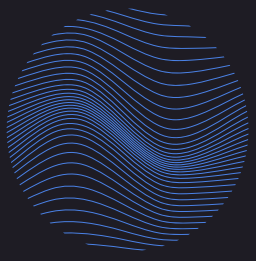


BEST CODING PRACTICES

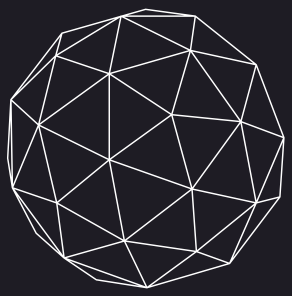
A brief guide to improve your coding skills in any language

APRIL 2023

Created by

Facundo Martínez

https://linkedin.com/in/fxmartinez

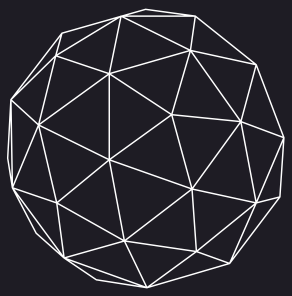


Using descriptive and meaningful variable names

When you define a new variable, use a name that precisely describes its nature, it does not matter if it's too long. Using descriptive variable names will allow you to quickly remember its purpose next time you review the code. Otherwise, you might forget its purpose.

Facundo Martínez

https://linkedin.com/in/fxmartinez

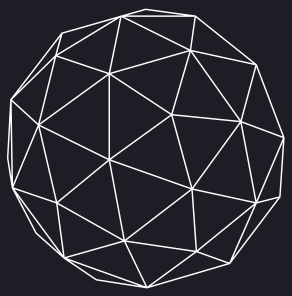


Writing modular code

If your codebase gets excessively large, you could try breaking it into smaller components. It will significantly improve readability and code maintenance if you ever find an issue in the near future. It will also allow you to test it in a more efficient way.

Facundo Martínez

https://linkedin.com/in/fxmartinez

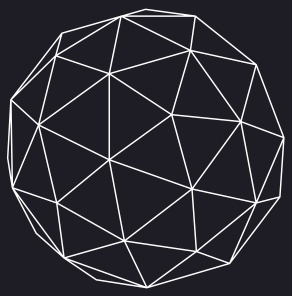


Writing documentation and commenting your code

Whether you're working alone or in a group project, proper documentation will allow all parties involved to get a better understanding of how the code works. Adding comments also allows for a quick reference during code writing.

Facundo Martínez

https://linkedin.com/in/fxmartinez

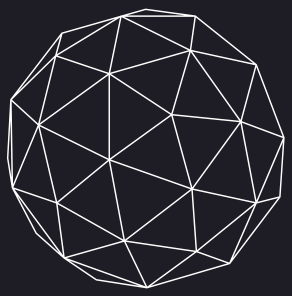


Using version control systems to keep track of the progress

Sometimes you will need to revert changes to your current project. Version control systems, like git, allow for a quick roll back. Sharing code in zip files is not a good coding practice, and there are several inconveniences related to it.

Facundo Martínez

https://linkedin.com/in/fxmartinez

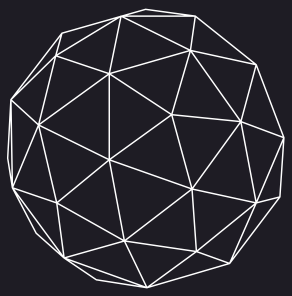


Writing readable code

Use indentation, spacing and proper formatting to make your code readable and easy to understand. Check the current best coding practices for the specific language you're working with, and encourage your team to comply with them. It will pay off in the long run.

Facundo Martínez

https://linkedin.com/in/fxmartinez

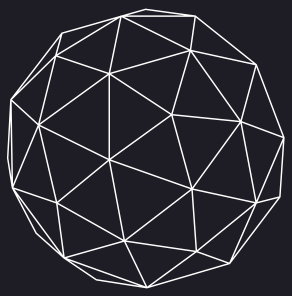


Writing unit test cases for your code

Writing unit tests will help ensuring your code is working as you want it to work, and help you catch potential bugs and refactor the code. Remember that unit tests must not impact performance significately.

Facundo Martínez

https://linkedin.com/in/fxmartinez

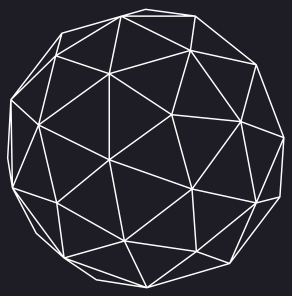


Handling errors in an informative way

Catch exceptions in a way that lets the user (and the developer) understand what went wrong during execution. You can do this by designing informative error messages.

Facundo Martínez

https://linkedin.com/in/fxmartinez

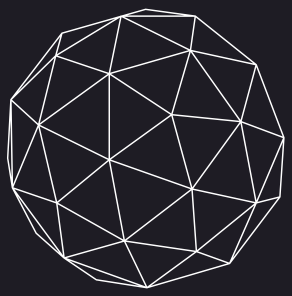


Use constants and variables instead of hard-coded values

Defining a While loop as "while (1)" is a bad coding practice. Use constants and variables instead of hard-coded values in the code. It will significantly improve readability and flexibility in case you need to refactor the code in the future.

Facundo Martínez

https://linkedin.com/in/fxmartinez

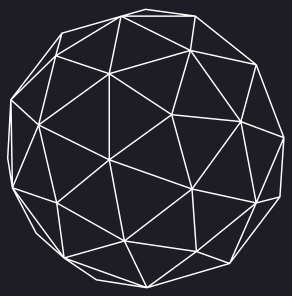


Following naming conventions for your specific language

Every programming language has specific naming conventions. For instance, Python's library NumPy is imported as "np". Naming conventions will help other developers understand your code and provide useful feedback faster.

Facundo Martínez

https://linkedin.com/in/fxmartinez



Refactoring the code periodically

Periodic code refactoring improves its quality and maintainability. You will get a better and more efficient code over time by constant refactoring.

Facundo Martínez

https://linkedin.com/in/fxmartinez

Thank you for reading!

Facundo Martínez

https://linkedin.com/in/fxmartinez