

### String and slice of bytes

|                 |  |
|-----------------|--|
| <code>%s</code> | the uninterpreted bytes of the string or slice       |
| <code>%q</code> | a double-quoted string safely escaped with Go syntax |
| <code>%x</code> | base 16, lower-case, two characters per byte         |
| <code>%X</code> | base 16, upper-case, two characters per byte         |

### General

|                   |  |
|-------------------|--|
| <code>%v</code>   | The value in a default format. When printing structs, the plus flag ( <code>%+v</code> ) adds field names. |
| <code> %#v</code> | a Go-syntax representation of the value  |
| <code>%T</code>   | a Go-syntax representation of the type of the value  |
| <code>%%</code>   | a literal percent sign; consumes no value  |

### The default format for %v

|  |   |
|--|---|
| <code>bool:</code>                     | <code>%t</code>                                       |
| <code>int, int8 etc.:</code>           | <code>%d</code>                                       |
| <code>uint, uint8 etc.:</code>         | <code>%d, %x</code> if printed with <code> %#v</code> |
| <code>float32, complex64, etc.:</code> | <code>%g</code>                                       |
| <code>string:</code>                   | <code>%s</code>                                       |
| <code>chan:</code>                     | <code>%p</code>                                       |
| <code>pointer:</code>                  | <code>%p</code>                                       |

### Other flags

|                             |  |
|-----------------------------|--|
| <code>+</code>              | always print a sign for numeric values; guarantee ASCII-only output for <code>%q</code> ( <code> %+q</code> ).   |
| <code>-</code>              | pad with spaces on the right rather than the left (left-justify the field).  |
| <code>#</code>              | alternate format: add leading 0 for octal ( <code> %#o</code> ), 0x for hex ( <code> %#x</code> ); 0X for hex ( <code> %#X</code> ); suppress 0x for <code>%p</code> ( <code> %#p</code> ); for <code>%q</code> , print a raw (backquoted) string if <code>strconv.CanBackquote</code> returns true; |
| <code>' '</code><br>(space) | leave a space for elided sign in numbers ( <code>% d</code> ); put spaces between bytes printing strings or slices in hex ( <code>% x, % X</code> ).   |
| <code>0</code>              | pad with leading zeros rather than spaces; for numbers, this moves the padding after the sign.   |

### Boolean

|                 |                        |
|-----------------|------------------------|
| <code>%t</code> | the word true or false |
|-----------------|------------------------|

### Integer

|                 |   |
|-----------------|---|
| <code>%b</code> | base 2  |
| <code>%c</code> | the character represented by the corresponding Unicode code point   |
| <code>%d</code> | base 10   |
| <code>%o</code> | base 8  |
| <code>%q</code> | a single-quoted character literal safely escaped with Go syntax     |
| <code>%x</code> | base 16, with lower-case letters for a-f                            |
| <code>%X</code> | base 16, with upper-case letters for A-F                            |
| <code>%U</code> | Unicode format: <code>U+1234</code> ; same as <code>"U+%04X"</code> |

### Floating-point and complex constituents

|                 |  |
|-----------------|--|
| <code>%b</code> | decimalless scientific notation with exponent a power of two, in the manner of <code>strconv.FormatFloat</code> with the 'b' format, e.g. <code>-123456p-78</code> |
| <code>%e</code> | scientific notation, e.g. <code>-1.234456e+78</code>   |
| <code>%E</code> | scientific notation, e.g. <code>-1.234456E+78</code>   |
| <code>%f</code> | decimal point but no exponent, e.g. <code>123.456</code>   |
| <code>%F</code> | synonym for <code>%f</code>  |
| <code>%g</code> | <code>%e</code> for large exponents, <code>%f</code> otherwise   |
| <code>%G</code> | <code>%E</code> for large exponents, <code>%F</code> otherwise   |

### Floating-point Precision

|                    |                                  |
|--------------------|----------------------------------|
| <code>%f</code>    | default width, default precision |
| <code>%9f</code>   | width 9, default precision       |
| <code>%.2f</code>  | default width, precision 2       |
| <code>%9.2f</code> | width 9, precision 2             |
| <code>%9.f</code>  | width 9, precision 0             |

### Pointer

|                 |                                   |
|-----------------|-----------------------------------|
| <code>%p</code> | base 16 notation, with leading 0x |
|-----------------|-----------------------------------|

