# Number formatting

Number formats can be classified into two categories:

* Standard formats - common formats which have short codes for easier use
* Custom formats - codes which allow for greater customizability, including custom suffixes, format widths, etc.

Even with formatting, the numbers need not be entered in that particular format - they will automatically be converted to the appropriate format.

## Standard formats

These are taken from formatting strings for .NET platform. They always follow the format of one letter (specifying the format), followed by 0-2 decimal digits specifying the width. If no width is specified, a default is used.

|  |  |  |
| --- | --- | --- |
| **Specifier Character** | **Name** | **Description** |
| “E” or “e” | Exponential | Result: Exponential notation  Width specifier: Number of decimal digits  Default width: 6 |
| “F” or “f” | Fixed point | Result: Integral and decimal digits with optional negative sign.  Width specifier: Number of decimal digits.  Default width specifier: Locale-specific |
| “N” or “n” | Numeric | Result: Integral and decimal digits, group separators, and a decimal separator with optional negative sign  Width specifier: Desired number of decimal places  Default width specifier: Locale-specific |

### The Exponential ("E") Format Specifier

The exponential ("E") format specifier formats the number as "-d.ddd…E+ddd" or "-d.ddd…e+ddd", where each "d" indicates a digit (0-9). The string starts with a minus sign if the number is negative. Exactly one digit always precedes the decimal point.

The width specifier indicates the desired number of digits after the decimal point. If the width specifier is omitted, a default of six digits after the decimal point is used. The number will either be padded with zeros, or rounded to that many decimal places if needed.

The case of the format specifier indicates whether to prefix the exponent with an "E" or an "e". The exponent always consists of a plus or minus sign and a minimum of three digits. The exponent is padded with zeros to meet this minimum, if required.

The following properties depend on the culture / locale of the machine:

* Decimal separator

Examples of formatting the value using the “en-US” locale:

* 15.0127 using “E” format → 1.501270E+001
* 15.0127 using “E4” format → 1.5013E+001
* 15.0127 using “e10” format → 1.5012700000e+001
* 15.0127 using “E” format (“fr-FR” locale) → 1,501270E+001

### The Fixed Point ("F") Format Specifier

The fixed-point ("F") format specifier converts a number to a string of the form "-ddd.ddd…" where each "d" indicates a digit (0-9). The string starts with a minus sign if the number is negative.

The width specifier indicates the desired number of decimal places. If the width specifier is omitted, the number of decimal places is locale specific, by default 2. The number will either be padded with zeros, or rounded to that many decimal places if needed.

If the width is 0, the number is effectively rounded to an integer.

The following properties depend on the culture / locale of the machine:

* Decimal separator
* Number of decimal digits (if not explicitly stated in the format)

Examples of formatting the value, using the “en-US” locale:

* 15.0127 using “F” format → 15.01
* 15.0127 using “F3” format → 15.013
* 15.0127 using “F6” format → 15.012700
* 15.0127 using “F0” format → 15
* 15.0127 using “F” format (“fr-FR” locale) → 15,01

### The Numeric ("N") Format Specifier

The numeric ("N") format specifier converts a number to a string of the form "-d,ddd,ddd.ddd…", where "-" indicates a negative number symbol if required, "d" indicates a digit (0-9), "," indicates a group separator, and "." indicates a decimal point symbol. The width specifier indicates the desired number of digits after the decimal point. If the width specifier is omitted, the number of decimal places is defined by the currentlocale.

The following properties depend on the culture / locale of the machine:

* Decimal separator
* Group separator
* Number of digits per group
* Number of decimal digits (if not explicitly stated in the format)

Examples of formatting the value, by default using the en-US locale:

* 15617.0127 using “N” format → 15,617.01
* 15617.0127 using “N3” format → 15,617.013
* 15617.0127 using “N6” format → 15,617.012700
* 15617.0127 using “N0” format → 15,617
* 15617.0127 using “N” format (“sv-SE” locale) → 15 617,01

## Custom formats

Custom formats can be used if additional customization is necessary. A custom format consists of the numeric part, which specifies how the number itself is formatted, and the optional prefix and suffix, which specify the text to precede or follow the number. The numeric part consists of:

* Digits **0** - denotes a digit of the number. If the number would not have a digit at that place, it will be padded with a zero.
* Characters **#** - denotes a digit of the number. If the number would not have a digit at that place, it would be skipped instead.
* Character **.** - denotes the decimal point in the number. If there is no decimal point in the format, number will be rounded. The decimal point will not be printed if no digits would be written after it. The actual character of the printed decimal point is dependant on the locale. The number of **0** and **#** characters on the right of the decimal point specify minimal and maximal number of decimal places. The number of **0** characters on the left of the decimal point specify the minimal number of integral digits. The number of **#** characters on the left is irrelevant, since number will never have its integral digits removed.
* Character **,** - denotes a number group separator. The presence of this character signifies that the number should be divided into groups for easier reading. The actual sizes of the group and the group separator character is dependant on the locale (usually it is every three digits)

Note: this character must **not** be the last character before the decimal point (e.g. **##,.0**) or at the end of the numeric part if there is no decimal point (e.g. **###0,**). Other than that, the position of the character has no influence on the formatting, only it’s presence in the format.

* Characters **E** or **e**, followed by optional **+** or **-**, followed by at least one **0** digit signifies that the number should be presented in exponential notation. The case of the first letter signifies which case should the notation be written in. If a **+** sign follows the **E** or **e**, a sign of the exponent is always printed, otherwise if there is a **-** or no sign, the sign of the exponent will be printed only if it’s negative. The number of **0** characters signifies the minimum number of digits in the exponent which will be padded with zeros if needed.

|  |  |  |
| --- | --- | --- |
| **Format** | **Explanation** | **Examples** |
| 0 | Just the integral part of the number, at least one digit. | 15.0127 → 15 |
| 00000 | Just the integral part of the number, with at least 5 digits, padded with zeros if needed. | 15.0127 → 00015  123456 → 123456 |
| 0.### | Number with 0-3 decimal digits. Note: if there are no decimal places, the decimal point is not printed | 15.0127 → 15.013  15.1 → 15.1  15 → 15 |
| 0.0## | Number with 1-3 decimal digits. Since there is always at least one digit after the decimal point, it is always printed. | 15.0127 → 15.013  15.1 → 15.1  15 → 15.0 |
| #,##0.00 | Number with 2 decimal places, and with group separators inserted. | 15 → 15.00  15000 → 15,000.00  12345.678 → 12,345.68 |
| 0.###E-0 | Exponential notation with one integral digit, and up to 3 decimal digits, with no exponent padding. | 15 → 1.5E1  15.1 → 1.51E1  15.0127 → 1.501E1 |
| 0.###E+0 | Exponential notation with one integral digit, and up to 3 decimal digits, with no exponent padding. The + sign will be printed if the exponent is positive. | 15 → 1.5E+1  15.1 → 1.51E+1  15.0127 → 1.501E+1 |
| 0.###E+000 | Exponential notation with one integral digit, and up to 3 decimal digits, with exponent padded to at least 3 digits. The + sign will be printed if the exponent is positive | 15 → 1.5E+001  15.1 → 1.51E+001  15.0127 → 1.501E+001  12345 → 1.235E+004 |

The prefix and suffix are text inserted before and after the numeric part. All the characters should be quoted using either single-quote ' or double-quote **"** characters. The \ character is used to escape other characters (including quotes and backslashes) if they need to be placed inside quoted prefixes or suffixes. While the quotes are not necessary, they are needed if special characters are used in the suffix, and it is recommended to place them anyway.

Examples:

* **0.#####" K"** will append “ K” to the formatted number (note the space inside the quotes, and no space between the number and the quotes. 12.345 → 12.345 K
* **"$"0.00** will prepend the dollar sign to the formatted number. 1.234 → $1.23
* **"±"0.##" mm"** the “±” sign to the number, and append “ mm”. 17.5 → ±17.5 mm

Due to technical limitations, the prefixes and suffixes must not be misinterpretable as a number - the prefix must not contain digits or +/- signs; and the suffix should not start with digits or the characters **e** or **E** if they can be misinterpreted as a continuation of the number.

Prefixes and suffixes can be used to append measurement units to the data: For example the format of **0.0##" GHz"** will append a GHz unit to the input value: 1.85 → 1.85 GHz.

Note: there is no automatic unit conversion performed: thus entering a value like “1500 MHz” will format it as “1500 GHz” ignoring the supplied units. The value entered must be in correct units.

# Date formatting

Dates are formatted by arranging the individual datetime elements, and the separators in between. For example, the format **dd-MM-yyyy** formats the 16th of August 2018 as **16-08-2018**.

The datetime elements are the following:

* **d** - Day of the month, as either one or two digits.
* **dd** - Day of the month, as two digits padded with zeros
* **ddd** - Abbreviated localized day of the week
* **dddd** - Full localized day of the week
* **f...f** - The fractional part of the second, rounded to as many characters as **f** characters (1-7)
* **h** - The hour, using a 12-hour clock
* **hh** - The hour, using a 12-hour clock, padded with a zero to two digits
* **H** - The hour, using a 24-hour clock
* **HH** - The hour, using a 24-hour clock, padded with a zero to two digits
* **m** - The minute, from 1 to 59
* **mm** - The minute, from 01 to 59, padded with a zero to two digits
* **M** - The month, from 1 to 12
* **MM** - The month, from 01 to 12, padded with a zero to two digits
* **MMM** - Abbreviated localized name of the month
* **MM** - Full localized name of the month
* **s** - The second, from 1 to 59
* **ss** - The second, from 01 to 59, padded with a zero to two digits
* **t** - One letter AM/PM specifier, localized
* **tt** - Full AM/PM specifier, localized
* **y** - The two digit year, not padded with zeros (2008 → 8 and 2018 → 18)
* **yy** - The two digit year, padded with a zero if needed (2008 → 08 and 2018 → 18)
* **:** - The localized time part separator
* **/** - The localized date part separator
* **'** and **"** - Used for quoting special characters
* **\** - Used for escaping the special meaning of the next character

Other alphabetic characters should not be used. All other characters will be copied to the output verbatim and used as separators. Note: to use **:** and **/** without localization, quote them using **'** or **"**.

Examples:

* **yyyy-MM-dd HH':'mm':'ss** → 2018-08-16 20:47:13
* **dd/MM/yyyy** → 16/08/2018
* **HH':'mm** → 20:47
* **hh':'mm tt** → 08:47 PM

# Endnotes

Some parts are based on modified .NET Documentation by Microsoft available at <https://docs.microsoft.com/en-us/dotnet>, released under CC BY 4.0 licence.