## Verbes de formatage général

Les verbes suivants peuvent être utilisés avec tous les types de données :

|  |  |
| --- | --- |
| Verb | Description |
| %v | Prints the value in the default format |
| %#v | Prints the value in Go-syntax format |
| %T | Prints the type of the value |
| %% | Prints the % sign |

### Exemple

package main  
import ("fmt")  
func main() {  
  var i = 15.5  
  var txt = "Hello World!"  
  fmt.Printf("%v\n", i)  
  fmt.Printf("%#v\n", i)  
  fmt.Printf("%v%%\n", i)  
  fmt.Printf("%T\n", i)  
  fmt.Printf("%v\n", txt)  
  fmt.Printf("%#v\n", txt)  
  fmt.Printf("%T\n", txt)  
}

Résultat:

15.5  
15.5  
15.5%  
float64  
Hello World!  
"Hello World!"  
string

## Verbes de formatage entier

Les verbes suivants peuvent être utilisés avec le type de données entier :

|  |  |
| --- | --- |
| Verb | Description |
| %b | Base 2 |
| %d | Base 10 |
| %+d | Base 10 and always show sign |
| %o | Base 8 |
| %O | Base 8, with leading 0o |
| %x | Base 16, lowercase |
| %X | Base 16, uppercase |
| %#x | Base 16, with leading 0x |
| %4d | Pad with spaces (width 4, right justified) |
| %-4d | Pad with spaces (width 4, left justified) |
| %04d | Pad with zeroes (width 4 |

### Exemple

package main  
import ("fmt")  
func main() {  
  var i = 15  
  fmt.Printf("%b\n", i)  
  fmt.Printf("%d\n", i)  
  fmt.Printf("%+d\n", i)  
  fmt.Printf("%o\n", i)  
  fmt.Printf("%O\n", i)  
  fmt.Printf("%x\n", i)  
  fmt.Printf("%X\n", i)  
  fmt.Printf("%#x\n", i)  
  fmt.Printf("%4d\n", i)  
  fmt.Printf("%-4d\n", i)  
  fmt.Printf("%04d\n", i)  
}

Résultat:

1111  
15  
+15  
17  
0o17  
f  
F  
0xf  
  15  
15  
0015

## Verbes de formatage de chaîne

Les verbes suivants peuvent être utilisés avec le type de données chaîne :

|  |  |
| --- | --- |
| Verb | Description |
| %s | Prints the value as plain string |
| %q | Prints the value as a double-quoted string |
| %8s | Prints the value as plain string (width 8, right justified) |
| %-8s | Prints the value as plain string (width 8, left justified) |
| %x | Prints the value as hex dump of byte values |
| % x | Prints the value as hex dump with spaces |

### Exemple

package main  
import ("fmt")  
func main() {  
  var txt = "Hello"  
  fmt.Printf("%s\n", txt)  
  fmt.Printf("%q\n", txt)  
  fmt.Printf("%8s\n", txt)  
  fmt.Printf("%-8s\n", txt)  
  fmt.Printf("%x\n", txt)  
  fmt.Printf("% x\n", txt)  
}

Résultat:

Hello  
"Hello"  
   Hello  
Hello  
48656c6c6f  
48 65 6c 6c 6f

## Verbes de formatage booléens

Le verbe suivant peut être utilisé avec le type de données booléen :

|  |  |
| --- | --- |
| Verb | Description |
| %t | Value of the boolean operator in true or false format (same as using %v) |

### Exemple

package main  
import ("fmt")  
func main() {  
  var i = true  
  var j = false  
  fmt.Printf("%t\n", i)  
  fmt.Printf("%t\n", j)  
}

Résultat:

true  
false

## Verbes de formatage flottant

Les verbes suivants peuvent être utilisés avec le type de données float :

|  |  |
| --- | --- |
| Verb | Description |
| %e | Scientific notation with 'e' as exponent |
| %f | Decimal point, no exponent |
| %.2f | Default width, precision 2 |
| %6.2f | Width 6, precision 2 |
| %g | Exponent as needed, only necessary digits |

### Exemple

package main  
import ("fmt")  
func main() {  
  var i = 3.141  
  fmt.Printf("%e\n", i)  
  fmt.Printf("%f\n", i)  
  fmt.Printf("%.2f\n", i)  
  fmt.Printf("%6.2f\n", i)  
  fmt.Printf("%g\n", i)  
}

Résultat:

3.141000e+00  
3.141000  
3.14  
  3.14  
3.141