常用功能速成练习

第 1 章 说明

下面演示自动标题标号的效果

第 1.1 节 标题

1.1.1. hao

1.1.1.1. hao

1.1.1.1.1. hao

1.1.1.1.1.1. hao

1.1.1.1.1.1.1. hao

1.1.1.1.1.1.1.1. hao

目前的字体目录使用环境变量实现，之后更换电脑需要配置 `TYPST_FONT_PATHS` 环境变量。

第 2 章 对齐

我是临时居中

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat.

第 3 章 表格

第 3.1 节 单位

`fr` 是一个神奇的单位，用于表示空间分配，在表格列宽上会有良好的体现。

表头 1	表头 2	表头 3
Lorem ipsum dolor.	Lorem ipsum dolor sit amet, consectetur.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Lorem ipsum dolor sit amet, consectetur adipiscing.	Lorem ipsum dolor sit amet, consectetur adipiscing.	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed.

第 4 章 图片

使用

来插入图片，这看起来会导致图片后面的文本被强制换行，并新起一段。我自己定义了一个 `no-indent` 的变量来实现这个效果。

第 5 章 代码块

现在是导入了美化样式的代码块。

```py
def function(string):
    print(f"Hello, {string}!")

if __name__ == "__main__":
    function("world!")
```

第 6 章 图形

`figure` 可以把几乎任何部分转为图形。

图 1　这是一张测试图片

值得注意的是，如果不将文档语言设置为中文，显示的就不是“图 1”，而是“Figure 1”。

表头 1	表头 2	表头 3
Lorem ipsum dolor.	Lorem ipsum dolor sit amet, consectetur.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Lorem ipsum dolor sit amet, consectetur adipiscing.	Lorem ipsum dolor sit amet, consectetur adipiscing.	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed.

表 1　这是一张测试表格

```py
def function(string):
    print(f"Hello, {string}!")

if __name__ == "__main__":
    function("world!")
```

代码 1　这是一段代码

根据上面的测试，我注意到，放入 `figure` 的内容总是居中的，对于表格中的内容尤其明显，但是经过 `codly` 美化的代码并没有出现这样的问题，我猜想是有类似继承和覆盖的机制。

另外，标号应该是自动计算[1]的。

```c
#include <stdio.h>
int main() {
    printf("Hello, world!\n");
    return 0;
```

代码 2　这是一段 C 语言代码

这里默认显示为“代码 2”，而我没有做任何修改。

[1]这里用来测试脚注

参考文献

是用了某些方法，使该标题没有编号了。

主要参考了 Typst 中文社区。