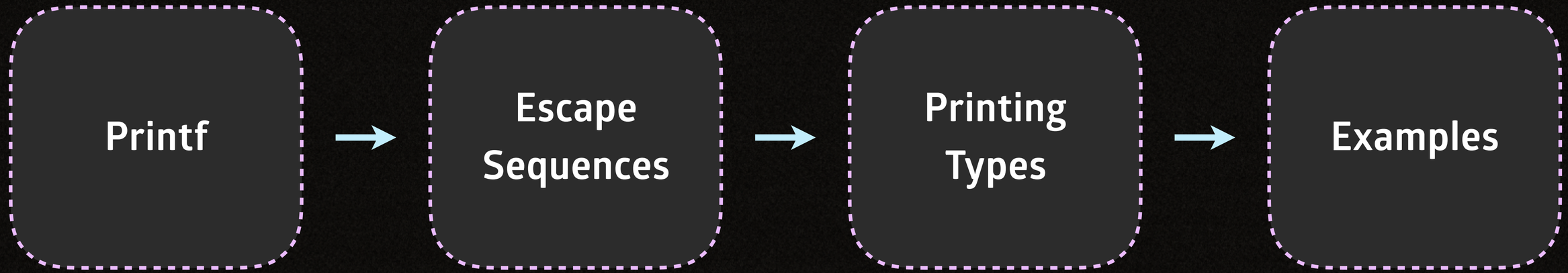
PRINTF

prints formatted output

PRINTF

You're going to learn how to use Printf



PRINTF

A little bit about Printf

```go
func main() {
  var brand string

  fmt.Printf("%q\n", brand)
}
```

PRINTF

A little bit about Printf

```go
func main() {
  var brand string

  fmt.Printf("%q\n", brand)
}
```

```
$ go run main.go
""
```

PRINTF

A little bit about Printf

```go
func main() {
  var brand = "Google"

  fmt.Printf("%q\n", brand)
}
```

```
$ go run main.go

"Google"
```

PRINTF

A little bit about Printf

```go
func main() {
  var brand = "Google"

  fmt.Printf("%s\n", brand)
}
```

```
$ go run main.go
Google
```

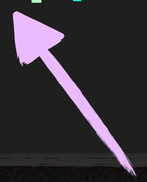
"Google"

PRINTF

A little bit about Printf

```go
func main() {
  var brand = "Google"

  fmt.Printf("%q\n", brand)
}
```



```
$ go run main.go
"Google"
```

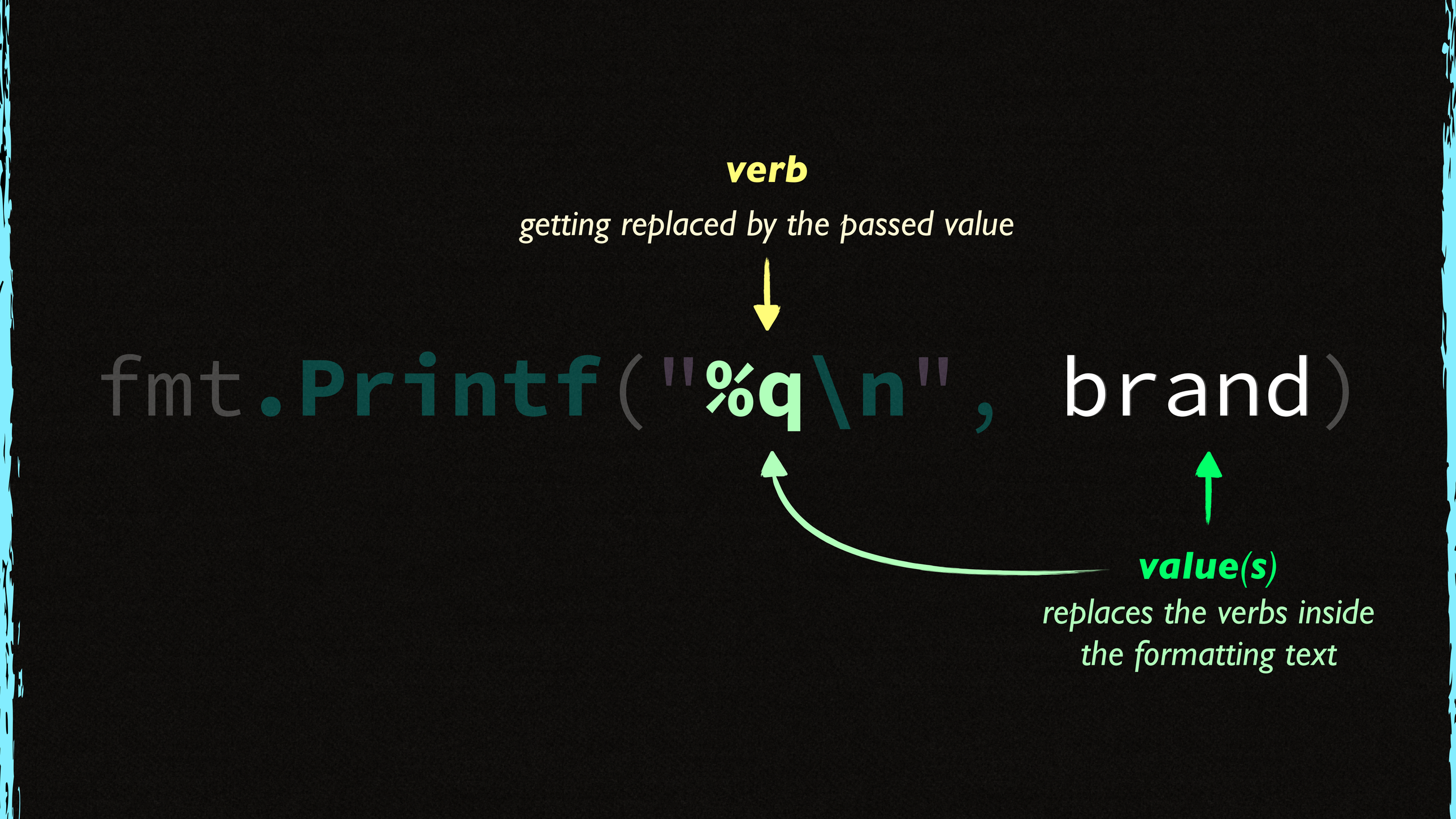
formatting text

determines "what" and "how" to print

```
fmt.Printf("%q\n", brand)
```

***replacer value**(s)*

replaces the verbs inside the formatting text


verb
getting replaced by the passed value
fmt.Printf("%q\n", brand)
value(s)
replaces the verbs inside
the formatting text

```
fmt.Printf("%q\n", brand)
```

escape sequence

prints a newline

PRINTLN vs PRINTF

```
total: 2350 success: 543 / 433
```

PRINTLN

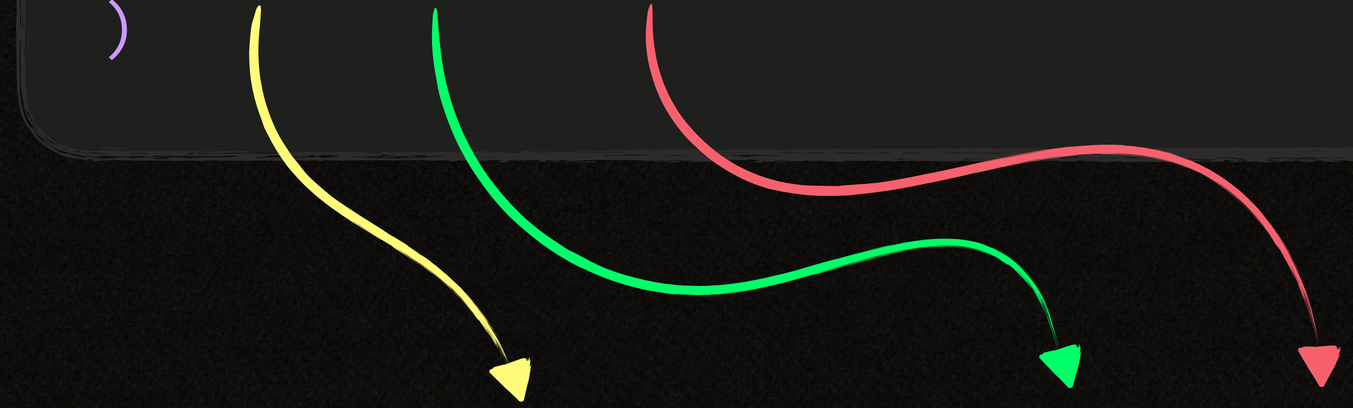
Please assume that "ops", "ok" and "fail" variables are declared already

```
fmt.Println(
  "total:", ops, "success:", ok, "/", fail,
)
```

```
total: 2350 success: 543 / 433
```

PRINTF

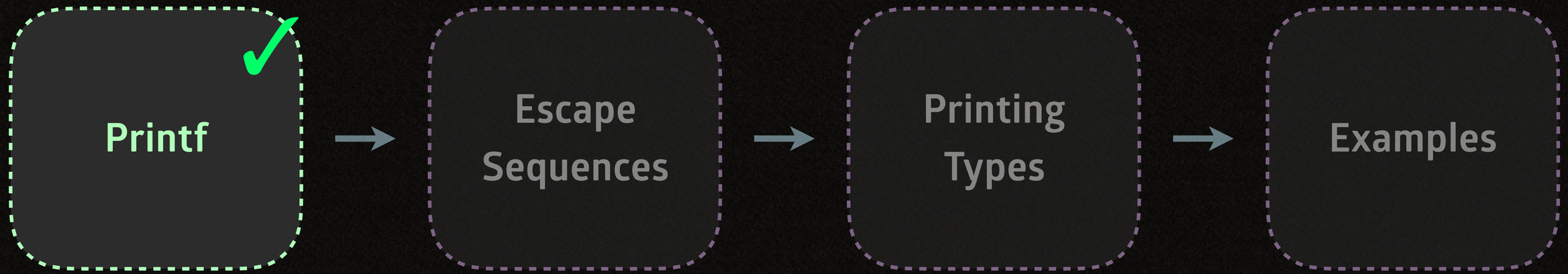
```
fmt.Printf(
   "total: %d success: %d / %d\n",
   ops, ok, fail,
)
```



```
total: 2350 success: 543 / 433
```

PRINTF

Well done!



ESCAPE SEQUENCES

Allow you to represent special characters

ESCAPE SEQUENCES

They give special meanings to characters

```go
func main() {
   fmt.Println( "hihi" )
}
```

```
$ go run main.go
hihi
```

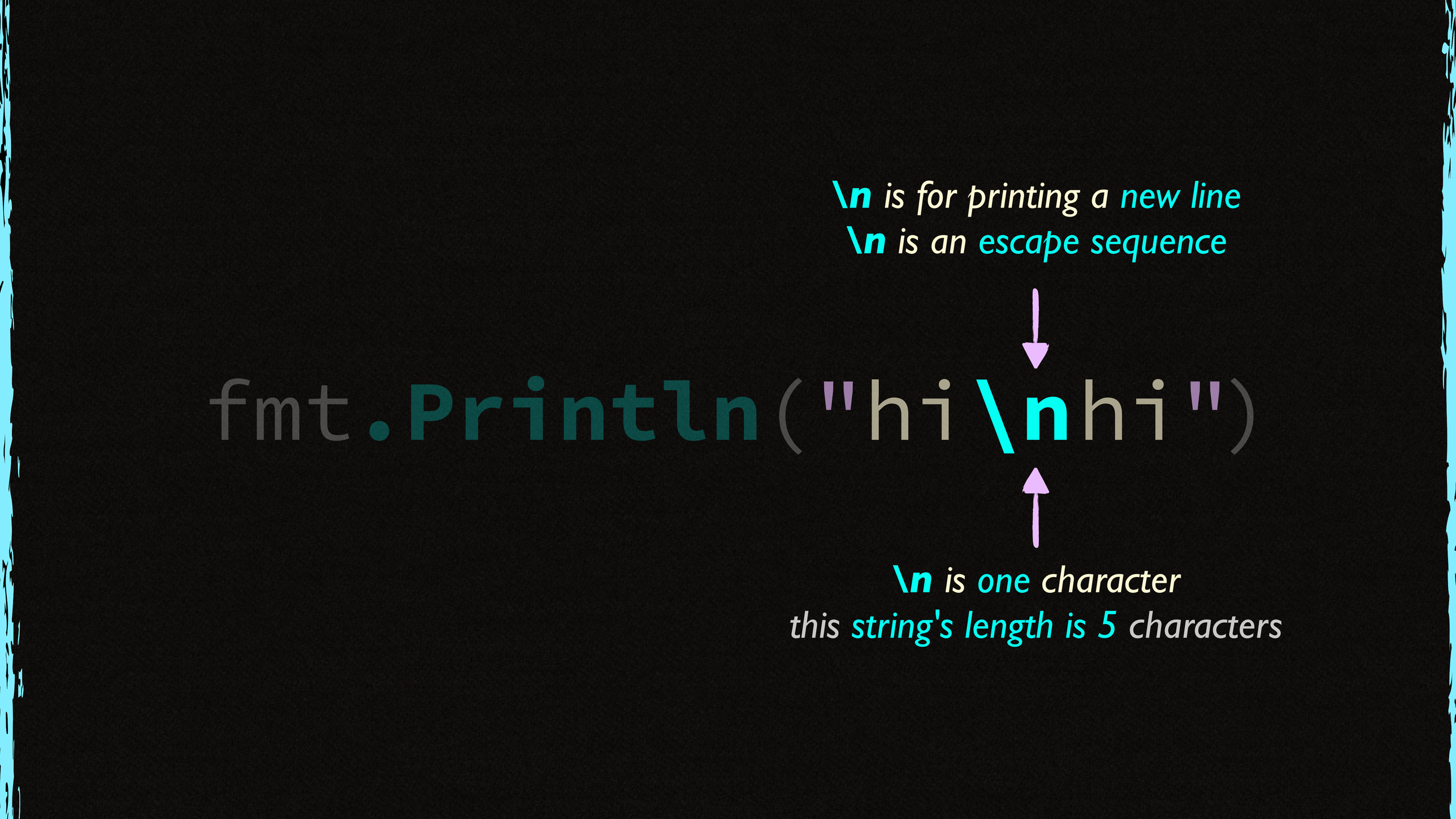
ESCAPE SEQUENCES

They give special meanings to characters

```go
func main() {
    fmt.Println("hi\nhi")
}
```

*Go **interprets** string values*

```
$ go run main.go
hi
hi
```

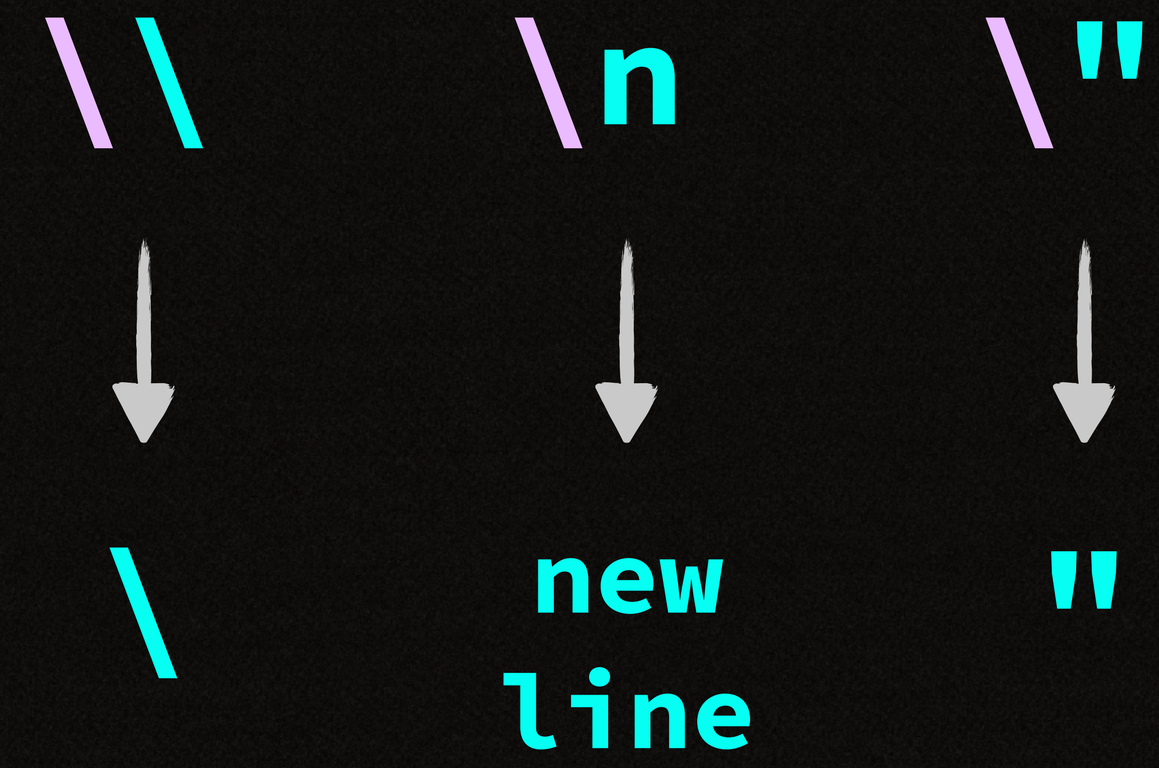

\n is for printing a new line
\n is an escape sequence
fmt.Println("hi\nhi")
\n is one character
this string's length is 5 characters

\n

When a character **starts** with a

\

it might be an **escape sequence**


\\
\n
\"
\
new
line
"


"hi\\n\"hi\""

ESCAPE SEQUENCES

They give special meanings to characters

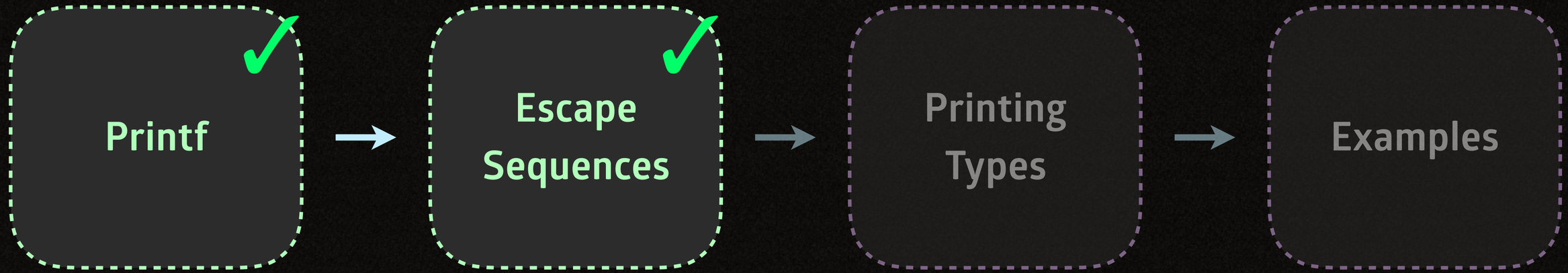
```go
func main() {
    fmt.Println("hi\\n\"hi\"")
}
```

*Go **interprets** string values*

```
$ go run main.go
hi\n"hi"
```

PRINTF

Congrats!



EXAMPLES

A few examples of printf

PRINTING TYPES

You can use printf to print the types of values (variables, constants, so on...)

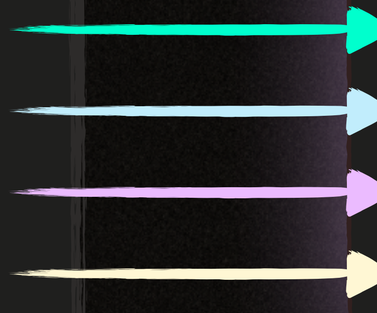
PRINTING TYPES

You can use printf to print the types of values (variables, constants, so on...)

```go
package main
import "fmt"

func main() {
   var speed int
   var heat  float64
   var off   bool
   var brand string

   fmt.Printf("%T\n", speed)
   fmt.Printf("%T\n", heat)
   fmt.Printf("%T\n", off)
   fmt.Printf("%T\n", brand)
}
```



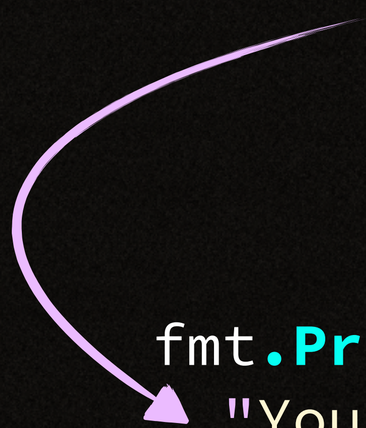
```
$ go run main.go
int
float64
bool
string
```

EXAMPLES

A few examples of printf

summary

format specifier



```
fmt.Printf(
  "Your age is %d, my age is %d.",
  age, yourAge,
)
```

verbs

%s %q %d %f %.2f %t %v %T

%v

(swiss-knife)

it doesn't check for the type-safety

but it can print any value

printing types with %T

escape sequences

`\n` `\"` `\\`

PRINTF

Done x 4!