

COMMENTS

- do not write comments to make up for bad coding
 - ↳ it's better to explain yourself through code

* GOOD PRACTICE

- ⇒ prevent the reader from having to read the whole code to understand
 - legal comments: copyright, authorship...
 - informative comments: facilitate the reading of the code (regular expression...)
 - ↳ use good function names to avoid explaining the intent of the function in a comment
 - explanation of intent: explain a choice or a result
 - clarification: translate the meaning (that can not be altered) into something that is more readable.
 - warning of consequences: warn the user about something that they can not unless they run the code.
 - TODO comments: warn about work that should be done.
 - can add what the code should be in the future
 - Amplification: highlight the importance of something that could seem insignificant.

* BAD PRACTICE

- redundant comments: comments can be use to avoid reading a long part of the code but shouldn't be use short part
- misleading comments: write clear comments
- mandated comments: obligating the write of comments for a parser can lead to redundant comments.
- position markers
- closing brace comments
- attribution and bylines
- commented-out code: people will wonder if it's there for a reason
 - ↳ add explanations if inevitable
- html comments
- nonlocal information: write the comment near the part of the code it concerns

A comment should have a new information, be concise and have an obvious link with the code