

---

# Typst 5.011 | An Example Assignment

张三 17113945

老师, 2024 夏 | 2024 年 5 月 11 日

---

## 一、快速开始

要开始使用此模板, 你需要

1. 安装必须的字体包, 包括:
  - [IBM Plex Sans, Mono, Math](#)
  - [Noto Serif CJK SC](#)
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. 在文档开头设置参数, 包括标题、作者、课程名、教师名、学期、截止时间和学号;

```
1 | #show: assignment_class.with(  
2 |   title: "An Example Assignment", // 标题  
3 |   author: "张三", // 作者  
4 |   course: "Typst 5.011", // 课程名  
5 |   professor_name: "老师", // 教师名  
6 |   semester: "2024 夏", // 学期  
7 |   due_time: datetime(day: 11, month: 5, year: 2024), // 截止时间  
8 |   id: "17113945", // 学号  
9 | )
```

4. 开始写作!

## 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. 引入 `numbly` 包, 支持中文样式的标题编号
2. 引入 `zebraw` 包, 支持代码块高亮
3. 引入 `frame-it` 包, 支持自定义背景色的块
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 | #show: assignment_class.with(  
2 |   title: "An Example Assignment", // 标题  
3 |   author: "张三", // 作者  
4 |   course: "Typst 5.011", // 课程名  
5 |   professor_name: "老师", // 教师名  
6 |   semester: "2024 夏", // 学期  
7 |   due_time: datetime(day: 11, month: 5, year: 2024), // 截止时间  
8 |   id: "17113945", // 学号  
9 | )
```

```
2 | title: "An Example Assignment",
3 | author: "张三",
4 | course: "Typst 5.011",
5 | professor_name: "老师",
6 | semester: "2024 夏",
7 | due_time: datetime(day: 11, month: 5, year: 2024),
8 | id: "17113945",
9 | )
```

接下来即可开始写作。

## 2. 正文

此处对于块的使用较原来的模版有所改变，请勿混用。

模版使用 frame-it 包预定义了 prob 和 qna 两种块，分别用于问题和问答。使用方法如下：

```
1 | #prob[问题描述][
2 |   问题的回答
3 | ]
```

问题描述

问题 1

问题的回答

```
1 | #qna[问题描述][
2 |   问题的回答
3 | ]
```

问题描述

问答 1

问题的回答

如果不需要问题的描述，可以直接使用：

```
1 | #prob[][
2 |   问题的回答
3 | ]
```

问题 2

问题的回答

## 3. 代码块

模板使用 zebraw 包预定义了代码块，支持高亮。使用方法如下：

```
1 | #zebraw(
2 |   highlight-lines: (3, 4, 5, 6),
3 |   ```cpp
4 |   #include <iostream>
5 |   using namespace std;
6 |   int main() {
7 |     cout << "Hello, World!" << endl;
```

```

8 |     return 0;
9 | }
10 | ...
11 | )

```

```

1 | #include <iostream>
2 | using namespace std;
3 | int main() {
4 |     cout << "Hello, World!" << endl;
5 |     return 0;
6 | }

```

## 4. 自定义

### 4.1. 标题编号

可以在文档设置参数后使用 `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个空格。

### 4.2. 字体

先在终端 / 命令行输入 `typst fonts` 查看当前可用的字体，以在文档开头加入 `font` 参数修改字体设置以及使用的字体：

```

1 | #let font = (
2 |     main: "IBM Plex Sans",
3 |     mono: "IBM Plex Mono",
4 |     cjk: "Noto Serif SC",
5 |     emph-cjk: "KaiTi",
6 |     math: "IBM Plex Math",
7 |     math-cjk: "Noto Serif SC",
8 | )
9 |
10 | #show: assignment_class.with(
11 |     // ... 保持原有的参数
12 |     font: font,
13 | )

```

## 四、参数说明

### assignment\_class

模板的核心类，规范了文档的格式。

## 参数

```
assignment_class(  
  size: length,  
  title: string,  
  author: string,  
  course: 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。

默认值: `none`

**author**    `string`

作者。

默认值: `none`

**course**    `string`

课程名。

默认值: `none`

**professor\_name**    `string`

教师名。

默认值: `none`

**semester**    `string`

学期。

默认值: `none`

**due\_time** `datetime`

截止时间。

默认值: `none`

**id** `string`

学号。

默认值: `none`

**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。