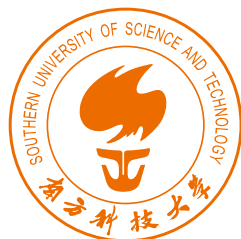


分类号 _____

编 号 _____

UDC _____

密 级 _____



南方科技大学

本科生毕业设计（论文）

题 目： _____ 南方科技大学学位论文模板

_____ 一个 Typst 模版

姓 名： _____ 张三

学 号： _____ 12010000

院 系： _____ 计算机科学与工程系

专 业： _____ 计算机科学与技术

指导教师： _____ 李四 教授

二零二四年六月 X 日

诚信承诺书

1. 本人郑重承诺所呈交的毕业设计（论文），是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。
2. 除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。
3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。
4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

作者签名：

二零二四年六月 X 日

南方科技大学学位论文模板

——一个 Typst 模版

张三

计算机科学与工程系 指导教师：李四 教授

[摘要]：本文介绍了 sustech-ug-thesis-typst 文档模板所提供的功能。

[关键词]：Typst；模板

[Abstract]: This document introduces the features of the sustech-ug-thesis-typst template.

[Keywords]: Typst; Template

目录

1 基本功能	1
1.1 标题	1
1.1.1 三级标题	1
1.2 粗体与斜体	2
1.3 脚注	2
1.4 图片	3
1.5 表格	3
1.6 公式	4
1.7 代码块	5
1.8 有序段落	7
1.9 参考文献	7
2 理论	8
2.1 理论一	8
3 展望	8
参考文献	10
附录	11
A1 关于 Typst	11
A1.1 在附录中插入图片和公式等	11
致谢	13

1 基本功能

1.1 标题

Typst 中的标题使用 `=` 表示，其后跟着标题的内容。`=` 的数量对应于标题的级别。

除了这一简略方式，也可以通过 `heading` 函数自定义标题的更多属性。具体可以参考文档中的有关内容。

下面是一个示例：

表 1 这是表序

代码	渲染结果
<code>#heading(level: 2, numbering: none, outlined: false, "二级标题")</code>	二级标题
<code>#heading(level: 3, numbering: none, outlined: false, "三级标题")</code>	三级标题
<code>#heading(level: 4, numbering: none, outlined: false, "四级标题")</code>	四级标题
<code>#heading(level: 5, numbering: none, outlined: false, "五级标题")</code>	五级标题

1.1.1 三级标题

1.1.1.1 四级标题

本模板目录的默认最大深度为 3，即只有前三级标题会出现在目录中。如果需要更深的目录，可以更改 `outlinedepth` 设置。

1.2 粗体与斜体

与 Markdown 类似，在 Typst 中，使用 `*...*` 表示粗体，使用 `_..._` 表示斜体。下面是一个示例：

表 2 这是一个三线表样例

代码	渲染结果
<code>*bold*</code> and <code>_italic_</code> are very simple.	bold and <i>italic</i> are very simple.

由于绝大部分中文字体只有单一字形，这里遵循 PKUTHSS 的惯例，使用**黑体**表示粗体，楷体表示斜体。但需要注意的是，由于语法解析的问题，`*...*` 和 `_..._` 的前后可能需要空格分隔，而这有时会导致不必要的空白。如果不希望出现这一空白，可以直接采用 `#strong` 或 `#emph`。

表 3 表序

代码	渲染结果
对于中文情形， <code>*使用 * 加粗*</code> 会导致额外的空白， <code>#strong</code> [使用 <code>\#strong</code> 加粗]则不会。	对于中文情形， 使用 * 加粗 会导致额外的空白， 使用 #strong 加粗 则不会。

1.3 脚注

从 v0.4 版本开始，Typst 原生支持了脚注功能。本模板中，默认每一章节的脚注编号从 1 开始。

表 4

代码	渲染结果
Typst 支持添加脚注 <code>#footnote</code> [这是一个脚注。]。	Typst 支持添加脚注 ¹ 。

1.4 图片

在 Typst 中插入图片的默认方式是 `image` 函数。如果需要给图片增加标题，或者在文章中引用图片，则需要将其放置在 `figure` 中，就像下面这样：

表 5

代码	渲染结果
<pre>#figure(image("resources/images/1-writing-app.png", width: 100%), caption: "Typst 网页版界面",) <web></pre>	 <p>图 1 Typst 网页版界面</p>

图 1 展示了 Typst 网页版的界面。更多有关内容，可以参考节 A1。代码中的 `<web>` 是这一图片的标签，可以在文中通过 `@web` 来引用。

1.5 表格

在 Typst 中，定义表格的默认方式是 `table` 函数。但如果需要给表格增加标题，或者在文章中引用表格，则可以将其放置在 `tbl` 中。需要注意的是下文表格中使用的 `tablex` 函数是第三方包，用来实现更灵活的表格如三线表等。它的语法

¹ 这是一个脚注。

和自带的 `table` 一样, 如果 不需要高级功能的话可以互相替换。`tbl` 函数的签名是 `tbl(table, caption: "", source: "")`, 其中 `table` 是要展示的表格, 可以是 `table` 或 `tablex` 的返回值, 第二个参数是表序, 可以省略, 第三个参数是数据来源, 可以省略。

表 6 这是表序

代码	渲染结果																								
<div>代码 1 默认表格</div> <pre>#tbl(tablex(columns: (auto, auto, auto, auto), inset: 10pt, align: horizon, [*姓名*],[*职称*],[*工作单位*],[*职责*], [李四],[教授],[北京大学],[主席], [王五],[教授],[北京大学],[成员], [赵六],[教授],[北京大学],[成员], [钱七],[教授],[北京大学],[成员], [孙八],[教授],[北京大学],[成员],), caption: "答辩委员会名单", source: "这是数据来源",) <tablex></pre>	<div>表 7 答辩委员会名单</div> <table><tr><th>姓名</th><th>职称</th><th>工作单位</th><th>职责</th></tr><tr><td>李四</td><td>教授</td><td>北京大学</td><td>主席</td></tr><tr><td>王五</td><td>教授</td><td>北京大学</td><td>成员</td></tr><tr><td>赵六</td><td>教授</td><td>北京大学</td><td>成员</td></tr><tr><td>钱七</td><td>教授</td><td>北京大学</td><td>成员</td></tr><tr><td>孙八</td><td>教授</td><td>北京大学</td><td>成员</td></tr></table> <div>数据来源：这是数据来源</div>	姓名	职称	工作单位	职责	李四	教授	北京大学	主席	王五	教授	北京大学	成员	赵六	教授	北京大学	成员	钱七	教授	北京大学	成员	孙八	教授	北京大学	成员
姓名	职称	工作单位	职责																						
李四	教授	北京大学	主席																						
王五	教授	北京大学	成员																						
赵六	教授	北京大学	成员																						
钱七	教授	北京大学	成员																						
孙八	教授	北京大学	成员																						

对应的渲染结果如所示。代码中的 `<tablex>` 是这一表格的标签, 可以在文中通过 `@tablex` 来引用。如果需要三线表, 可以在 `tablex` 中传入参数 `auto-vlines: false`, 如同的代码所示。

1.6 公式

式 1 是一个公式。代码中的 `<eq>` 是这一公式的标签, 可以在文中通过 `@eq` 来引用。

表 8

代码	渲染结果
<code>\$ E = m c^2 \$ <eq></code>	$E = mc^2$ 1

式 2 是一个多行公式。

表 9

代码	渲染结果
<pre>\$ sum_(k=0)^n k &= 1 + ... + n \ &= (n(n+1)) / 2 \$ <eq2></pre>	$\sum_{k=0}^n k = 1 + \dots + n$ $= \frac{n(n+1)}{2}$ 2

式 3 到式 6 中给出了更多的示例。

表 10

代码	渲染结果
<code>\$ frac(a^2, 2) \$ <eq3></code>	$\frac{a^2}{2}$ 3
<code>\$ vec(1, 2, delim: "[") \$</code>	$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ 4
<code>\$ mat(1, 2; 3, 4) \$</code>	$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ 5
<pre>\$ lim_x = op("lim", limits: #true)_x \$ <eq6></pre>	$\lim_x = \lim_x$ 6

1.7 代码块

像 Markdown 一样，我们可以在文档中插入代码块：

表 11 代码

代码	渲染结果
----	------

<pre> ```c int main() { printf("Hello, world!"); return 0; } ``` </pre>	<pre> int main() { printf("Hello, world!"); return 0; } </pre>
---	--

如果想要给代码块加上标题，并在文章中引用代码块，可以使用本模板中定义的 `codeblock` 命令。其中，`caption` 参数用于指定代码块的标题，`outline` 参数用于指定代码块显示时是否使用边框。下面给出的代码 2 是一个简单的 Python 程序。其中的 `<code>` 是这一代码块的标签，意味着这一代码块可以在文档中通过 `@code` 来引用。

表 12

代码	渲染结果
<pre> #codeblock(```python def main(): print("Hello, world!") ``` , caption: "一个简单的 Python 程序", outline: true,) <code> </pre>	<div> <div>代码 2 一个简单的 Python 程序</div> <div> <pre> def main(): print("Hello, world!") </pre> </div> </div>

代码 3 中给出了本模板中定义的 `codeblock` 命令的实现。

代码 3 codeblock 命令的实现

```
#let codeblock(raw, caption: none, outline: false) = {
  figure(
    if outline {
      rect(width: 100%)[
        #set align(left)
        #raw
      ]
    } else {
      set align(left)
      raw
    },
    caption: caption, kind: "code", supplement: ""
  )
}
```

1.8 有序段落

对于内容较长的列表，每项条目的宽度均和列表的第一行相等，即每一行都会缩进。有时如果需要对段落进行标号而不想让整段内容缩进的话，可以使用 `numberedpar` 命令实现。

表 13

有序段落	渲染结果
<pre>#numberedpar([本人郑重承诺所呈交的毕业设计（论文），是在 导师的指导下]， [独立进行研究工作所取得的成果，所有数据、图 片资料均真实可靠]，)</pre>	<p>(1) 本人郑重承诺所呈交的毕业设计 (论文)，是在导师的指导下</p> <p>(2) 独立进行研究工作所取得的成果， 所有数据、图片资料均真实可靠</p>

1.9 参考文献

Typst 支持 BibLaTeX 格式的 .bib 文件，同时也新定义了一种基于 YAML 的文献引用格式。要想在文档中引用参考文献，需要在文档中通过调用 `bibliography` 函数来引用参考文献文件。下面是一个示例：

表 14

代码	渲染结果
可以像这样引用参考文献: <code>@wang2010guide</code> 和 <code>@kopka2004guide</code> 。 <code>#biblio("ref.bib")</code>	可以像这样引用参考文献: [1] 和 [2]。

注意代码中的 "ref.bib" 也可以是一个数组，比如 ("ref1.bib", "ref2.bib")。

2 理论

2.1 理论一

让我们首先回顾一下 1 中的部分公式：

$\frac{a^2}{2}$	7
$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$	8
$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$	9
$\lim_x = \lim_x$	10

如式 7 所示，它是一个公式

3 展望

目前本模板还有一些不足之处，有待进一步完善：

- 参考文献格式，特别是中文参考文献的格式不完全符合学校有关规定。

Discord 上的这个对话显示，Typst 有关功能还在开发中。待有关接口对外开放后，本模板将会进行相应的适配。

参考文献

- [1] 王晓华, 闫其涛, 程智强, 和 张睿, 《科技论文中文摘要写作要点分析》, 编辑学报, 期 S1, 页 53–55, 2010.
- [2] H. Kopka, P. W. Daly, 和 S. Rahtz, *Guide to LATEX*, 卷 4. Addison-Wesley Boston, MA, 2004.

附录

A1 关于 Typst

A1.1 在附录中插入图片和公式等

附录中也支持脚注¹。

附录中也可以插入图片，如图 2。

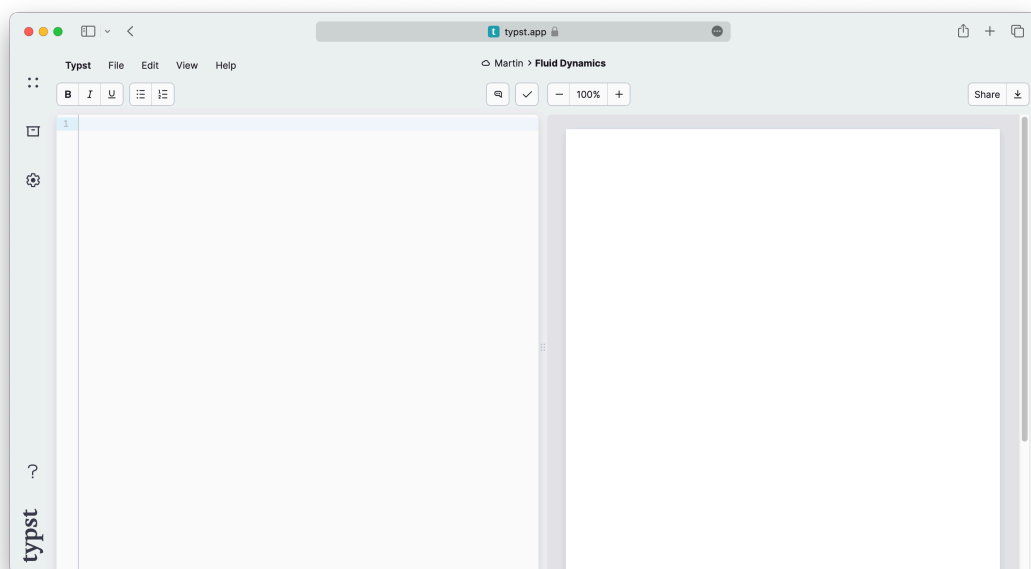


图 2 Typst 网页版界面

附录中也可以插入公式，如式 11。

表 15

代码	渲染结果
----	------

¹ 这是一个附录中的脚注。

<pre> \$ S = pi r^2 \$ <appendix-eq> \$ mat(1, 2, ..., 10; 2, 4, ..., 20; 3, 6, ..., 30; dots.v, dots.v, dots.down, dots.v; 10, 20, ..., 100) \$ \$ cal(A) < bb(B) < frac(C) < mono(D) < sans(E) < serif(F) \$ \$ bold(alpha < beta < gamma < delta < epsilon) \$ \$ upright(zeta < eta < theta < iota < kappa) \$ \$ lambda < mu < nu < xi < omicron \$ \$ bold(Sigma < Tau) < italic(Upsilon < Phi) < Chi < Psi < Omega \$ </pre>	<div> $S = \pi r^2$ K </div> <div> $\begin{pmatrix} 1 & 2 & \dots & 10 \\ 2 & 4 & \dots & 20 \\ 3 & 6 & \dots & 30 \\ \vdots & \vdots & \ddots & \vdots \\ 10 & 20 & \dots & 100 \end{pmatrix}$ L </div> <div> $\mathcal{A} < \mathbb{B} < \mathfrak{C} < \mathsf{D} < \textit{E} < \textbf{F}$ M </div> <div> $\alpha < \beta < \gamma < \delta < \varepsilon$ N </div> <div> $\zeta < \eta < \theta < \iota < \kappa$ O </div> <div> $\lambda < \mu < \nu < \xi < o$ P </div> <div> $\Sigma < \mathbf{T} < \mathcal{U} < \Phi < \mathbf{X} < \Psi < \Omega$ Q </div>
--	--

式 18 是一个非常复杂的公式的例子：

表 16

代码	渲染结果
<pre> \$ vec(overline(underbracket(underline(1 + 2) + overbrace(3 + dots.c + 10, "large numbers"), underbrace(x + norm(y), y^(w~u) - root(t, z))))), dots.v, u)^(frac(x + 3, y - 2)) \$ <complex> </pre>	<div> $\left(\frac{\overbrace{1 + 2 + 3 + \dots + 10}^{\text{large numbers}}}{\underbrace{\frac{x + \ y\ }{y^{w^u} - \sqrt[t]{z}}}_{\vdots} u} \right)^{\frac{x+3}{y-2}}$ R </div>

附录中也可以插入代码块，如代码 1。

代码 1 一个简单的 Rust 程序

```

fn main() {
    println!("Hello, world!");
}

```

致谢

感谢 Typst 开发者的辛勤付出。