
Typst 5.011 | An Example Assignment

张三 17113945

老师, 2024 夏 | 截止时间: 2024 年 5 月 11 日

一、开始

要开始使用此模板, 你需要

1. 安装必须的字体包, 包括:

- [IBM Plex](#)
- [Noto Serif SC](#)

或者, 你可以在文档开头的 `font` 参数修改字体设置:

- 1) 在终端 / 命令行输入 `typst fonts` 查看当前可用的字体;
- 2) 在正文开头使用

```
1 #let font = (  
2   main: "IBM Plex Sans",  
3   mono: "IBM Plex Mono",  
4   cjk: "Noto Serif SC",  
5   math: "IBM Plex Math",  
6   math-cjk: "Noto Serif SC",  
7 )
```

修改字体, 其中 `main` 为主要字体, `mono` 为等宽字体, `cjk` 为中文字体。然后在 `assignment_class` 函数中使用 `font: font` 来设置字体。

2. 下载 `template.typ` 并在你的文档开头中使用

```
1 #import "template.typ": *
```

来导入模板。

3. clone 本项目并在 `assignment_example.typ` 中进行修改。

二、特性

本模板基于 [gRox167 的 typst-assignment-template](#) 修改, 缝合了许多作者喜欢的特性, 包括:

1. 自动编号的问题块
2. 自定义标题的特殊块
3. 引入 `numbley` 包, 支持中文样式的标题编号
4. 美观整洁的排版

三、使用

1. 导入和配置

首先先在开头导入模板:

```
1 #import "template.typ": *
```

在文档的开头设置参数：

```
1 #let title = "An Example Assignment"
2 #let author = "hongjr03"
3 #let course_id = "Typst 5.011"
4 #let instructor = "John"
5 #let semester = "2024 Spring"
6 #let due_time = datetime(day: 11, month: 5, year: 2024)
7 #let id = "17113945"
```

以及使用的字体：

```
1 #let font = (
2   main: "IBM Plex Sans",
3   mono: "IBM Plex Mono",
4   cjk: "Noto Serif SC",
5   math: "IBM Plex Math",
6   math-cjk: "Noto Serif SC",
7 )
```

接着使用 `numbly` 包设置标题编号样式：

```
1 #set heading(
2   numbering: numbly(
3     "{1:一}、",
4     "{2:1}.",
5     "{2:1}.{3}.",
6   ),
7 )
```

参数中，`{*:1}` 的 `*` 代表标题的级别，`1` 代表标题的格式。`{1:一}`、代表一级标题的格式为一、，`{2:1}.` 代表二级标题的格式为 1.，`{2:1}.{3}.` 代表三级标题的格式为 1.1.。

注意，本模板默认去除了标题 numbering 后的空格，所以在设置标题编号时请注意空格的使用。如 `"{2:1}."` 的末尾有一个空格，这样在标题编号后会有一空格。

然后使用 `assignment_class` 函数生成文档：

```
1 #show: assignment_class.with(font: font, title, author, course_id, instructor,
   semester, due_time, id)
```

这样就创建了本文档的开头部分，以及后面每一页的页眉。

2. 正文

在正文部分，你可以按照格式自由组合问题块、特殊块、问题和解答块。

2.1. 问题块

如何使用问题块来创建问题？

你可以使用 `prob` 块来创建问题。

如果有连续的问题，如何处理？

你可以使用 `cprob` 块来创建问题。例如：

```
1 #cprob[这是一个问题][
2   这是问题的内容。
3 ]
```

会生成一个带编号的问题块。请看下面的例子。

问题 1：这是一个问题

这是问题的内容。

问题 2：这是另一个问题

这是另一个问题的内容。

有编号的问题块会自动编号。你可以通过下面这一行代码重置问题计数器，使下一个问题从 1 开始编号。

```
1 #problem_counter.update(0) // 重置问题计数器
```

问题 1：这是一个新的问题

这是新问题的内容。编号已经重置为 1。

2.2. 特殊块

特殊块允许你自定义标题、内容和背景色。在我去年的作业中，一般使用特殊块来作为“问题-解答”块以和作业的解答区分开。比如下面的例子。

如何使用特殊块？

你可以使用 `speci_block` 块来创建特殊块。

问题 1：这是一个问题和解答块**解决方式**

这是问题和解答块的内容。

同样的，你可以通过下面这一行代码重置问题和解答计数器，使下一个问题从 1 开始编号。

```
1 #prob-solution_counter.update(0) // 重置问题和解答计数器
```

问题 1：这是一个新的问题和解答块**解决方式**

这是新问题和解答块的内容。编号已经重置为 1。

3. 自定义

你可以在 `template.typ` 中自定义 `prob`、`cprob`、`cqa`、`prob_block`、`speci_block` 函数，以满足你的需求。

四、参数说明

assignment_class

模板的核心类，规范了文档的格式。

变量

```
assignment_class(  
  size: length,  
  title: string,  
  author: string,  
  course_id: string,  
  professor_name: string,  
  semester: string,  
  due_time: datetime,  
  id: string,  
  font: object,  
  lang: string,  
  region: string,  
  body: content  
) -> content
```

size `length`

字体大小。默认为 10.5pt。

默认值: 10.5pt

title `string`

文档的标题。

author `string`

作者。

course_id `string`

课程名。

professor_name `string`

教师名。

semester `string`

学期。

due_time `datetime`

截止时间。

id `string`

学号。

font `object`

字体。默认为 `default_font`。如果你想使用不同的字体，可以传入一个字典，包含 `main`、`mono`、`cjk`、`math` 和 `math-cjk` 字段。

默认值: `default_font`**lang** `string`语言。默认为 `zh`。默认值: `"zh"`**region** `string`地区。默认为 `cn`。默认值: `"cn"`**body** `content`

文档的内容。

cprob

创建一个有编号的问题描述块。

`cprob` [这是一个问题的描述。] [这是问题的解答。]

问题 2： 这是一个问题的描述。

这是问题的解答。

有编号的问题描述块会自动编号，如下一个例子会变成“问题 2”：

`cprob` [这是一个问题的描述。] [这是问题的解答。]

问题 3：这是一个问题的描述。

这是问题的解答。

变量

```
cprob(
  text: content,
  body: content
) -> content
```

text content

问题。

body content

问题的解答。

cqa

创建一个自动编号的带问题的特殊块。

`cqa` [这是一个问题。] [这是特殊块的内容。]

问题 2：这是一个问题。

解决方式

这是特殊块的内容。

变量

```
cqa(
  title: content,
  body: content,
  color: color
) -> content
```

title content

特殊块的问题。

body content

特殊块的内容。

color `[color]`

特殊块的背景色。默认为 `rgb("#1f84c7")`。

默认值: `rgb("#1f84c7")`

prob

创建一个有问题描述的块。

```
prob[这是一个问题的描述。][这是问题的解答。]
```

这是一个问题的描述。

这是问题的解答。

变量

```
prob(
  text: content,
  body: content
) -> content
```

text `[content]`

问题。

body `[content]`

问题的解答。问题的解答可以为空，如下面的例子：

```
prob[这是一个没有解答的问题。][ ]
```

这是一个没有解答的问题。

或者，也可以是一个空的块：

```
prob[这是一个没有解答的问题。][ ]
```

这是一个没有解答的问题。

prob_block

创建一个块。由于这个块的样式只有问题使用，所以叫它 `prob_block`。

```
prob_block[这是一个块。]
```

这是一个块。

变量

```
prob_block(body: content) -> content
```

body `content`

块的内容。

speci_block

创建一个特殊块。

```
speci_block[这是特殊块的标题。][这是特殊块的内容。]
```

这是特殊块的标题。

这是特殊块的内容。

变量

```
speci_block(  
  title: content,  
  body: content,  
  color: color  
) -> content
```

title `content`

块的标题。

body `content`

块的内容。

color `color`

块的边框和标题栏颜色。默认为 `rgb("#1f84c7")`。

默认值: `rgb("#1f84c7")`