# Tips for Conducting a Code Review

Now that we know what we are looking for, let's go over some tips on how to actually write your code review. When your coworker finishes up some code that they want to merge to the team's code base, they might send it to you for review. You provide feedback and suggestions, and then they may make changes and send it back to you. When you are happy with the code, you approve and it gets merged to the team's code base.

As you may have noticed, with code reviews you are now dealing with people, not just computers. So it's important to be thoughtful of their ideas and efforts. You are in a team and there will be differences in preferences. The goal of code review isn't to make all code follow your personal preferences, but a standard of quality for the whole team.

> ### Tip: Use a code linter

This isn't really a tip for code review, but can save you lots of time from code review! Using a Python code linter like pylint can automatically check for coding standards and PEP 8 guidelines for you! It's also a good idea to agree on a style guide as a team to handle disagreements on code style, whether that's an existing style guide or one you create together incrementally as a team.

> ### Tip: Explain issues and make suggestions

Rather than commanding people to change their code a specific way because it's better, it will go a long way to explain to them the consequences of the current code and *suggest* changes to improve it. They will be much more receptive to your feedback if they understand your thought process and are accepting recommendations, rather than following commands. They also may have done it a certain way intentionally, and framing it as a suggestion promotes a constructive discussion, rather than opposition.

```
BAD: Make model evaluation code its own module - too repetitive.

BETTER: Make the model evaluation code its own module. This will simplify models.py to be less repetitive

GOOD: How about we consider making the model evaluation code its own module? This would simplify models.p
```

> ### Tip: Keep your comments objective

Try to avoid using the words "I" and "you" in your comments. You want to avoid comments that sound personal to bring the attention of the review to the code and not to themselves.

```
BAD: I wouldn't groupby genre twice like you did here... Just compute it once and use that for your aggre

BAD: You create this groupby dataframe twice here. Just compute it once, save it as groupby_genre and the

GOOD: Can we group by genre at the beginning of the function and then save that as a groupby object? We c
```

> ### Tip: Provide code examples