

Typst 简明使用教程

卓能文

2023-11-29

目 录

1 Typst 简介	1
2 Typst 安装	1
3 Typst 使用	1
3.1 创建文件	1
3.2 章节设置	1
3.3 显示图片	2
3.3.1 设置宽度:	2
3.3.2 居中显示:	2
3.3.3 设置标题:	3
3.3.4 多图并列	3
3.3.5 多图并列 (带标题)	4
3.3.6 多图并列含间距 (带标题)	4
3.3.7 多图并列 (带子标题)	5
3.3.8 多图并列 (带子标题、子图无编号)	5
3.4 显示表格	6
3.5 显示公式	6
3.6 显示代码	7
3.6.1 添加标题	7
3.6.2 居左显示	7
3.6.3 显示代码文件	8
3.7 标签与引用	8
3.8 参考文献设置	9
4 写在最后	9
A 附源码	9
A.1 article 模板	9
A.2 本文档源码	12
参考文献	19

图形列表

图 3.1: 玫瑰	3
图 3.2: 多图并列 (带标题)	4
图 3.3: 多图并列含间距 (带标题)	5
图 3.4: 多图并列 (带子标题)	5
图 3.5: 玫瑰 1	5
图 3.6: 玫瑰 2	5
图 3.7: 多图并列 (带子标题、子图无编号)	6
图 3.8: 玫瑰	8

表格列表

表 3.1: 示例表格	6
-------------------	---

代码列表

代码 3.1: 计算斐波纳契	7
代码 3.2: 计算斐波纳契	8
代码 3.3: 计算斐波纳契	8

1 Typst 简介

Typst 是撰写任何长篇文本（如论文、文章、科学论文、书籍、报告和家庭作业）的优秀工具。此外，Typst 非常适合于编写任何包含数学符号的文档，例如在数学、物理和工程领域的论文。最后，由于其强大的风格化和自动化功能，它是任何一组具有共同风格的文件的绝佳选择，例如丛书。Typst 文档风格和 md 文档类似，所以很容易上手，同时内置了强大的脚本功能及较多的排版原语，因此，能比较轻松完成优质文档的撰写及排版工作。

2 Typst 安装

Typst 的本地安装非常简单，直接从 <https://github.com/typst/typst/releases> 下载适合自己操作系统的版本，解压到适当的地方即完成安装。另外，也可以在 <https://typst.app> 上注册账号，在线编辑 typst 文档，并下载生成的 PDF 文档。

编辑器建议采用 visual studio code，并安装 Typst LSP 和 Typst Preview 插件。

3 Typst 使用

3.1 创建文件

新建文本文档，以 .typ 为后缀。建议克隆 <https://github.com/soarowl/typst.git> 到本地，并将其中的 article.typ 复制到文档所在的目录，并适当进行修改。然后在文档头部添加如下内容：

```
1 #import "article.typ":*
2
3 #show: article.with(title: "Typst 简明使用教程", authors: ("卓能文",))
```

3.2 章节设置

格式有点类似 markdown，比较简单：

```
1 == 第一章
2 内容
3
4 == 第一节
5 内容
6
7 == 第二节
8 内容
9
10 == 第三节
11 内容
12
13 == 第二章
14 == 第一节
15
```

16
17 == 第二节
18 内容
19
20 == 第三章

3.3 显示图片

建议将图片保存在一个特定的目录，如 `images`、`img` 之类的地方。

```
1 #image("images/rose.jpg")
```

typst



3.3.1 设置宽度:

```
1 #image("images/rose.jpg", width: 50%)
```

typst



3.3.2 居中显示:

```
1 #align(center,image("images/rose.jpg", width: 50%))
```

typst



3.3.3 设置标题:

```
1 #figure(  
2   caption: [玫瑰],  
3   image("images/rose.jpg", width: 50%)  
4 )
```

typst



图 3.1: 玫瑰

注：放入 `#figure` 命令中的图片同时会在图形列表中出现。

3.3.4 多图并列

```
1 #grid(  
2   columns: (1fr, 1fr),  
3   image("images/rose.jpg"),  
4   image("images/rose.jpg"),  
5 )
```

typst



3.3.5 多图并列（带标题）

```
1 #figure(typst
2   caption: [多图并列（带标题）],
3   grid(
4     columns: (1fr, 1fr),
5     image("images/rose.jpg"),
6     image("images/rose.jpg"),
7   )
8 )
```



图 3.2: 多图并列（带标题）

3.3.6 多图并列含间距（带标题）

```
1 #figure(typst
2   caption: [多图并列含间距（带标题）],
3   grid(
4     columns: (1fr, 1fr),
5     gutter: 10pt,
6     image("images/rose.jpg"),
7     image("images/rose.jpg"),
8   )
9 )
```




图 3.3: 多图并列含间距（带标题）

3.3.7 多图并列（带子标题）

```

1 #figure(
2   caption: [多图并列（带子标题）],
3   grid(
4     columns: (1fr, 1fr),
5     gutter: 10pt,
6     figure(
7       caption: [玫瑰 1],
8       image("images/rose.jpg")
9     ),
10    figure(
11      caption: [玫瑰 2],
12      image("images/rose.jpg")
13    ),
14  )
15 )
    
```



图 3.5: 玫瑰 1



图 3.6: 玫瑰 2

图 3.4: 多图并列（带子标题）

3.3.8 多图并列（带子标题、子图无编号）

```

1 #figure(
2   caption: [多图并列（带子标题、子图无编号）],
3   grid(
4     columns: (1fr, 1fr),
5     gutter: 10pt,
6     [#image("images/rose.jpg")玫瑰 1],
7     [#image("images/rose.jpg")玫瑰 2],
8   )
    
```



```
)
9 )
```



玫瑰 1



玫瑰 2

图 3.7: 多图并列 (带子标题、子图无编号)

3.4 显示表格

```
1 #figure(
2   caption: [示例表格],
3   kind: table,
4   supplement: "表",
5   ```tbl
6     Rx   Nx
7     Rx   Nx.
8   _
9   software|version
10  _
11     AFL|2.39b
12     Mutt|1.8.0
13     Ruby|1.8.7.374
14 TeX Live|2015
15 _
16 ```
17 )
```

表 3.1: 示例表格

software	version
AFL	2.39b
Mutt	1.8.0
Ruby	1.8.7.374
TeX Live	2015

注: 由于目前 Typst 中有 bug, 显示表格时, 必须加上 `kind` 和 `supplement` 字段。

更多用法请参考 <https://github.com/maxcrees/tbl.typ>

3.5 显示公式

```
1 勾股定理可用公式:  $a^2 + b^2 = c^2$  表示。
```

勾股定理可用公式: $a^2 + b^2 = c^2$ 表示。

```
1 $ \sum_{k=1}^n k = (n(n+1)) / 2 $
```

$$\sum_{k=1}^n k = \frac{n(n+1)}{2} \quad (1)$$

Typst 默认只能显示一级公式，不能按章节重新计数，可采用第三方包 `i-figured` 实现，本模板已经内置。格式请参考 `latex` 相关文档。

3.6 显示代码

代码可以很容易添加，格式和 `markdown` 一样。

```
1  ```py3typst
2  def fibonacci(n):
3      if n <= 1:
4          return n
5      else:
6          return fibonacci(n - 1) + fibonacci(n - 2)
7  ```
```

```
1  def fibonacci(n):python
2      if n <= 1:
3          return n
4      else:
5          return fibonacci(n - 1) + fibonacci(n - 2)
```

3.6.1 添加标题

```
1  #figure(typst
2      caption: [计算斐波纳契],
3      ```py3
4      def fibonacci(n):
5          if n <= 1:
6              return n
7          else:
8              return fibonacci(n - 1) + fibonacci(n - 2)
9      ```
10 )
```

代码 3.1: 计算斐波纳契

```
1      def fibonacci(n):python
2          if n <= 1:
3              return n
4          else:
5              return fibonacci(n - 1) + fibonacci(n - 2)
```

3.6.2 居左显示

因为 `figure` 命令会导致代码居中显示，添加 `align(start)` 命令让代码居左：

```
1  #figure(typst
2      caption: [计算斐波纳契],
3      align(start)[
4          ```py3
5          def fibonacci(n):
6              if n <= 1:
7                  return n
8          ```
```

```
9         return fibonacci(n - 1) + fibonacci(n - 2)
10     """
11 ]
12 )
```

代码 3.2: 计算斐波纳契

```
1 def fibonacci(n):
2     if n <= 1:
3         return n
4     else:
5         return fibonacci(n - 1) + fibonacci(n - 2)
```

python

3.6.3 显示代码文件

在 Typst 文档中添加太多代码，导致可读性降低，也不便于后续采用相应的工具进行编辑、更新、管理与维护，建议将代码组织在一个文件夹中。

```
1 #figure(
2     caption: [计算斐波纳契],
3     align(start, raw(read("src/fibonacci.py"), lang: "py3", block: true))
4 )
```

typst

代码 3.3: 计算斐波纳契

```
1 def fibonacci(n):
2     if n <= 1:
3         return n
4     else:
5         return fibonacci(n - 1) + fibonacci(n - 2)
6
```

python

3.7 标签与引用

在被引用的图表等地方用<name>设置标签，在打算引用的地方输入@name 即可。name 后面如果是中文，添加一个空格可避免编译错误。在 i-figured 中，需要在引用的地方添加 fig:、tbl:、lst: 等，形成@fig:name 形式。如：图 3.8 所示。



图 3.8: 玫瑰

3.8 参考文献设置

参考文献设置也比较简单，只需在文件尾部加入 `#bibliography("example.yml", style: "gb-7114-2015-numeric")` 即可。yml 格式如下：

```
1 audio-descriptions:
2   affiliated:
3     - names: Taylor, Dallas
4       role: narrator
5   author: Barrows, Miellyn Fitzwater
6   date: 2017-02-07
7   issue: 8
8   parent:
9     author: Taylor, Dallas
10    title: Twenty Thousand Hertz
11    type: Audio
12  title: Audio Descriptions
13  type: Audio
14  url: https://www.20k.org/episodes/audio
15 barb:
16   author: Günther-Haug, Barbara
17   date: 2020
18   language: de-DE
19   location: München
20   publisher: MVG
21   title: Den Boden unter den Füßen verlieren
22   type: Book
```

在文章适当的地方插入 `@audio-descriptions[1]` 或 `@barb[2]` 这类的键。

4 写在最后

Typst 相对来说还比较新，功能和 latex 相比稍弱，同时还存在一些 bug。如果使用过程中有任何建议 或模板上有什么问题，请到 <https://github.com/soarowl/typst.git> 提要求。

A 附源码

A.1 article 模板

```
1 #import "@preview/codly:0.1.0":*
2 #import "@preview/i-figured:0.2.1"
3 #import "@preview/tbl:0.0.4"
4
5 #let article(
6
```

typst

```
    title: "",
7   authors: (),
8   date: datetime.today().display(),
9   logo: none,
10  body,
11 ) = {
12   set heading(numbering: "1.1")
13   set page(paper: "a4")
14   set text(font: ("Times New Roman", "SimSun"), lang: "zh")
15
16   //***** 图形、代码及表格列表设置
17   // this `level: 2` instructs the figure counters to be reset for every
18   // level 2 section, so at every level 1 and level 2 heading.
19   show heading: i-figured.reset-counters.with(level: 1)
20   // this `level: 2` instructs the figure numbering to include the first
21   // two levels of the current heading numbering.
22   // how this should behave with zeros can be set using `zero-fill`.
23   // e.g., setting `zero-fill: false` and `leading-zero: false` assures
24   // there is never a `0` in the numbering.
25   show figure: i-figured.show-figure.with(level: 1)
26   // master 版本不能编译
27   // show math.equation: i-figured.show-equation
28
29   set figure.caption(separator: ": ")
30   show figure.where(kind: raw): set figure.caption(position: top)
31   show figure.where(kind: table): set figure.caption(position: top)
32   //*****
33
34   //***** 代码框设置
35   let icon(codepoint) = {
36     box(height: 0.8em, baseline: 0.05em, image(codepoint))
37     h(0.1em)
38   }
39
40   let nameColor = rgb("#CE412B")
41
42   show: codly-init.with()
43
44   codly(languages: (
45     py3: (name: "python", icon: none, color: nameColor),
46     rs: (name: "rust", icon: none, color: nameColor),
47     typ: (name: "typst", icon: none, color: nameColor),
48     typc: (name: "typst", icon: none, color: nameColor),
49
```

```
    ))
50 //*****
51
52 //***** 表格设置
53 show: tbl.template.with(box: false, tab: "|")
54 //*****
55
56 //***** 标题页设置
57 // The page can contain a logo if you pass one with `logo: "logo.png"`.
58 if logo != none {
59   v(0.4fr)
60   align(right, image(logo, width: 26%))
61   v(9.6fr)
62 }
63
64 set align(center)
65
66 v(20fr, weak: true)
67 text(2em, weight: 700, title)
68
69 // 作者
70 v(1.5em, weak: true)
71 let by = authors.map(author => [#strong(author)]).join(" ")
72 text(1.2em, by)
73
74 v(70fr)
75 text(1.1em, date)
76
77 set align(left)
78 pagebreak()
79 //*****
80
81 //***** 页眉、页脚
82 set page(
83   header: [#h(1fr)#title#h(1fr)#line(length: 100%, stroke: 2pt)],
84   number-align: center,
85 )
86 //*****
87
88 //***** 目录
89 set page(numbering: "I")
90 counter(page).update(1)
91
92
```

```

93   show heading: set align(center)
94   it
95 }
96 outline(title: [目#h(2em)录], indent: true)
97 i-figured.outline(title: [图形列表])
98 i-figured.outline(target-kind: table, title: [表格列表])
99 i-figured.outline(target-kind: raw, title: [代码列表])
100 // master 版本不能编译
101 // i-figured.outline(target: math.equation, title: [公式列表])
102 pagebreak()
103 //*****
104
105 //***** 正文
106 set page(numbering: "1")
107 counter(page).update(1)
108 set par(first-line-indent: 2em, justify: true)
109 show par: set block(spacing: 0.65em)
110 // Workaround 3: Automatically add empty paragraph after heading
111 show heading: it => {
112   it
113   par(text(size: 0.35em, h(0.0em)))
114 } // Only works for paragraphs directly after heading
115
116 body
117 //*****
118 }
119

```

A.2 本文档源码

```

1  #import "article.typ":*
2
3  #show: article.with(title: "Typst 简明使用教程", authors: ("卓能文",))
4
5  // 加入公式编号
6  #set math.equation(numbering: "(1.1)")
7
8  // 正式版编译时不能访问图片
9  #show raw.where(block: true, lang: "typst-ex"): it => {
10   let txt = it.text
11   raw(lang: "typ", txt, block: true)
12   eval(txt, mode: "markup")
13 }
14

```

typst

15 `= Typst` 简介

16 `Typst` 是撰写任何长篇文本（如论文、文章、科学论文、书籍、报告和家庭作业）的优秀工具。此外，
 17 `Typst` 非常适合于编写任何包含数学符号的文档，例如在数学、物理和工程领域的论文。最后，
 18 由于其强大的风格化和自动化功能，它是任何一组具有共同风格的文件的绝佳选择，例如丛书。
 19 `Typst` 文档风格和 `md` 文档类似，所以很容易上手，同时内置了强大的脚本功能及较多的排版原语，因此，
 20 能比较轻松完成优质文档的撰写及排版工作。

21

22 `= Typst` 安装

23 `Typst` 的本地安装非常简单，直接从`<https://github.com/typst/typst/releases>`
 24 下载适合自己操作系统的版本，解压到适当的地方即完成安装。另外，也可以在`<https://typst.app>`
 25 上注册账号，在线编辑 `typst` 文档，并下载生成的 PDF 文档。

26

27 编辑器建议采用`visual studio code`，并安装`Typst LSP`和`Typst Preview`插件。

28

29 `= Typst` 使用

30

31 `== 创建文件`

32 新建文本文档，以`.typ`为后缀。建议克隆`<https://github.com/soarowl/typst.git>`到本地，
 33 并将其中的`article.typ`复制到文档所在的目录，并适当进行修改。然后在文档头部添加如下内容：

```
34 ```typ
35 #import "article.typ":*
36
37 #show: article.with(title: "Typst 简明使用教程", authors: ("卓能文",))
38 ```
```

39

40 `== 章节设置`

41 格式有点类似 markdown，比较简单：

```
42 ```typ
```

43 `= 第一章`

44 内容

45

46 `== 第一节`

47 内容

48

49 `== 第二节`

50 内容

51

52 `== 第三节`

53 内容

54

55 `= 第二章`

56 `== 第一节`

57

	内容
58	
59	== 第二节
60	内容
61	
62	= 第三章
63	```
64	
65	== 显示图片
66	建议将图片保存在一个特定的目录，如`images、img`之类的地方。
67	```typst-ex
68	#image("images/rose.jpg")
69	```
70	
71	=== 设置宽度:
72	```typst-ex
73	#image("images/rose.jpg", width: 50%)
74	```
75	
76	=== 居中显示:
77	```typst-ex
78	#align(center,image("images/rose.jpg", width: 50%))
79	```
80	
81	=== 设置标题:
82	```typst-ex
83	#figure(
84	caption: [玫瑰],
85	image("images/rose.jpg", width: 50%)
86)
87	```
88	
89	注：放入`#figure`命令中的图片同时会在图形列表中出现。
90	
91	=== 多图并列
92	```typst-ex
93	#grid(
94	columns: (1fr, 1fr),
95	image("images/rose.jpg"),
96	image("images/rose.jpg"),
97)
98	```
99	
100	

```
--- 多图并列（带标题）
101 ```typst-ex
102 #figure(
103   caption: [多图并列（带标题）],
104   grid(
105     columns: (1fr, 1fr),
106     image("images/rose.jpg"),
107     image("images/rose.jpg"),
108   )
109 )
110 ```
111
112 === 多图并列含间距（带标题）
113 ```typst-ex
114 #figure(
115   caption: [多图并列含间距（带标题）],
116   grid(
117     columns: (1fr, 1fr),
118     gutter: 10pt,
119     image("images/rose.jpg"),
120     image("images/rose.jpg"),
121   )
122 )
123 ```
124
125 === 多图并列（带子标题）
126 ```typst-ex
127 #figure(
128   caption: [多图并列（带子标题）],
129   grid(
130     columns: (1fr, 1fr),
131     gutter: 10pt,
132     figure(
133       caption: [玫瑰 1],
134       image("images/rose.jpg")
135     ),
136     figure(
137       caption: [玫瑰 2],
138       image("images/rose.jpg")
139     ),
140   )
141 )
142 ```
143
```

```

144 === 多图并列（带子标题、子图无编号）
145 ```typst-ex
146 #figure(
147   caption: [多图并列（带子标题、子图无编号）],
148   grid(
149     columns: (1fr, 1fr),
150     gutter: 10pt,
151     [#image("images/rose.jpg")玫瑰 1],
152     [#image("images/rose.jpg")玫瑰 2],
153   )
154 )
155 ```
156
157 == 显示表格
158 ```typst-ex
159 #figure(
160   caption: [示例表格],
161   kind: table,
162   supplement: "表",
163   `tbl
164     Rx   Nx
165     Rx   Nx.
166 _
167 software|version
168 _
169   AFL|2.39b
170   Mutt|1.8.0
171   Ruby|1.8.7.374
172 TeX Live|2015
173 _
174 ```
175 )
176 ```
177
178 注： 由于目前 Typst 中有 bug，显示表格时，必须加上`kind`和`supplement`字段。
179
180 更多用法请参考`https://github.com/maxcrees/tbl.typ`
181
182 == 显示公式
183 ```typst-ex
184 勾股定理可用公式：$a^2 + b^2 = c^2$ 表示。
185 ```
186

```

```

187 ```typst-ex
188 $ \sum_{k=1}^n k = (n(n+1)) / 2 $
189 ```
190
191 Typst 默认只能显示一级公式，不能按章节重新计数，可采用第三方包`i-figured`实现，
192 本模板已经内置。格式请参考`latex`相关文档。
193
194 === 显示代码
195 代码可以很容易添加，格式和 markdown 一样。
196 ```typst-ex
197 ```py3
198 def fibonacci(n):
199     if n <= 1:
200         return n
201     else:
202         return fibonacci(n - 1) + fibonacci(n - 2)
203 ```
204 ====
205
206 === 添加标题
207 ```typst-ex
208 #figure(
209     caption: [计算斐波纳契],
210 ```py3
211 def fibonacci(n):
212     if n <= 1:
213         return n
214     else:
215         return fibonacci(n - 1) + fibonacci(n - 2)
216 ```
217 )
218 ====
219
220 === 居左显示
221 因为`figure`命令会导致代码居中显示，添加`align(start)`命令让代码居左：
222 ```typst-ex
223 #figure(
224     caption: [计算斐波纳契],
225     align(start)[
226 ```py3
227 def fibonacci(n):
228     if n <= 1:
229         return n
230

```

```

    else:
231         return fibonaci(n - 1) + fibonaci(n - 2)
232 ```
233 ]
234 )
235 ```
236
237 === 显示代码文件
238 在 Typst 文档中添加太多代码，导致可读性降低，也不便于后续采用相应的工具进行编辑、更新、
239 管理与维护，建议将代码组织在一个文件夹中。
240 ```typst-ex
241 #figure(
242   caption: [计算斐波纳契],
243   align(start, raw(read("src/fibonaci.py"), lang: "py3", block: true))
244 )
245 ```
246
247 == 标签与引用
248 在被引用的图表等地方用`<name>`设置标签，在打算引用的地方输入`@name`即可。
249 name 后面如果是中文，添加一个空格可避免编译错误。在`i-figured`中，
250 需要在引用的地方添加`fig:`、`tbl:`、`lst:`等，形成`@fig:name`形式。如：@fig:rose 所示。
251
252 #figure(caption: [玫瑰], image("images/rose.jpg", width: 50%)) <rose>
253
254 == 参考文献设置
255 参考文献设置也比较简单，只需在文件尾部加入`#bibliography("example.yml", style:
256 "gb-7114-2015-numeric")`即可。yml 格式如下：
257 ```yaml
258 audio-descriptions:
259   affiliated:
260     - names: Taylor, Dallas
261       role: narrator
262   author: Barrows, Miellyn Fitzwater
263   date: 2017-02-07
264   issue: 8
265   parent:
266     author: Taylor, Dallas
267     title: Twenty Thousand Hertz
268     type: Audio
269   title: Audio Descriptions
270   type: Audio
271   url: https://www.20k.org/episodes/audio
272 barb:
273

```

```

author: Günther-Haug, Barbara
274 date: 2020
275 language: de-DE
276 location: München
277 publisher: MVG
278 title: Den Boden unter den Füßen verlieren
279 type: Book
280 ```
281
282 在文章适当的地方插入`@audio-descriptions`@audio-descriptions 或`@barb`@barb 这类的键。
283
284 == 写在最后
285 Typst 相对来说还比较新，功能和 latex 相比稍弱，同时还存在一些 bug。如果使用过程中有任何建议
286 或模板上有什么问题，请到`https://github.com/soarowl/typst.git`提要求。
287
288 #counter(heading).update(0)
289 #set heading(numbering: "A.1")
290 == 附源码
291
292 == article 模板
293 #raw(read("article.typ"), lang: "typ", block: true)
294
295 == 本文档源码
296 #raw(read("article_tutor.typ"), lang: "typ", block: true)
297
298 // master 版本编译错误。
299 // #bibliography("basic.yml", style: "gb-7114-2015-numeric")
300 #bibliography("basic.yml")
301

```

参考文献

- [1] M. F. Barrows, 《Audio Descriptions》, 期 8, 2017 年 2 月 7 日. [在线]. 载于: <https://www.20k.org/episodes/audio>
- [2] B. Günther-Haug, *Den Boden unter den Füßen verlieren*. München: MVG, 2020.