

tex.stackexchange.com

What's the difference between `\newcommand` and `\newcommand*`?

Caramdir
Caramdir 76.2k2121 gold badges224224 silver badges275275 bronze badges

3 minutes

A bit of background first. When Knuth wrote TeX, he realised that most macros would not need to absorb more than one paragraph as an argument. As a result, a good way to test for errors such as a missing `}` is to forbid macros to accept paragraph tokens (either those generated by blank lines or explicit `\par` tokens). So he created `\def` for definitions which cannot take a paragraph token, and the `\long` prefix to allow them to:

```
\def\examplea#1{% #1 cannot contain \par
}
\long\def\exampleb#1{% #1 can contain \par
}
```

When LaTeX was written, Lamport created `\newcommand` as a wrapper around `\def` with various bits of error checking. He found that paragraph tokens can pop up in odd places. So he

decided that `\newcommand` would be 'long'. When the LaTeX team took over for LaTeX2e, they decided that you would need to add the modifier `*` to prevent this:

```
\newcommand{\examplea}[1]{% #1 can contain  
\par  
}  
\newcommand*{\exampleb}[1]{% #1 cannot contain  
\par  
}
```

Most of the time, `\newcommand*` is the best choice as you want the error-checking that it provides. That is why examples given by experienced LaTeX users normally use this form, rather than just `\newcommand`.

The same behaviour is seen with `\newenvironment`:

```
\newenvironment{examplea}[1]{% #1 can contain  
\par  
}{}  
\newenvironment*{exampleb}[1]{% #1 cannot  
contain \par  
}{}  
}
```

This works by defining `\examplea` and `\endexamplea` more-or-less using `\newcommand`, and `\exampleb` and `\endexampleb` more-or-less using `\newcommand*`. As a result, the 'end' macros have the same 'long' status as the 'begin' ones, even though they never take arguments. (Note

What's the difference between `\newcommand` and `\newcommand*`?

that this does not affect what can go into the *body* of the environment, only the *arguments* at the start.) Environments are covered in more detail in [What is the difference between `\newenvironment` and `\newenvironment*`?](#)

For LaTeX3, we've decided to take a somewhat hybrid approach. If you use `xparse` for defining document commands, they are only 'long' if you ask for it:

```
\NewDocumentCommand\examplea{m}{% #1 cannot  
contain \par  
}
```

```
\NewDocumentCommand\examplab{+m}{% #1 can  
contain \par  
}
```

(We've decided on + to represent a long argument). On the other hand, for internal use it is normally best to accept paragraph tokens, and to leave the filtering to the user interface level.