Coding Best Practices
Summary

Variable Names
- should explain the variable
- better too long than too short
- Include units if the variable has one (ms, hz…)

Repeated Logic
Use **loops** for logic that is applied for multiple objects or iterations.
Use **functions** for logic that is or might be reused.

Modules
Move functions to modules with a shared topic (e.g. I/O, statistics, signal processing).
Goal: purely descriptive main scripts (what instead of how)
Bonus: reuse modules in future projects

Comments and Blocks
Use blocks to separate code into logical segments (paragraphs).
Describe each block with a comment.
Goal: understandable code without reading code

Documentation
Project level: README.md (*)
Module level: summary and authorship info at the top of scripts
Function level: summary, inputs, outputs, errors (*)

Collaboration (GitHub)
Use branches per work package and/or contributor.
Use pull requests to review branches before merging into main.
Use .gitignore to exclude sensitive, private or large data from Git and GitHub.

Markdown - README syntax

`#, ##, …, ######`
Headlines (main headlines to small headlines)

`[Link text](https://…)`
Links

`1. / 2. / 3.`
Ordered list

`* / + / -`
Unordered list

Function Documentation in R

```
#’ Documentation lines start with “#’”
#’
#’ Start with a Title
#’
#’ and a short summary
#’ focus on inputs, outputs, errors
#’
#’ @param parameter1 describe parameters here
#’ @param parameter2 focus on types (e.g. numeric) and units
(e.g. hz)
#’
#’ @return describe the function output (type, unit…)
demo_function <- function(parameter1, parameter2) { … }
```