

Code Review and Best Practices

What is code review?

- Talking about or looking at code line by line
- Providing feedback

Why do code review?

- Catch bugs
- Improve code
- Get help thinking through design
- See different coding styles/approaches
- Learn about best practices
- Learn about new packages and methods
- Learn who to talk to about problems

What to think about

- Does the code perform as expected?
- Are coding guidelines followed (e.g. PEP-8)?
- Is the code easy to read and understand (variable names, function names, order)?
- Is the code properly documented (e.g. doc strings, references to origin of algorithms)?
- Is the code written efficiently/flexibly (e.g. hard-coded values, uses of functions/classes)?
- Is the code tested?
- Is there a license? (<https://choosealicense.com>)

Effective code review

- Choose a small piece of code (< 200 lines)
- Don't get hung up on stylistic choices
- Avoid stating opinion as fact
- Don't shame raw code (or coding language)
- Be careful of assumed knowledge (e.g. jargon or tools)
- Works best when you are familiar with the problem (e.g. within research groups)

Let's get started

- Make small groups
- Have one member present either their 3-State CA or some other code
- Talk through your thought process and any challenges you had in the codes development
- Remaining members feedback and