

# LaTeX伪代码写法总结

原创

Zerg\_Wang

于 2021-02-11 13:46:57 发布

6425

收藏 43

编辑 版权

分类专栏： Management

文章标签： latex

伪代码



Management 专栏收录该内容

0 订阅

10 篇文章

## 1. 伪代码 所用包

一般会接触到的包有algorithm、algorithmic、algorithmicx、algorithm2e这四种包。

algorithm用于给伪代码提供一个浮动体环境，防止其换页或其他因素导致的内容中断，从而跨页显示。

algorithmic用于编辑伪代码的内容，一些for、while、if等语句通过该包中的命令进行编写。

algorithmicx可以看作algorithmic的升级版，提供了一些自定义命令

algorithm2e则是独立于algorithmic和algorithmicx的另一套伪代码环境，两套环境语法、排版上均不相同，**本篇博文聚焦于algorithmicx环境中所用的一些简单命令，旨在简单快速上手LaTeX伪代码。**

阅读此博文前，请先熟悉LaTeX基本语法，可参考本人博文：[https://blog.csdn.net/Zerg\\_Wang/article/details/104120617](https://blog.csdn.net/Zerg_Wang/article/details/104120617)

## 2. 环境构筑

### algorithmic环境

\begin{algorithm}后面跟着的“H”是指定伪代码浮动体的位置，语法规则与普通浮动体一致。

```
1 \documentclass{article}
2 \usepackage{algorithm}
3 \usepackage{algorithmic}
4
5 \begin{document}
6 \begin{algorithm}[H]
7   \caption{title}%标题
8   \label{alg1}%标签
9   \begin{algorithmic}
10    %这里是伪代码内容
11   \end{algorithmic}
12 \end{algorithm}
13 \end{document}
```

### algorithmicx环境

若要使用algorithmicx的语法环境，调用包的时候将algorithmic包替换为一个叫algpseudocode的包即可（该包会自动调用algorithmicx包），其他地方与algorithmic的环境一致。

## 3. 语法规则

这里介绍较为常用的algorithmicx的语法命令（**algorithmic命令与algorithmicx类似，区别是：前者的命令全为大写，后者仅首字母大写**）

### \State <text>

用于一般内容显示，当然不用该命令直接打内容也可以，但\State会为内容自动且正确地缩进，无论外面嵌套了多少个if或是循环。

### \Return <text>

返回语句，一般配合\State使用：`\State \Return <text>`

### \Comment {<text>}

注释内容，一般跟在\State命令正常内容之后，注释的内容在排版时是右对齐的。

条件判断语句

```
\If{<condition>} <text> \EndIf

\If{<condition>} <text> \Else <text> \EndIf

\If{<condition>} <text> \ElsIf{<condition>} <text> \Else <text> \EndIf
```

其中\ElsIf语句可以多重嵌套

效果展现：

```
if  $a > b$  then
     $max = a$ 
else if  $a < b$  then
     $max = b$ 
else
    same
end if
```

```
1 | \If{$a>b$}
2 |   \State $max = a$
3 | \ElsIf{$a<b$}
4 |   \State $max = b$
5 | \Else
6 |   \State same
7 | \EndIf
```

循环语句

```
\For{<condition>} <text> \EndFor

\ForAll{<condition>} <text> \EndFor

\While{<condition>} <text> \EndWhile

\Repeat <text> \Until{<condition>}

\Loop <text> \EndLoop
```

输入输出语句

```
\Require <text>

\Ensure <text>
```

当然，如果想改成“Input”“Output”形式，或者“输入”“输出”这种形式，可以在导言区重新定义该命令：

```
1 | \renewcommand{\algorithmicrequire}{\textbf{自定义输入}}
2 | \renewcommand{\algorithmicensure}{\textbf{自定义输出}}
```

函数体

```
\Function{<name>}{<params>} <text> \EndFunction
```

特殊字符

以下字符仅用于algorithmic环境中

```
\TRUE \FALSE \AND \OR \XOR \NOT
```

4. 其他

## 显示行号

```
\begin{algorithmic}[1]
```

“1”表示每行均显示行号，如果填“2”，意思是每2行显示一次行号。

## 标题与标号更改

如果不作额外更改，标题一行会显示为：

---

**Algorithm 1** title

如果有多个伪代码，标号为2或者其他数字，则可以在编辑标题内容的语句（环境区）前加：

```
\setcounter{algorithm}{1}
```

命令中的数字为1，显示的数字则为1+1=2，以次类推。

**Algorithm 2** title

如果连标题中的“Algorithm”也要更改为其他内容，可在导言区自定义标签：

```
\floatname{algorithm}{自定义名}
```

## 5. 参考资料

<https://zhuanlan.zhihu.com/p/145195565>

<http://hustsxh.is-programmer.com/posts/38801.html>