**CONVENTIONAL COMMIT**

**Warm-up: Hãy đọc tài liệu/blog dưới đây để hiểu hơn các khái niệm trong bài**

* [**Semantic Versioning**](https://viblo.asia/p/semantic-versioning-OeVKBN2EKkW)

**1. Giới thiệu**

* Conventional Commit là một bộ quy tắc viết commit message để cả cho người và máy đều có thể đọc được.

| **General Commit** | **Conventional Commit** |
| --- | --- |
| Lịch sử commit không được đẹp mắt khi không thống nhất chung một kiểu commit | Lịch sử commit chỉ theo một bộ quy tắc, trực quan, dễ đọc |
| Khó phán đoán được commit đó được sử dụng cho mục đích gì | Hiểu được tác dụng, mô tả cơ bản của commit |

**2. Cấu trúc**

* Cấu trúc chung của một commit message:

<type>[optional scope]: <description>

[optional body]

[optional footer(s)]

**3. Thành phần**

* Một commit bao gồm các thành phần sau:

1. **type:** từ khóa để phân loại commit
2. **scope (optional):** cũng dùng để phân loại commit, thể hiện đối tượng được thêm vào hay chỉnh sửa, có thể có hoặc không
3. **description:** mô tả ngắn về những gì được chỉnh sửa trong commit đó
4. **body (optional):** mô tả chi tiết về commit, có thể có hoặc không nhưng rất hữu dụng cho những commit lớn
5. **footer (optional):** được sử dụng để xác định BREAKING CHANGE (liên quan đến **MAJOR** trong Semantic Versioning), hoặc là số ID của pull request, issue,..., có thể có hoặc không

**4. Các kiểu commit**

* Giới thiệu về các kiểu commit trong conventional commit:

1. **build**: những thay đổi ảnh hưởng đến
2. **feat**: được sử dụng khi chúng ta thêm một tính năng mới (liên quan đến **MINOR** trong Semantic Versioning)
3. **fix**: được sử dụng khi chúng ta sửa được lỗi trong chương trình (liên quan đến khái niệm **PATCH** trong Semantic Versioning)
4. **docs**: được sử dụng khi mà tài liệu trong code bị thay đổi
5. **refactor**: sửa code nhưng không phải fix bug hay thêm một feature
6. **chore**: những sửa đổi nhỏ không liên quan đến code
7. **perf**: những thay đổi về code giúp tăng hiệu năng xử lý
8. **vendor**: cập nhật version cho các packages, dependency

Example:

* Commit message sử dụng **feat** khi có một chức năng mới trong chương trình

feat(lang): add Polish language

* Commit message với dấu ! thể hiện sự thay đổi bất ngờ

feat!: send an email to the customer when a product is shipped

* Commit message khi chúng ta sửa bug trong codebase:

fix(button): change the padding and margin which cover the button nearby