

---

# 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

你说得对, 但《Typst》是一款 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do., 后面忘了。

— Lorem.

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，你可以使用以下块来创建问题、特殊块、问题和解答块：

**问题 1：如何使用问题块来创建问题？**

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，如本块所示。

**问题 2：我有好多个问题，不想依次编号，怎么办？**

使用 `cprob` 块时会自动编号问题，同时在导出的 PDF 文档中会显示问题的索引。

`cprob` 的 `c` 代表 counter，即计数器。你可以使用 `#problem_counter.update(0)` 来重置计数器，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：这就是效果。**

问题计数器已重置为 0，下一个问题将从 1 开始编号。

**不想要编号？**

你可以使用 `prob` 块来创建无编号的问题。例如：

```
1 #prob[这是一个无编号的问题][
2   这是问题的内容。
3 ]
```

会生成一个无编号的问题块，如本块所示。

**只想要问题，不需要内容？**

你可以使用 `prob` 块来创建无内容的问题。例如：

```
1 #prob[这是一个无内容的问题][ ]
```

会生成一个无内容的问题块，如下：

**这是一个无内容的问题**

或者，使用 `cprob` 块并留空内容：

```
1 #cprob[这是一个无内容的问题][ ]
```

会生成带编号的无内容问题块，如下：

**问题 2：这是一个无内容的问题**

当然，你也可以使用 `prob_block` 块来创建问题块。例如：

```
1 #prob_block[
2   这是一个问题块的内容。
3 ]
```

会生成一个只有内容部分的问题块，如本块所示。

### 我想要一个特殊块，怎么办？

你可以使用 `speci_block` 块来创建特殊块。例如：

```
1 #speci_block[这是一个特殊块][
2   这是特殊块的内容。
3 ]
```

会生成一个带标题的特殊块，如本块所示。

### 问题 1：我想要一个问题和解答的块，怎么办？

#### 解决方式

你可以使用 `cqa` 块来创建问题和解答的块。例如：

```
1 #cqa[这是一个问题和解答块][
2   这是问题和解答块的内容。
3 ]
```

会生成一个带编号的问题和解答块，如本块所示。

### 问题 2：需要注意的是：

#### 解决方式

问题和解答块的计数器和问题块的计数器是分开的，你可以使用

`#prob-solution_counter.update(0)` 来重置计数器，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：这就是效果。

#### 解决方式

问题和解答计数器已重置为 0，下一个问题将从 1 开始编号。

## 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。

The end.