

Defining your own commands

I will only touch on this subject. In the preamble, you may define your own commands with the `\newcommand{[]}{}` command. These become indispensable when writing long, structured documents. For example, all through these example documents, I have been writing package names in a sans serif font and commands in a fixed-width typewriter font, but using `\textsf` and `\texttt` respectively.

It would have been much better of me to define *new* commands, let's say `\pkg` and `\cmd`, for these tasks. Have a look in the preamble to this document how I should have gone about this:

```
\newcommand{\pkg}[1]{\textsf{#1}}
\newcommand{\cmd}[1]{\texttt{#1}}
```

The commands should be fairly self-evident. The first argument is the name of the new command. The second optional argument is the number of arguments your new command is to have; and the third argument stands for the definition of your command, with passed arguments `#1...#9` (yes, that means no more than nine arguments).

So now let's say I'm talking about the `\cmd{href}` command of the `\pkg{hyperref}` package. Notice I'm using my new commands. Not only are they styled how I like, they even make *logical* sense now. The source now contains more *meaning* than if I'd written “the `\texttt{href}` command of the `\textsf{hyperref}` package”. The same is even more true for my other new command `\strong`—see the first word of the next paragraph.

Furthermore, what if I wish to *change* the way I'm displaying those commands? Imagine if I'd typed in explicitly every font-changing command (as is the case for all of these example documents). To change packages to bold and commands to small caps I'd have to go through the whole lot of them and alter *every single instance*. If I've defined `\pkg` in the preamble and used it instead, I simply have to alter my definition and re-compile my document. I'm sure you can see the advantage.