

# 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) .....	3
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) .....	4
3.3.8 多图并列 (带子标题、子图无编号) .....	5
3.4 显示表格 .....	5
3.5 显示公式 .....	6
3.6 显示代码 .....	6
3.6.1 添加标题 .....	6
3.6.2 居左显示(codly 包) .....	7
3.6.3 显示代码文件 .....	7
3.6.4 显示代码文件(sourcerer) .....	7
3.6.5 显示代码文件(codly) .....	8
3.7 标签与引用 .....	8
3.8 参考文献设置 .....	8
4 写在最后 .....	9
A 附源码 .....	9
A.1 article 模板 .....	9
A.2 本文档源码 .....	11
参考文献 .....	15

## 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) .....	4
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) .....	5
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 .....	7
代码 3.3: 计算斐波纳契 .....	7
代码 3.4: 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 内容
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(  
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 )
```

typst



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(  
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 )
```

typst



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  )
```

typst



```

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 )

```

typst



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   _

```

typst



```

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。
```

typst

勾股定理可用公式:  $a^2 + b^2 = c^2$  表示。

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

typst

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

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

注：目前，`codly` 显示代码有些问题，如部分代码在换页时被遮挡，超长代码不自动换行处理等。暂时换为 `sourcerer` 包进行代码显示。

代码可以很容易添加，格式和 `markdown` 一样。

```

1  ```py3
2  def fibonacci(n):
3      if n <= 1:
4          return n
5      else:
6          return fibonacci(n - 1) + fibonacci(n - 2)
7  ```

```

typst

```

1  def fibonacci(n):
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(
2      caption: [计算斐波纳契],
3  ```py3

```

typst

```

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

```

### 3.6.2 居左显示(codly 包)

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

```

typst

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)

```

### 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     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

```

### 3.6.4 显示代码文件(sourcerer)

```

1 #figure(
2     caption: [计算斐波纳契],

```

```
3 code(raw(read("src/fibonacci.py"), lang: "py3", block: true), lang: "python")
4 )
```

### 3.6.5 显示代码文件(codly)

```
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 3.4: 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
```

## 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/i-figured:0.2.1"
2  #import "@preview/sourcerer:0.2.1": code
3  #import "@preview/tbl:0.0.4"
4
5  #let article(
6    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   show raw.where(block: true): it => {
36     code(it)
37   }
38   //*****
39
40   //***** 表格设置
41   show: tbl.template.with(box: false, tab: "|")
42   //*****

```

```

43
44 //***** 标题页设置
45 // The page can contain a logo if you pass one with `logo: "logo.png"`.
46 if logo != none {
47   v(0.4fr)
48   align(right, image(logo, width: 26%))
49   v(9.6fr)
50 }
51
52 set align(center)
53
54 v(20fr, weak: true)
55 text(2em, weight: 700, title)
56
57 // 作者
58 v(1.5em, weak: true)
59 let by = authors.map(author => [#strong(author)]).join(" ")
60 text(1.2em, by)
61
62 v(70fr)
63 text(1.1em, date)
64
65 set align(left)
66 pagebreak()
67 //*****
68
69 //***** 页眉、页脚
70 set page(
71   header: [#h(1fr)#title#h(1fr)#line(length: 100%, stroke: 2pt)],
72   number-align: center,
73 )
74 //*****
75
76 //***** 目录
77 set page(numbering: "I")
78 counter(page).update(1)
79
80 show outline: it => {
81   show heading: set align(center)
82   it
83 }
84 outline(title: [目#h(2em)录], indent: true)
85 i-figured.outline(title: [图形列表])
86 i-figured.outline(target-kind: table, title: [表格列表])
87 i-figured.outline(target-kind: raw, title: [代码列表])
88 // master 版本不能编译
89 // i-figured.outline(target: math.equation, title: [公式列表])
90 pagebreak()
91 //*****
92
93 //***** 正文
94 set page(numbering: "1")
95 counter(page).update(1)
96 set par(first-line-indent: 2em, justify: true)
97 show par: set block(spacing: 0.65em)
98 // Workaround 3: Automatically add empty paragraph after heading
99 show heading: it => {
100   it
101   par(text(size: 0.35em, h(0.0em)))
102 } // Only works for paragraphs directly after heading
103
104 body
105 //*****
106 }
107

```

## 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     code(raw(txt, lang: "typ", block: true), lang: "typst")
12     eval(txt, mode: "markup")
13 }
14
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，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。
17
18 == Typst 安装
19 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。
20
21 编辑器建议采用`visual studio code`，并安装`Typst LSP`和`Typst Preview`插件。
22
23 == Typst 使用
24
25 == 创建文件
26 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：
27 ```typ
28 #import "article.typ":*
29
30 #show: article.with(title: "Typst 简明使用教程", authors: ("卓能文",))
31 ```
32
33 == 章节设置
34 格式有点类似 markdown，比较简单：
35 ```typ
36 = 第一章
37 内容
38
39 == 第一节
40 内容
41
42 == 第二节
43 内容
44
45 == 第三节
46 内容
47
48 = 第二章
49 == 第一节
50 内容
51
52 == 第二节
53 内容
54
55 = 第三章
56 ```
57

```

```

58 == 显示图片
59 建议将图片保存在一个特定的目录，如`images`、`img`之类的地方。
60 ```typst-ex
61 #image("images/rose.jpg")
62 ```
63
64 === 设置宽度:
65 ```typst-ex
66 #image("images/rose.jpg", width: 50%)
67 ```
68
69 === 居中显示:
70 ```typst-ex
71 #align(center, image("images/rose.jpg", width: 50%))
72 ```
73
74 === 设置标题:
75 ```typst-ex
76 #figure(
77   caption: [玫瑰],
78   image("images/rose.jpg", width: 50%)
79 )
80 ```
81
82 注：放入`#figure`命令中的图片同时会在图形列表中出现。
83
84 === 多图并列
85 ```typst-ex
86 #grid(
87   columns: (lfr, lfr),
88   image("images/rose.jpg"),
89   image("images/rose.jpg"),
90 )
91 ```
92
93 === 多图并列（带标题）
94 ```typst-ex
95 #figure(
96   caption: [多图并列（带标题）],
97   grid(
98     columns: (lfr, lfr),
99     image("images/rose.jpg"),
100    image("images/rose.jpg"),
101   )
102 )
103 ```
104
105 === 多图并列含间距（带标题）
106 ```typst-ex
107 #figure(
108   caption: [多图并列含间距（带标题）],
109   grid(
110     columns: (lfr, lfr),
111     gutter: 10pt,
112     image("images/rose.jpg"),
113     image("images/rose.jpg"),
114   )
115 )
116 ```
117
118 === 多图并列（带子标题）
119 ```typst-ex
120 #figure(
121   caption: [多图并列（带子标题）],
122   grid(
123     columns: (lfr, lfr),

```



```

124     gutter: 10pt,
125     figure(
126       caption: [玫瑰 1],
127       image("images/rose.jpg")
128     ),
129     figure(
130       caption: [玫瑰 2],
131       image("images/rose.jpg")
132     ),
133   )
134 )
135 ```
136
137 === 多图并列（带子标题、子图无编号）
138 ```typst-ex
139 #figure(
140   caption: [多图并列（带子标题、子图无编号）],
141   grid(
142     columns: (lfr, lfr),
143     gutter: 10pt,
144     [#image("images/rose.jpg")玫瑰 1],
145     [#image("images/rose.jpg")玫瑰 2],
146   )
147 )
148 ```
149
150 == 显示表格
151 ```typst-ex
152 #figure(
153   caption: [示例表格],
154   kind: table,
155   supplement: "表",
156   `tbl
157     Rx    Nx
158     Rx    Nx.
159   _
160   software|version
161   _
162     AFL|2.39b
163     Mutt|1.8.0
164     Ruby|1.8.7.374
165   TeX Live|2015
166   _
167   ```
168 )
169 ```
170
171 注： 由于目前 Typst 中有 bug，显示表格时，必须加上`kind`和`supplement`字段。
172
173 更多用法请参考`https://github.com/maxcrees/tbl.typ`
174
175 == 显示公式
176 ```typst-ex
177 勾股定理可用公式:  $a^2 + b^2 = c^2$  表示。
178 ```
179
180 ```typst-ex
181 $ \sum_{k=1}^n k = (n(n+1)) / 2 $
182 ```
183
184 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。
185
186 == 显示代码

```

```

187 注：目前，`codly` 显示代码有些问题，如部分代码在换页时被遮挡，超长代码不自动换行处理等。暂时换为`sourcerer`包进行代码
    显示。
188
189 代码可以很容易添加，格式和 markdown 一样。
190 ```typst-ex
191 ```py3
192 def fibonacci(n):
193     if n <= 1:
194         return n
195     else:
196         return fibonacci(n - 1) + fibonacci(n - 2)
197 ```
198 ```
199
200 === 添加标题
201 ```typst-ex
202 #figure(
203     caption: [计算斐波纳契],
204     ```py3
205     def fibonacci(n):
206         if n <= 1:
207             return n
208         else:
209             return fibonacci(n - 1) + fibonacci(n - 2)
210     ```
211 )
212 ```
213
214 === 居左显示(codly 包)
215 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：
216 ```typst-ex
217 #figure(
218     caption: [计算斐波纳契],
219     align(start)[
220     ```py3
221     def fibonacci(n):
222         if n <= 1:
223             return n
224         else:
225             return fibonacci(n - 1) + fibonacci(n - 2)
226     ```
227 ]
228 )
229 ```
230
231 === 显示代码文件
232 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。
233 ```typst-ex
234 #figure(
235     caption: [计算斐波纳契],
236     raw(read("src/fibonacci.py"), lang: "py3", block: true)
237 )
238 ```
239
240 === 显示代码文件(sourcerer)
241 ```typ
242 #figure(
243     caption: [计算斐波纳契],
244     code(raw(read("src/fibonacci.py"), lang: "py3", block: true), lang: "python")
245 )
246 ```
247
248 === 显示代码文件(codly)
249 ```typst-ex
250 #figure(

```

```

251   caption: [计算斐波纳契],
252   align(start, raw(read("src/fibonacci.py"), lang: "py3", block: true))
253 )
254 ```
255
256 == 标签与引用
257 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：@fig:rose 所示。
258
259 #figure(caption: [玫瑰], image("images/rose.jpg", width: 50%))
260
261 == 参考文献设置
262 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：
263 ```yaml
264 audio-descriptions:
265   affiliated:
266     - names: Taylor, Dallas
267       role: narrator
268   author: Barrows, Miellyn Fitzwater
269   date: 2017-02-07
270   issue: 8
271   parent:
272     author: Taylor, Dallas
273     title: Twenty Thousand Hertz
274     type: Audio
275   title: Audio Descriptions
276   type: Audio
277   url: https://www.20k.org/episodes/audio
278 barb:
279   author: Günther-Haug, Barbara
280   date: 2020
281   language: de-DE
282   location: München
283   publisher: MVG
284   title: Den Boden unter den Füßen verlieren
285   type: Book
286 ```
287
288 在文章适当的地方插入`@audio-descriptions`@audio-descriptions 或`@barb`@barb 这类的键。
289
290 == 写在最后
291 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。
292
293 #counter(heading).update(0)
294 #set heading(numbering: "A.1")
295 == 附源码
296
297 == article 模板
298 #raw(read("article.typ"), lang: "typ", block: true)
299
300 == 本文档源码
301 #raw(read("article_tutor.typ"), lang: "typ", block: true)
302
303 // master 版本编译错误。
304 // #bibliography("basic.yml", style: "gb-7114-2015-numeric")
305 #bibliography("basic.yml")
306

```

## 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.