

Contents

1	Format specifications	2
2	Formatting various data-types	3
3	Possibilities, and things to do	3
	Index	3

The l3str-format package: formatting strings of characters

The L^AT_EX3 Project*

Released 2017/05/13

1 Format specifications

In this module, we introduce the notion of a string $\langle format \rangle$. The syntax follows that of Python's `format` built-in function. A $\langle format specification \rangle$ is a string of the form

$$\langle format specification \rangle = [[\langle fill \rangle]\langle alignment \rangle][\langle sign \rangle][\langle width \rangle][.\langle precision \rangle][\langle style \rangle]$$

where each [...] denotes an independent optional part.

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

The choice of $\langle alignment \rangle =$ is only valid for numeric types: in this case the padding is inserted between the sign and the rest of the number.

*E-mail: latex-team@latex-project.org

2 Formatting various data-types

<hr/>	
<code>\tl_format:Nn</code> ★	<code>\tl_format:nn {<token list>} {<format specification>}</code>
<code>\tl_format:cn</code> ★	
<code>\tl_format:nn</code> ★	Converts the <i><token list></i> to a string according to the <i><format specification></i> . The <i><style></i> , if present, must be <i>s</i> . If <i><precision></i> is given, all characters of the string representation of the <i><token list></i> beyond the first <i><precision></i> characters are discarded.
<hr/>	
<code>\seq_format:Nn</code> ★	<code>\seq_format:nn {<sequence>} {<format specification>}</code>
<code>\seq_format:cn</code> ★	Converts each item in the <i><sequence></i> to a string according to the <i><format specification></i> , and concatenates the results.
<hr/>	
<code>\int_format:nn</code> ★	<code>\int_format:nn {<intexpr>} {<format specification>}</code>
	Evaluates the <i><integer expression></i> and converts the result to a string according to the <i><format specification></i> . The <i><precision></i> argument is not allowed. The <i><style></i> can be <i>b</i> for binary output, <i>d</i> for decimal output (this is the default), <i>o</i> for octal output, <i>X</i> for hexadecimal output (using capital letters).
<hr/>	
<code>\fp_format:nn</code> ★	<code>\fp_format:nn {<fpexpr>} {<format specification>}</code>
	Evaluates the <i><floating point expression></i> and converts the result to a string according to the <i><format specification></i> . The <i><precision></i> defaults to 6. The <i><style></i> can be <ul style="list-style-type: none"> • <i>e</i> for scientific notation, with one digit before and <i><precision></i> digits after the decimal separator, and an integer exponent, following <i>e</i>; • <i>f</i> for a fixed point notation, with <i><precision></i> digits after the decimal separator and no exponent; • <i>g</i> for a general format, which uses style <i>f</i> for numbers in the range $[10^{-4}, 10^{<precision>})$ and style <i>e</i> otherwise.

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.

F

fp commands:

`\fp_format:nn` *3, 3*

I

int commands:

`\int_format:nn` *3, 3*

S		T	
seq commands:		tl commands:	
		<code>\tl_format:Nn</code> 3
<code>\seq_format:Nn</code> 3, 3	<code>\tl_format:nn</code> 3, 3