{EPITECH}

MAKEFILE

GAIN TIME WHEN COMPILING YOUR CODE

MAKEFILE

When automatically evaluated, your source files are compiled using your own Makefile.
The following rules will be called (and thus are mandatory):

- ✓ **$(NAME)**: generates a binary called **NAME**.
- ✓ **clean**: removes all intermediate and temporary files.
- ✓ **fclean**: removes all non-source files.
- ✓ **re**: forces compilation.
- ✓ **all**: main rule

Obviously, all your source files should not re-compile each time you call *make*.
Only the minimum number of files should re-compile.

> If the Makefile re-compiles without proprely checking the dependencies, your project will be considered as not functional and will not be evaluated.

The **.PHONY**, even though not explicity compulsory, is to be considered, since it makes you Makefile functionnal.

> If your project calls a library (such as your *libmy* for instance), your Makefile should contain a rule to build it.
> This rule should be called by the **all** rule.

{ EPITECH }

Advice

Read the official documentation!

> Makefiles offer a lot of relevant possibilities.
> Study them and use them extensively!

Here are some good practices:

- ✓ defining variables make your Makefile clearer and easier to debug,
- ✓ using automatic variables can make your Makefile faster to modify and avoid mistakes,
- ✓ having pertinent rules to match each use case,
- ✓ adding human-readable outputs to have more pleasant displays.

On the contrary, using wildcards obfuscates the behavior of your Makefile and should be avoided.

> You should always modify your Makefile and its rules to match your project.
> Don't systematically re-use an old (and shoddy) one…

{EPITECH}

{EPITECH}