

Git 提交信息样式指南

点此查看文档英文版 ([pdf](#), [GitHub.io](#))

简介

该指南是你在项目进行过程中所需遵守的官方指南。优达学城的评估人员会根据该指南为你的项目打分。在数据分析的世界中，有很多“最佳”样式供你选择。因此，为了减少学生在项目过程中因选择何种样式所产生的困惑，我们强烈建议所有学生在其项目中遵循这个样式指南。

提交信息

信息结构

提交信息由三个不同的部分构成，这些部分均由空行分隔：标题、可选的消息体和可选的注释。其布局大致如此：

```
类型：主题  
消息正文  
注释
```

标题由消息类型和主题构成。

类型

类型位于在标题内，有以下几种可能：

- **feature**：新功能
- **fix**：错误修复

- **docs**: 文档修改
- **style**: 格式、分号缺失等，代码无变动
- **refactor**: 生产代码重构
- **test**: 测试添加、测试重构等，生产代码无变动
- **chore**: 构建任务更新、程序包管理器配置等，生产代码无变动。

标题

主题不得超过50个字符，首字母大写，末尾不加句号。以祈使语气描述提交的任务，而不是其已完成的任务。例如，使用 **change...**，而不是 **changed** 或 **changes**。

信息正文

并不是所有的提交信息都复杂到需要主体，因此这是可选内容，仅在提交信息需要一定的解释和语境时使用。消息体是用于解释提交任务的内容和原因，而不是方法。

在编写正文时，需要在标题和正文间加一个空行，且每行的内容应控制在72个字符内。

注释

注释是可选内容，用于引用问题跟踪的 ID。

提交信息示例

Feature: 总结变动的内容，保持在50个字符内

如有需要，使用更详细的说明性文字，将其大概控制在72个字符。在部分语境中，第一行被视为

提交信息的主题，余下的文本被视为主体。分隔总结与主体的空行十分重要（除非你完全忽略主体）；否则`log`、`shortlog`和`rebase`等多个工具容易发生混淆。

解释该提交信息所解决的问题，说明你进行该变动的原因，而不是方法（代码本身可以解释方法）。

该变动是否存在副作用或其他直觉性后果？在这里进行解释。

后续段落前需加空行。

可以列出要点

- 通常情况下，要点会使用空格加上连字符或星号，中间用空行分隔，但该规定存在差别。

如果你使用问题追踪，将其引用放在末尾，例如：

解决了问题：#123

另见：#456，#789