The \texttt{l3str-format} package: formatting strings of characters

The \LaTeX{} Project\textsuperscript{*}

Released 2021-05-27

1 Format specifications

In this module, we introduce the notion of a string \texttt{(format)}. The syntax follows that of Python’s \texttt{format} built-in function. A \texttt{(format specification)} is a string of the form

\[
\texttt{(format specification)} = \[[\texttt{[fill]}](\texttt{alignment})][\texttt{[sign]}][\texttt{[width]}][.(\texttt{precision})][\texttt{[style]}]
\]

where each \texttt{[\ldots]} denotes an independent optional part.

- \texttt{(fill)} can be any character: it is assumed to be present whenever the second character of the \texttt{(format specification)} is a valid \texttt{(alignment)} character.
- \texttt{(alignment)} can be \texttt{<} (left alignment), \texttt{>} (right alignment), \texttt{^} (centering), or \texttt{=} (for numeric types only).
- \texttt{(sign)} is allowed for numeric types; it can be \texttt{+} (show a sign for positive and negative numbers), \texttt{-} (only put a sign for negative numbers), or a space (show a space or a \texttt{-}).
- \texttt{(width)} is the minimum number of characters of the result: if the result is naturally shorter than this \texttt{(width)}, then it is padded with copies of the character \texttt{(fill)}, with a position depending on the choice of \texttt{(alignment)}. If the result is naturally longer, it is not truncated.
- \texttt{(precision)}, whose presence is indicated by a period, can have different meanings depending on the type.
- \texttt{(style)} is one character, which controls how the given data should be formatted. The list of allowed \texttt{(styles)} depends on the type.

The choice of \texttt{(alignment)} = is only valid for numeric types: in this case the padding is inserted between the sign and the rest of the number.

\textsuperscript{*}E-mail: latex-team@latex-project.org
2 Formatting various data-types

\texttt{\textbackslash tl\_format:nn} *(token list)* *(format specification)*

Converts the *(token list)* to a string according to the *(format specification)*. The *(style)*, if present, must be *s*. If *(precision)* is given, all characters of the string representation of the *(token list)* beyond the first *(precision)* characters are discarded.

\texttt{\textbackslash seq\_format:nn} *(sequence)* *(format specification)*

Converts each item in the *(sequence)* to a string according to the *(format specification)*, and concatenates the results.

\texttt{\textbackslash int\_format:nn} *(intexpr)* *(format specification)*

Evaluates the *(integer expression)* and converts the result to a string according to the *(format specification)*. The *(precision)* argument is not allowed. The *(style)* can be *b* for binary output, *d* for decimal output (this is the default), *o* for octal output, *X* for hexadecimal output (using capital letters).

\texttt{\textbackslash fp\_format:nn} *(fpexpr)* *(format specification)*

Evaluates the *(floating point expression)* and converts the result to a string according to the *(format specification)*. The *(style)* can be

- *e* for scientific notation, with one digit before and *(precision)* digits after the decimal separator, and an integer exponent, following *e*;
- *f* for a fixed point notation, with *(precision)* digits after the decimal separator and no exponent;
- *g* for a general format, which uses style *f* for numbers in the range \([10^{-4}, 10^{\langle \text{precision} \rangle})\) and style *e* otherwise.

When there is no *(style)* specifier nor *(precision)* the number is displayed without rounding. Otherwise the *(precision)* defaults to 6.

3 Possibilities, and things to do

- Provide a token list formatting *(style)* which keeps the last *(precision)* characters rather than the first *(precision)*.

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

\text{F}

fp commands:

\texttt{\textbackslash fp\_format:nn} ....................... 2
<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>int commands:</td>
<td>\texttt{\textbackslash int_format:nn} \hfill 2</td>
<td>\texttt{\textbackslash tl_format:Nn} \hfill 2</td>
</tr>
<tr>
<td></td>
<td>\texttt{\textbackslash tl_format:nn} \hfill 2</td>
<td>\texttt{\textbackslash tl_format:nn} \hfill 2</td>
</tr>
<tr>
<td>seq commands:</td>
<td>\texttt{\textbackslash seq_format:Nn} \hfill 2</td>
<td></td>
</tr>
</tbody>
</table>