

Coding guide

When we examine your project, we check for functionality and style. Passing the functional requirements is not enough for the project to pass. It is also required that the code is easy to understand and maintain.

Comments

We expect your code to be commented in a reasonable way. It should be possible to understand what a particular segment of code does by reading the comments. Since many of our teaching assistants (and fellow students) come from abroad, it is advised to code and comment in English (as it is done in practice).

Function and variable names

Use descriptive names. By reading the variable or function name, it should be possible to understand what purpose it has.

Modularization

Build your code using functions.

Don't repeat yourself

Don't copy code segments repeatedly and change small parts. Try to put the logic in a function, or create a loop.

Understand when to use *for* and when to use *while*

for-loops are used when you know a-priori the number of iterations that will be performed. *while*-loops are used if the number of iterations depends on computations performed inside the loop.

Efficiency

We do not expect you to write highly optimized code. It is however problematic if your code is so slow that it takes us several minutes to run. To find out why your code is slow, use the profiler (MENU/DESKTOP/PROFILER). A typical reason is that calculations are done inside loops when they could have been placed outside, or increasing the dimension of a matrix or vector inside a loop.