

实验一 | 一个作为 Example 的实验

张三 23000000000

老师, 比较高端的课程, 2024 夏 | 2024 年 5 月 11 日

一、二改介绍

基于 [hongjr03](#) 学长的实验报告进行微调, 主要更改了样式以供自用。

感谢学长的精心制作和无私分享。

二、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:

经过个人修改, 更改了字体设定:

- Times New Roman
- JetBrainsMono NF
- Noto Serif CJK SC
- IBM Plex Math

原模板的要求如下:

- [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   subtitle: "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!

三、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 `numbly` 包, 支持中文样式的标题编号

4. 美观整洁的排版

四、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   subtitle: "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

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

3.1. 更多自定义

如果你需要更多的自定义，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。

3.2. 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:~}、",  
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:~}`、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。

3.3. 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   math: "IBM Plex Math",  
6   math-cjk: "Noto Serif SC",  
7 )  
8  
9 #show: assignment_class.with(  
10  // ... 保持原有的参数  
11  font: font,  
12 )
```

五、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,  
  subtitle: string,  
  author: string,  
  professor_name: string,  
  course: 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`

主标题（曾命名为 `course`）

默认值: `none`

subtitle `string`

文档的子标题。（曾命名为 title）

默认值: `none`**author** `string`

作者。

默认值: `none`**professor_name** `string`

教师名。

默认值: `none`**course** `string`

课程名。

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。

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。

参数

```
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。

参数

```
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。

参数

```
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。

参数

`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。

参数

```
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")`