

文章编号: 2095-2163(2019)03-0199-03

中图分类号: TP317.2

文献标志码: A

基于程序员思维研究 LaTeX 排版

王爽¹, 赵会洋^{1,2}

(1 许昌学院 信息工程学院, 河南 许昌 461000; 2 北京邮电大学 自动化学院, 北京 100876)

摘要: LaTeX 是国际公认的一个非常出色的排版工具, 对各类科技工作者的研究发表起着非常重要的辅助作用。不同于 Word 排版的所见即所得, LaTeX 是基于不同的命令实现排版的, 因此对于初学者来说有一定的难度。本文基于程序员的思维, 研究编程中的变量、函数等重要内容与 LaTeX 排版的关系。这对快速学习与掌握 LaTeX 具有重要的作用, 对灵活运用 LaTeX 进行排版具有实际意义。

关键词: LaTeX; 编程; 排版; 变量; 函数

Research on LaTeX typesetting from programmers' thinking

WANG Shuang¹, ZHAO Huiyang^{1,2}

(1 School of Information Engineering, Xuchang University, Xuchang Henan 461000, China;

2 School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China)

[Abstract] LaTeX is internationally recognized as a great typography tool. For all kinds of scientific and technological workers, it has great supporting effects on the research and paper publication. Word is a tool by which what you see is what you get. However, LaTeX is different from the Word. A typesetting task is finished based on some different command in LaTeX. Thus LaTeX is certainly difficult for beginners. In a way of programmers' thinking, the paper explores the import contents in programming which is related to LaTeX, such as variables, functions, and so on. The research plays an important role in fast learning and mastering LaTeX, and has practical significance on the flexible typesetting using LaTeX.

[Key words] LaTeX; programming; typesetting; variable; function

0 引言

TEX 是高德纳 (Donald E. Knuth) 开发的、以排版文字和数学公式为目的的一个计算机软件^[1]。LaTeX 是对 TEX 的封装和拓展, 拥有比原来 TEX 格式 (Plain TEX) 更为规范的命令和一整套预定义的格式, 隐藏了不少排版方面的细节, 可以让使用者相对容易地以较高质量排版和印刷拟发表的作品。LaTeX 的最初开发者为 L^ampo^rt 博士^[2]。目前, LaTeX 已成为国际学术界热门流行的排版系统之一, 许多国际著名的出版机构和期刊编辑部都要求或推荐其作者使用 LaTeX 投稿^[3]。关于 LaTeX 的基础知识和在科技论文写作中的应用, 在国内外的文献中也可见到相关报道及探讨^[4-7]。对于初学者而言, LaTeX 让人望而却步^[3]。考虑到时下有数量颇为可观的科技工作者具备编程的基础, 因此, 本文将以前程序员的思维研究 LaTeX 学习, 通过对比学习的方法及通俗易懂的案例使读者能够达到 LaTeX 的快速入门。

1 变量的定义与使用

变量是用来存储各类数据的符号。在变量定义之后, 就可以在其它位置使用该变量名代替其所要表达的值。如果要改变变量的值, 也只需要在定义的位置修改, 所有使用该变量的地方都会用新的值代替。这就是设置变量的好处。下面将 Java 中的变量定义和 LaTeX 中的变量定义格式进行对比。

在 Java 中自定义变量的格式如下:

变量类型 变量名 = 变量值

在 LaTeX 中自定义变量的格式如下:

`\newcommand{变量名}{变量值}`

通过上面的定义对比可以看出, LaTeX 中定义变量不需要类型, 但需要一个关键字 `newcommand`。LaTeX 中的自定义变量名和系统内置的命令一样, 必须以反斜杠开头, 一般由若干个字母组成。LaTeX 中的变量值可以是字符串、数值、布尔值等内容, 但由于没有不同的类型符, 所以格式是统一的。另外,

基金项目: 河南省高等学校重点科研项目 (15B520031)。

作者简介: 王爽 (1982-), 女, 硕士, 讲师, 主要研究方向: 数据挖掘、复杂网络; 赵会洋 (1981-), 男, 博士, 讲师, 主要研究方向: 复杂网络、机器学习。

收稿日期: 2019-02-03

LaTeX 中的变量名和变量值之间没有赋值符号,而是通过括号来区分二者。文中将通过实验来说明变量如何定义与使用,主要包含 2 个步骤。对此可做解析分述如下。

(1)变量定义。LaTeX 中定义变量是在导言区进行的,下一节要讨论的函数也在此区域定义。本实验中定义 3 个类型的变量,分别是字符串型、数值型和布尔型,如图 1 所示。

```
\newcommand{\varstr}{width}
\newcommand{\varnum}{5}
\newcommand{\varbool}{true}
```

图 1 变量定义

Fig. 1 Definition of variables

需要注意的是,LaTeX 是将用户想呈现的内容经过排版之后进行输出,是一个科技文献排版工具,而不是一种软件开发语言。因此,从本例中可以看出 3 个变量的值在定义时没有太大的区别,都是直接书写。特别是第一个字符串并没有双引号或单引号,第三个布尔型只是根据其值判断的,从形式上看与字符串并没有什么区别。

(2)变量使用。LaTeX 中使用变量的方法是直接使用变量名,使用变量的位置可以在导言区、也可以在文档区。如果是需要直接输出的变量,一般放在文档区。如果是中间变量,根据使用场合可能出现在文档区或导言区。在图 2 中即显示了上述 3 个变量的使用。

```
\varstr\par
\varnum\par
\varbool\par
```

图 2 变量使用

Fig. 2 Usage of variables

width
5
true

图 3 变量实验结果

Fig. 3 Experiment result of variables

由图 2 看出,变量在使用时的格式和定义时是一样的,也需要以反斜杠开头,然后是变量名。变量名后的\par 是另起一段的标志,是为了输出效果而添加的。

(3)结果输出。LaTeX 中所有要输出的内容都要放在文档区,经过编译之后会输出在 pdf 文档中。本节中的实验输出结果如图 3 所示。将图 3 和图 1 对比后可以看出,输出的变量值和定义时是一样的。

2 函数

函数是用来完成特定功能的若干代码的封装。因此,相较于变量值是直接给出具体值,函数体会有长有短,具体由函数的功能复杂程度决定。另外,函数需要满足传参数的需要。这里即将 Java 中和

LaTeX 中函数的定义与使用进行对比,研究内容详见如下。

2.1 函数的定义

Java 中定义函数的格式:

[权限标识符] 返回值 函数名(参数列表) {函数体}

LaTeX 中定义函数的格式:

\newcommand{函数名}[参数个数]{函数体}

通过对比发现,首先,LaTeX 中定义函数也需要 newcommand 关键字。其次,LaTeX 中没有权限标识符,在 tex 文件的导言区定义的函数,其作用范围是整个 tex 文件。LaTeX 中也没有返回值标识,是否有返回值取决于函数体是否有输出。再者,形式上两者之间也有相似的地方,即函数名、参数和函数体。但是在运行实施中,这些相似的地方还是有些区别的。LaTeX 的函数名与变量名一样,也需要以反斜杠开头。Java 中的参数部分用的是以圆括号包括起来的参数列表,包括若干参数类型和参数名,当然也可以为空,即没有任何参数。而 LaTeX 中的参数部分用的是以中括号包括起来的参数个数,即只定义了参数的个数而没有参数名。最后要比较的是函数体。Java 中的函数体内容非常丰富,而 LaTeX 则相对简单,因为 LaTeX 函数体仅服务于要排版的内容及其格式。

在此基础上,下面将通过 3 个函数来说明 LaTeX 中函数的使用,其中分别定义了 0 个参数、1 个参数和 2 个参数。研究中的案例描述如图 4 所示。

```
\newcommand{\noparfunction}{This function is invoked!}
\newcommand{\oneparfunction}[1]{Its parameter value is #1.}
\newcommand{\twoparfunction}[2]{#1#2}
```

图 4 具有不同参数的函数定义

Fig. 4 Definition of functions with different parameters

在图 4 中,函数 \noparfunction 没有参数,函数体是输出一个字符串,证明该函数被调用。函数 \oneparfunction 具有一个参数,函数体是输出字符串和该函数的参数值。函数 \twoparfunction 具有 2 个参数,函数体的功能是直接执行 2 个参数。

2.2 函数的使用

接下来调用上面定义的 3 个函数,来说明各函数的使用方式。案例描述如图 5 所示。

```
\noparfunction\par
\oneparfunction{3}\par
\twoparfunction{This function is invoked!}{Its parameter value is 3.}\par
\twoparfunction{\noparfunction}{\oneparfunction{3}}
```

图 5 函数的使用

Fig. 5 Usage of functions