Chapter 11

Commands and Environments

This chapter studies user-defined commands and environments in \LaTeXe, which is the \LaTeX implementation that was current at the time this chapter was written.

11.1 Some Terminology

This section briefly introduces some terminology for the remainder of this chapter. Throughout this chapter we shall use the word parameter for an argument or parameter of a macro or command. An actual parameter is a parameter that is passed to an existing command. A formal parameter is a placeholder in the definition of a command for an actual parameter.

For example, consider the mathematical function definition that is given by

\[ f : \mathbb{N} \to \mathbb{N} \]
\[ x \mapsto 2x. \]

The definition of \( f(\cdot) \) has two parts. Equation 11.1 is the first part of the definition; it determines the signature of the function. In \LaTeX there is no equivalent for the signature. Equation 11.2 is the second part of the definition; it defines the semantics of the function. The semantics may be regarded as an input-output transformation, with \( x \) determining the input parameter and \( 2x \) determining the output. The input parameter \( x \) defines a name that may be used in the output expression. This name acts as the first (and only) formal parameter in the definition of \( f(\cdot) \). Using the function, one may write \( f(1) \), or even \( f(x) \), assuming that \( x \) has a proper context. Here \( 1 \) and \( x \) are actual parameters. The result of the expression \( f(a) \), where \( a \) is an actual parameter, can be found by substituting the actual parameter \( a \) for the formal parameter \( x \) in the method’s definition: \( f(a) = (2x) \big|_{x=a} \), which gives us \( 2a \). Effectively, formal and actual parameters in \LaTeX work in a similar way.

11.2 Advantages and Disadvantages

\LaTeX is a programmable typesetting engine. Commands are the key
to controlling your document. The advantages of using commands in \LaTeX are similar to the advantages of using functions and procedures in high-level programming languages. However, \LaTeX commands also have disadvantages. We shall first study advantages and then disadvantages. The following are some advantages.

**software engineering** Tedioustasks can be automated. This has the following advantages.

- **reusability** Commands that are defined once can be reused several times.
- **simplicity** Carrying out a complex task using a simple command with a well-understood interface is much easier and leads to fewer errors.
- **refinement** You can stepwise refine the implementation of certain tasks. This lets you postpone certain decisions. For example, if you haven’t been able to decide how to typeset certain symbols that serve a certain purpose, then you may start typesetting them using a command that typesets them in a simple manner. This lets you start writing the document in terms of high-level notions (procedural markup). By refining the command at a later stage, you can fine-tune the typesetting of all the relevant symbols.
- **maintainability** This advantage is related to the previous item. Unforeseen changes in requirements can be implemented easily by making a few local changes.
- **consistency** Typesetting entities using carefully chosen commands guarantees a consistent appearance of your document. For example, if you typeset your pseudo-code identifiers using a pseudo-code identifier typesetting command in a “pseudo-code identifier” style, then your identifiers will have a consistent feel.

**computing** Tasks and results may be computed depending on document options. This has the following advantages.

- **style control** Things may be typeset in a style that depends on class or package options. For example, the `article` class typesets the main text in 10 pt by default but providing a `12pt` option gives you a 12 pt size.
- **content control** Commands may result in different output depending on a global mode. For example, consider the `beamer` class, which lets you prepare a computer presentation and lecture notes in the same input. In `presentation` mode the `beamer` class results in a computer presentation but in `article` mode it may result in lecture notes. You can share text for the notes and the presentations but you can also hide text in the notes or in the presentation. This is very a strong feature because it allows sharing and guarantees consistency between the notes and the presentation.
- **typeset results** This issue is related to the previous item. \LaTeX can do basic arithmetic, can branch and iterate, and can typeset the results of computations. For example, the `lipsum` package