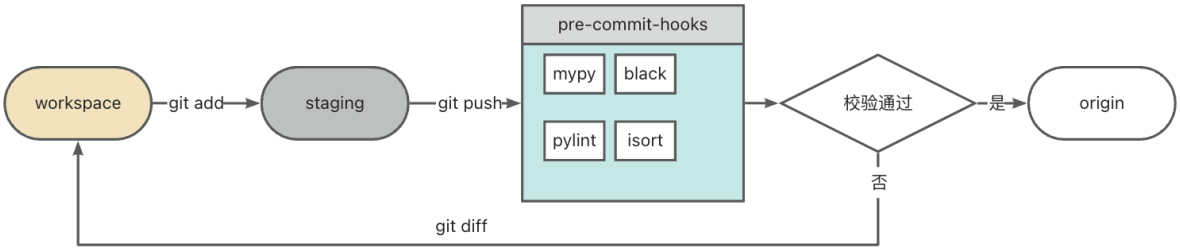
Gitlab协同规范



前言：遵循PEP8规范，针对Python本身动态编译型语言，没有静态类型检查、无统一规范化格式的特性，采用当下开源的流行库pre-commit + isort + pylint + mypy 黄金组合，

能减少人为85%+错误问题，意在培养良好的编程习惯，熟悉运用此工具库以后，更加注重功能本身的实现。

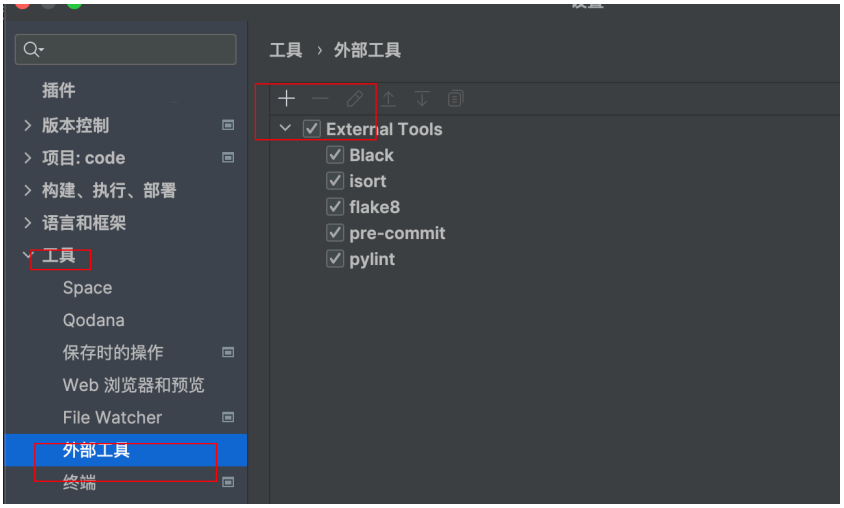
1. pre-commit: 代码提交钩子，针对本地commit 和 push 操作前的预制处理
2. isort: 整理from import | import 导包分组顺序格式类的问题，遵循PEP8
3. pylint: 代码检查：对不符合PEP8规范的代码进行说明提示，包括不限于：代码命名、长度、参数、变量数量、未引用的变量、缺乏文档（模块、类、函数）注释、算法优化等等...
4. mypy: 静态类型检查：针对py3开放注解代码风格以后，Python也对变量的类型支持了类型注解，但Python底层并未对注解的类型进行强限制，实际传入过程中仍然受用户控制，mypy帮助用户检查非类型传入，像其他主流静态编译检查类型开发语言考虑。

安装步骤：

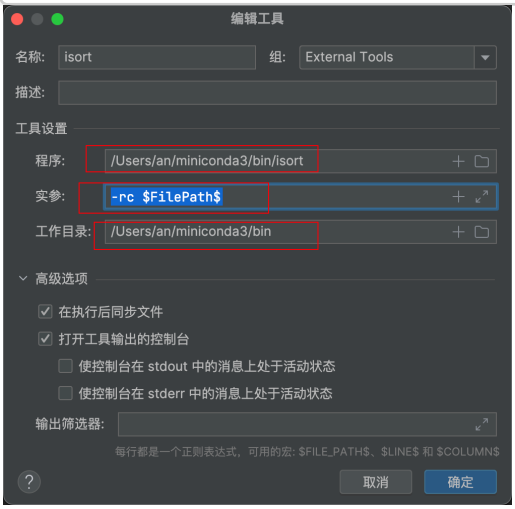
1、安装pip包

```
pip install pre-commit mypy black pylint isort
```

2、pycharm配置上述工具
【设置】->【工具】→【外部工具】【创建工具】



程序：pip install isort 后找到当前安装环境下bin目录下可执行二进制文件
实参：-rc $FilePath$　　（其他工具不需要-rc）
工作目录：程序可执行二进制文件的的上级目录



3、pre-commit钩子文件，将钩子文件放到工程应用主目录下

4、将hooks安装到本地的.git中

```
pre-commit install
```

5、执行commit 验证

概述

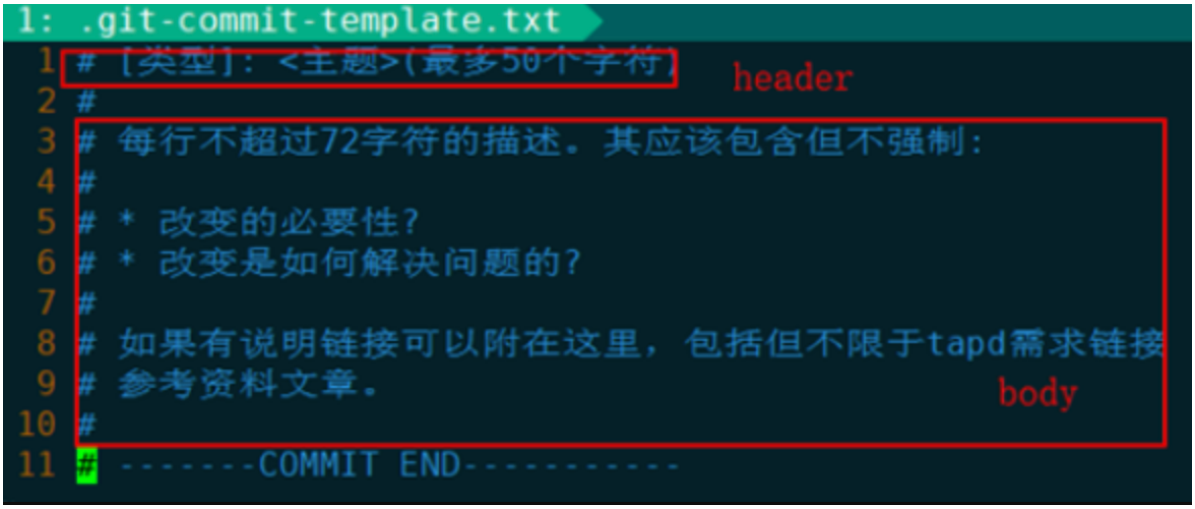
Git commit规范目的
1. 方便后来人员查看历史板本
2. 方便自动化生成Change Log
3. 提供更多历史信息, 方便快速浏览
4. 可以过滤某些commit便于查找信息 git log –grep feat

规范说明

1. Git commit message(以下简称msg)用中文书写

2. 不要使用commit -m 写commit log

3. msg分为两部分, header ;body 如下图



4. header 和body 用空行分开

5. header要求: 最多包含30个字符, 由两部分组成英文尖括号包围的类型, 以及commit主题, 类型是由固定的集中类型标签用于说明commit的类别,如下表1-1 主题简要描述本次提交的核心内容

```
feat        新特性开发（feature）
fix         修复bug（bugfix）
docs        文档增加或修改
style       格式修改（不影响代码的运行）
refactor    重构（既不是新增功能，也不是修复bug）
test        增加测试用例
chore       构建过程或构建工具改变

body要求: 每行不超过72字符， 包括但不限制， 改变的必要性， 改变是如何解决的， 参考资料链接等.
```

提交示例

- 建议将提交示例写入一个模板文件, 然后用如下命令应用模板: git config –global commit.template <.git-commit-template.txt file path>

```
# [类型]: <主题>(最多50个字符)
#
# 每行不超过72字符的描述。其应该包含但不强制:
#
# * 改变的必要性?
# * 改变是如何解决问题的?
#
# 如果有说明链接可以附在这里，包括但不限于tapd需求链接
# 参考资料文章。
#
# -------COMMIT END-----------
```

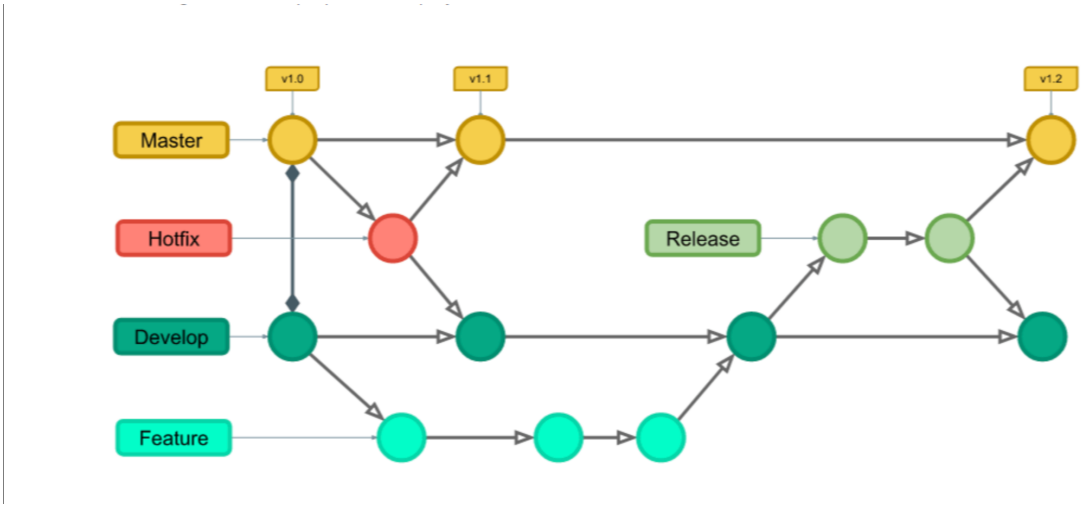
Vim增强gitcommit强制换行, 保证单行不超过72个字符在.vimrc中增加

```
autocmd Filetype gitcommit setlocal spell textwidth=72
```

Commit示例

关于中间小提交, 可以用refactor标签标识, 某个特性开发完成后用feat标签标识

Git flow流程



参考文献

https://robots.thoughtbot.com/5-useful-tips-for-a-better-commit-message
https://github.com/commitizen/cz-cli
https://github.com/thoughtbot/dotfiles/blob/master/gitmessage
http://www.ruanyifeng.com/blog/2016/01/commit_message_change_log.html