Tutorial 8: A brief introduction to LaTeX 2ε macro commands

Pierre Chardaire ©UEA

July 2, 2017

This tutorial gives an introduction to the macro facilities provided by LaTeXe. Macros can help you become more productive[1]. In the next two sections you will learn how to create your own commands and environments.

First, create a LaTeX 2ε document in the usual way.

1 Creating commands

LaTeX 2ε enables you to create your own commands to speed up your work. For example, add to the preamble (before the `\begin{document}`) the following

```
\newcommand{\uea}
{the University of East Anglia}
```

Now type the following in your document:

```
I am a student at \uea. I think \uea{} is a
good place to study \ldots{} \LaTeXe!
```

[1] Sorry, I should have said "*even* more productive"!

After compiling this should produce

> I am a student at the University of East Anglia. I think the University of East Anglia is a good place to study … LaTeX 2ε!

Macros are useful to save typing. Experienced typists would look at the text they have to type and would define macros to simplify repetitive typing. Macros are also useful when some text may be changed. For example you may want to change the macro `\uea` to

```
\newcommand{\uea}
{the School of Computing Sciences at UEA}
```

Then if you reprocess your document the text you previously typed should become

> I am a student at the School of Computing Sciences at UEA. I think the School of Computing Sciences at UEA is a good place to study … LaTeX 2ε!

A simple command like `\uea` is defined by typing `\newcommand{name}{def}` where *name* is the name of the command and *def* is what it does. If definition is some text then LaTeX 2ε replaces each occurrence of the command with the definition. Names of commands must start with a "\" as in `\uea`. They cannot start with `\end` (e.g. `\endthingmebob`) and they cannot be formed with the name of an existing environment, e.g. you cannot use `\center`.

Suppose now that you want to write an article that involve many vectors of the form $(x_p, \ldots, x_n)$. You can create the following command[2]

```
\newcommand{\vect}[3]
{#3_{#1},\ldots,#3_{#2}}
```

In this definition `[3]` indicates that 3 parameters are passed and these parameters are identified by `#1`,`#2` and `#3`.

[2] Don't forget commands must be created in the preamble

You use the command as follows

```
\[\vect{1}{10}{x}\qquad\vect{0}{n}{y}\]
```

to produce

$$x_1,\ldots,x_{10}\qquad y_0,\ldots,y_n$$

In `\vect{1}{10}{x}` the parameter `#1`, `#2` and `#3` are given the values 1, 10 and `x` respectively.

You can also define a default value for the first parameter `#1`. Modify your command `vector` as follow

```
\newcommand{\vect}[3][1]
{#3_{#1},\ldots,#3_{#2}}
```

The second `[1]` indicates that the default value for `#1` is 1. Now you can type

```
\[\vect{10}{x}\qquad\vect[0]{n}{y}\]
```

to get

$$x_1,\ldots,x_{10}\qquad y_0,\ldots,y_n$$

Notice that the optional parameter when provided must be between square brackets.

2 Creating environments

You can also create your own environments. Environments need to be defined in the preamble.

Here is an example:

```
\newenvironment{example}
{% This is the beginning code
\itshape\par\noindent\ignorespaces}
{% This is the end code
\par\noindent\ignorespacesafterend}
```

In your text you can type

```
Now we shall give an example:
\begin{example}
This is a little example
\end{example}
What an interesting example!
```

After processing you should get:

> Now we shall give an example:
> *This is a little example*
> What an interesting example!

In general you define a simple environment by typing

```
\newenvironment{name}
{opening section}
{closing section}
```

where *name* is the name of the environment. Note that the name of an environment is a simple name without a starting backslash.

Have a look at the `lshort.pdf` guide on blackboard to have an explanation of the use of `\ignorespaces` and `\ignorespacesafterend`

As with commands you can pass parameters to an environment.

Let us modify the `example` environment as follows:

```
\newenvironment{example}[1]
{% This is the beginning code
\itshape\par\noindent\ignorespaces
{\bf\small Example~#1}\par}
{% This is the end code
\par\noindent\ignorespacesafterend}
```

Now in your text you can type

```
A numbered example:
\begin{example}{42}
This is a little example
\end{example}
So many examples?
```

After processing you should get:

A numbered example:
Example 42
This is a little example
So many examples?

You can use up to nine parameters in an environment.

In the above case we want to be able to number examples. To do so it is much better to use a LaTeX 2ε counter. Modify the example environment and introduce counters as follows

```
\newcounter{Examplecount}
\setcounter{Examplecount}{0}
\newenvironment{example}
{% This is the beginning code
\itshape\par\noindent\ignorespaces
\stepcounter{Examplecount}
{\bf\small Example~\arabic{Examplecount}}
\par}
{% This is the end code
\par\noindent\ignorespacesafterend}
```

And in the text type

```
\begin{example}
My first example.
\end{example}
\begin{example}
My second example.
\end{example}
```

You should get

Example 1
My first example.
Example 2
My second example.

Sometimes you want to use a passed parameter in the closing section. Suppose you want to create an environment that will be used as follows

```
\begin{aphorism}{Lao Tse}
A tree that can fill the span
of a man's arms grows from a downy tip;\\
A terrace nine stories high rises from level earth;\\
A journey of a thousand miles starts
from beneath one's feet.
\end{aphorism}
```

to produce

> *A tree that can fill the span of a man's arms grows from a downy tip;*
> *A terrace nine stories high rises from level earth;*
> *A journey of a thousand miles starts from beneath one's feet.*
>
> Lao Tse

A first attempt may be to define `aphorism` as follows:

```
\newenvironment{aphorism}[1]
{\begin{quote}\itshape}
{\par\hfill\normalfont#1\end{quote}}
```

If you try this, LaTeX 2ε produces an error (in the `messages` window)

```
Illegal parameter number in definition of
\endaphorism
```

This is normal. LaTeX 2ε does not allow you to use the passed parameters in the closing section of the environment. The way to solve the problem is by using `\newcommand` to define a command `\name` to memorise the parameter.

```
\newenvironment{aphorism}[1]
{\begin{quote}\newcommand{\name}{#1}\itshape}
{\par\hfill\normalfont\name\end{quote}}
```

Processing the *aphorism* typed above you should get what you wanted to produce.

3 For the LaTeX 2ε enthusiast

There is the great package `xparse` that simplifies the definition of complex commands and environments. It is part of the LaTeX3 development, and is included in the LaTeX 2ε distribution.