Coding-Style and How It Changed My Code

And Might Change Yours Too.

Mathieu Rivier
mathieu.rivier@epita.fr
Computer Science Student

May 2021

Abstract

Do you sometimes have trouble reading others code or grasp the full extent of what an algorithm is trying to do? Or maybe you find code found in books much easier to read than some other random code. Or do you have trouble reading and understanding a pieces of code you wrote a long time ago? While a well defined coding style will not do the work of understanding at your place, a properly defined one would help you alleviate some of those problems.

article_presentation.py

```python
def present_article(article_author):
    """ Prints the presentation of the Article"""

    article_title = "Coding-Style and how it changed my code"
    article_contact = article_author + "@epita.fr"
    article_date = "May 2021"

    print("A Paper on " + article_title
        + ", written by: " + article_author
        + " in " + article_date
        + ". Contact " + article_contact + ".")


present_article("mathieu.rivier")
```

my_shell

```
$python3 article_presentation.py
A Paper on Coding-Style and how it changed my code, written by: mathieu.rivier
↪  in May 2021. Contact mathieu.rivier@epita.fr.
```

Contents

1 Introduction **3**

2 Coding Style and Coding Conventions: What Is Their Purpose? **3**

2.1 Coding Conventions 3

2.2 Coding Style 4

3 Coding Convention – Coding Style: The Benefits – Through Examples **4**

3.1 Line Length 5

3.1.1 Line Length 101 - Proof of Concept 5

3.1.2 Line Length 202 - Implementation 5

3.2 Structure 6

3.2.1 Structure 101 7

3.2.2 Structure 202 8

3.3 Consistency 8

3.4 Documentation 9

3.5 Conclusion – On the Example + Answer 11

4 Concrete Tools To Help the Reader in His Journey – Making the Switch **11**

4.1 Code Format Tool - Code Formatters 11

4.1.1 Some Examples 11

4.2 Checking Coding Style in GitHub 11

5 Conclusion **12**

1 Introduction

Most developers who do not use a coding style find them to be cumbersome or even counterproductive. While this outlook might be enough for some of us, this article will go into further detail on why you should reconsider the use of a coding-style, for yourself and your organisation. This article will be composed of three sections; The initial aim will be to go over the definitions of coding style and coding convention. The second will be to put in exergue the numerous benefits you will be able to grasp. The final aim will be to see concrete tools to help one's journey in improving code quality.

2 Coding Style and Coding Conventions: What Is Their Purpose?

Do you sometimes have trouble reading others code or grasp the full extent of what an algorithm is trying to do? Or maybe you find code found in books much easier to read than some other random code. Or do you have trouble reading and understanding your pieces of code that you wrote a long time ago? While a well defined coding style will not do the work of understanding at your place, a properly defined one would help you alleviate some of those problems.

2.1 Coding Conventions

You might know some languages that force some coding conventions on you, for better or worse. A famous example of this is the Python language.

```
1 # This code would not work!
2 def myLongFuncName():
3     print("Hello")
4       print("How are you?")
5 # Line 1:
6 # Python Coding Convention Broken : Python uses Snake_Case
```

Program 1: Example of a Simple Non-Coding-Convention Compliant code

While those two programs are similar, anyone who has coded in Python before would know: only the second one would work. While using camelCase instead of Snake_case at line 2 in **Program 1** is not a problem for compilation, the python coding convention is still not respected. On the other hand, at line 4 of **Program 1**, the wrong indentation will prevent the code from running. There is more to the story (part of it has to do with the fact, Python is not using brackets to delimit scope). Although this last part is out of the scope of this subject, pun intended.

```
1 # This code would work!
2 def my_long_func_name():
3     """The function prints : Hello\nHow are you?"""
4     print("Hello")
5     print("How are you?")
```

Program 2: Example of a Simple Coding-Convention Compliant code

So are there other such rules? Yes! Python has, like many languages, a style convention that they call [PEP-8(2013)]. For example: in Python, if needed, you should always put a line break before a binary operation rather than after it.

You have taken Python as an example; is it the only language to have a coding convention? No, in fact, [JavaScriptCodingStyle(2021)] is another famous one.

So what happens if one does not follow those Coding Conventions? Well, as the name suggests, they are conventions, so officially nothing. But your code will become ever so harder to read by other developers (See Section 3. for more information), or even yourself in the future.

The reader might now be wondering about the usefulness of a coding style. Why create a coding style when style guides made by the maintainers of a language themselves exist.

2.2 Coding Style

Contrarily to a Coding Convention, written by the maintainers of a language, anyone can create a coding style. That can be both a great and terrible thing depending on whom is creating the coding style.

Generally, a coding style is similar to the coding conventions of a given language (official or not). This for a simple but important reason, the universality of the code!

A coding style might differ slightly from coding conventions to have some aspects or rules that may be more restrictive than the convention. As an example: Imagine a rule: limiting the length of a function to 20 lines. Such a rule would encourage one to divide their code more cleanly hence enabling a better code structure.

3 Coding Convention – Coding Style: The Benefits – Through Examples

This section is explained through a master example showing everything one should not do, proceeding point by point on everything that can improve efficacy, readability, comfort and comprehension.

`my_python_file.py`

```
1 query = "WITH subject AS (SELECT studentIDENTITY, subject_id, schoolNumber, sum(exam_score) AS total_score  FR
```

Program 3: *Version 1* - Very bad code style

But why does it even matter?

Well, **Program 3** is, to say the least unreadable. As Robert C.Martin wrote in his book: "Indeed, the ratio of time spent reading versus writing is well over 10 to 1. We are constantly reading old code as part of the effort to write new code. [...][Therefore,] making it easy to read makes it easier to write." [Martin(2008)]. As pointed out in the aforementioned, as a developer, one will spend most of their time reading code rather than writing it. As such, a developer needs to have a clean and easy to understand code.

The following code comes from [Pankowski(2017)] modified for the example. Believe me, if you will, but I encountered a very similar line as **Program 3** in production code, about two weeks ago. Did someone say there is a part of the query missing? Well, that is how it was in the programmer's editor. One might believe it uncommon to find such code. In reality, it is most plausible that either the reader or an acquaintance has already written such a line. It is pure cruelty towards the ensuing developer! If one were required to explain the purpose of the query, it simply would not be possible.

3.1 Line Length

3.1.1 Line Length 101 - Proof of Concept

Let's make it better; the following is the same line as **Program 3** but with line breaks, so one can examine the entirety of the code. It is the equivalent of adding `"editor.wrappingColumn": 0` to your `settings.json` in VS-Code. To help the reader further, the following code is also highlighted as SQL code, enabling a better understanding.

```
WITH subject AS (SELECT studentIDENTITY, subject_id, schoolNumber, sum(exam_score) AS
↪  total_score  FROM subject_marks GROUP BY studentIDENTITY, subject_id, schoolNumber),
↪  student AS ( SELECT studentIDENTITY, schoolNumber, avg(total_score) AS
↪  STUDENT_average  FROM subject GROUP BY studentIDENTITY, schoolNumber) SELECT
↪  schoolNumber, avg(STUDENT_average) FROM student GROUP BY schoolNumber;
```

Program 4: *Version 2.1* - Line Breaks Example

Unlike **Program 3** the new **Program 4** is decryptable, meaning any developer given 5-10 minutes max would be able to understand what the line does. Although being honest, it is still far from ideal. Moreover, the reader is benefiting from SQL-Highlighting which would not be possible in VS-Code. Furthermore, this code would be an ordeal to debug! Finally, mistakes can easily be made.

Although, this *hack* has drawbacks. Firstly it only applies to a single machine and does not permanently improve the code. Secondly, it remains overly complicated to read and does not follow coding conventions.

3.1.2 Line Length 202 - Implementation

OK. So it is now time to apply the Line Length convention. Most languages set a limit to 80 characters. Although some as famous as Java used to have as a convention 80 characters [JavaCodeConventionsARCHIVE(1999)], and have since opted to favour name clarity over line length. It is nowadays a recurrent convention to respect 120 character line length limit in Java. In this example: SQL conventions will be followed inside a python String.

my_python_file.py

```
query= "WITH subject AS ( \
 SELECT \
 studentIDENTITY, \
 subject_id, \
 schoolNumber, \
 sum(exam_score) AS total_score \
 FROM subject_marks \
 GROUP BY studentIDENTITY, subject_id, schoolNumber \
), \
 \
student AS ( \
 SELECT \
 studentIDENTITY, \
 schoolNumber, \
 avg(total_score) AS schoolNumber \
 FROM subject \
 GROUP BY studentIDENTITY, schoolNumber \
) \
 \
SELECT \
schoolNumber, \
 avg(schoolNumber) \
FROM student \
GROUP BY schoolNumber"
```

Program 5: *Version 2.2* - Line Length Convention - Using SQL convention inside Python String

Wow, this is a lot. Well firstly, the code in **Program 5** is much more readable than both **Program 3** and **Program 4**. Secondly, to make it clear: the \ character at the end of each line is the way to tell Python the string continues onto the next line. It respects both Python and SQL coding Conventions. Although, respecting a convention does not mean it is well-executed. Indeed, modification of this variable would be a nightmare. Merely imagine deleting a comma by inadvertence; it would most certainly take a while to solve the enigma. What is clear is that one wants a separate file for this query. That is code structure and the next section!

3.2 Structure

The initial `my_python_file.py` has come a long way. The code is now readable-ish, as well as understandable-ish. What would now be the most valuable is to restructure the organisation. I omitted to mention it, but the initial variable in the production code mentioned in the intro was global. It is not something a developer wants, especially as, in this situation, there was no good reason to use a global variable. The access to the database was not in a function either. The beforementioned causes problems of understanding and structure. And would make refactoring much harder.

3.2.1 Structure 101

The 101 section of a structure is deceptively easy. It consists of creating a new SQL file and moving the query to it. What are the main benefits of doing this? Firstly, the developer will now have access to SQL highlighting. Secondly, the one line aspect, for conciseness in the `my_python_file.py` file, is back. Thirdly, it now much easier to edit and comprehend the SQL query in **Program 6** than it was in previous versions.

`my_python_file.py`

```
1 avg_pts_stds_each_school = open("avg_pts_stds_each_school.sql").read()
```

`avg_pts_stds_each_school.sql`

```
WITH subject AS (
 SELECT
 studentIDENTITY,
 subject_id,
 schoolNumber,
 sum(exam_score) AS total_score
 FROM subject_marks
 GROUP BY studentIDENTITY, subject_id, schoolNumber
),

student AS (
 SELECT
 studentIDENTITY,
 schoolNumber,
 avg(total_score) AS schoolNumber
 FROM subject
 GROUP BY studentIDENTITY, schoolNumber
)

SELECT
schoolNumber,
 avg(schoolNumber)
FROM student
GROUP BY schoolNumber;
```

Program 6: *Version 3.1* - Code Structure - Segregating the SQL query into its own file

The reader will note that the renaming of the query and the name of the SQL file. Making; much clearer as to what it is supposed to do. Furthermore, the purpose of the function now expressed through the function name; the newly internal variable can thus also be renamed.

3.2.2 Structure 202

Enough SQL queries for now. One might aspire to concentrate on the Python part and polish it up. Admittedly, the variable is still global, and nothing defines or creates an environment in which it is easy to use the query anywhere in the project.

my_python_file.py

```
1 def get_avg_pts_stds_schools(connection):
2     get_avgs = open("avg_pts_stds_each_school.sql").read()
3     # We could Imagine : - The following is just for demonstration
4     return execute_read_query(connection, get_avgs)
```

Program 7: *Version 3.2* - Code Structure - Defining Function in python

The structure component will settle here.

It is now possible for developers to write on the same file without fear of breaking code each time a modification is needed. It does not eliminate mistakes but renders them less likely. Moreover, the newfound clarity would permit a new developer to be operational much quicker.

Nevertheless, appreciate **Program 7** as a simple illustration. It is not the way to do things. The function could be within a class, including a class connection variable set upon instantiation. As a note, the connection variable passed as an argument of the function represents the connection to the SQL database. The latter created in another function hidden here for brevity. If you want to learn more regarding Python and SQL please see [Malik(2019)].

3.3 Consistency

As Human we necessitate consistency. That is why the reader most likely takes his coffee at the same time every day and eats a similar breakfast most days. And if you can believe it, this behaviour also extends to code.

Most advanced developers are dependent on high code quality to be able to perform best. The following study considered the difference in speed and correctness of code corrections on disparate types of programs. The forthcoming represents the comparison between two program types, Alphas (with code consistency), Betas (without), "The programmers made almost three times as many changes on the beta programs [without code consistency] as they did on the alpha programs" [SOLOWAY and EHRLICH(1984)], the authors, depict the importance of a well structured, coding-style driven program. The authors concluded that while this dependence on well-structured code is beneficial in code understanding, maintainability and conciseness, they mention how dependent expert developers are towards high code quality.

avg_pts_stds_each_school.sql

```sql
WITH subject AS (
 SELECT
 student_id,
 subject_id,
 school_id,
 sum(exam_score) AS total_score
 FROM subject_marks
 GROUP BY student_id, subject_id, school_id
),

student AS (
 SELECT
 student_id,
 school_id,
 avg(total_score) AS average_student
 FROM subject
 GROUP BY student_id, school_id
)

SELECT
school_id,
 avg(average_student)
FROM student
GROUP BY school_id;
```

Program 8: *Version 4* - Code Consistency - renaming variables

Hence why code consistency was the penultimate subject of this thinking process; If one needed to retain something, it is that: consistency is the key. What changed in the program is the names of the variables. In a real-life environment, one might not always have writing access to databases. But for this example, there was access to the database.

In this way, the core component of this journey finishes. There remains one more section to talk about the documentation.

3.4 Documentation

Documentation is a funny one. Indeed, some people say it is mandatory; others say it is a dangerous practice. But why is that? It is true that, more often than not, functions get modified while documentation might remain left behind. One can picture the predicament of having a faulty description of what a function does. It is misleading and defeats the purpose of documentation. So what do these people use? TESTS. Yes, tests, assuming the reader follows a TDD process or follows AGILE methodology; tests are a core part of his development process. As such, they will always be up to date and accurately describe how and what a function does. It is, supposing that one writes tests in the first place. Yes, shockingly enough, many people still write code without tests.

Consequently, should one ban documentation from their code?

No, documentation is essential. Although, to keep its purpose, make sure to maintain it adequately.

`my_python_file.py`

```python
def get_avg_pts_stds_schools(connection):
    """ Return the average amount of points for every
    student from every school
    """
    get_avgs = open("avg_pts_stds_each_school.sql").read()
    # We could Imagine : - The following is just for demonstration
    return execute_read_query(connection, get_avgs)
```

Program 9: *Version 5*- Code Documentation - Python documentation

By now, the reader most likely understands what the function does. NICE. The documentation is the last component of clarity. Although Python documentation is a bit special, so for this one, allow me to consider a complementary example from another language more representative of what one might call normality.

`util_is_whitespace.h`

```c
/**
** \brief checks if the char given as argument is a white space
**
** \param c is the char to check.
**
** \return returns 0 if is white space else 1.
*/
int is_whitespace(char c);
```

`util_is_whitespace.c`

```c
#include "util_is_whitespace.h"

int is_whitespace(char c)
{
    return c == ' ' || c == '\t' || c == '\r' || c == '\n' || c == '\v'
        || c == '\f';
}
```

Program 10: Documentation in C

What is interesting in **Program 10** is the documentation, do not pay too much attention to the fact there are two files. The header file (.h) is just a C perk. This documentation is what you would find in C++ and very similar to what you will find in Java. Hence why it is such a great example. The `\brief` indicates what the function does, `\param` represents a parameter of the function, and describes it. `\return` is what the function returns pretty self explanatory. As a side bonus, this documentation is compatible with Doxygen, which is a standard for C, C++, Java and more [van Heesch(1997-present)]. Other documentation formats exist for C, C++ and JAVA, although Doxygen remains one of the most famous.

3.5 Conclusion – On the Example + Answer

Here for the long-awaited answer! It is the extract from [Pankowski(2017)] explaining what the query does: "We want the result of this query to show us the average amount of points for every student from every school. The first stage (the WITH subject clause) finds the sum of points for every subject. The second stage (the WITH student clause) averages points for every student. By the time we reach the SELECT, all we need to do is find the average amount of points and group the results by school."

Did you get it right? I am sure you did!

Now, what about the leanings of this section. I hope to have managed to convince the reader about the importance of following a coding style and improving their code quality. Through the use of Structure, Line Length, Consistency and Documentation, we were able to transform a scary program into one that is: simple to understand, easy to read and easier to debug.

4 Concrete Tools To Help the Reader in His Journey – Making the Switch

Now that the reader is knowledgeable on the importance of producing a high-quality code. And further comprehends the founding principles of one, here are some tips to help start their journey in writing coding style compliant code.

4.1 Code Format Tool - Code Formatters

Code Format Tools are brilliant, usually called code formatters. They do the job of formatting the code for you, respecting the 80 line char limit, deleting trailing white space, indent properly, and much more. They are the developer best friend to produce quality code.

4.1.1 Some Examples

Honorable mention: If you write C, C++ or a C family language, then the Clang-Format [ClangTeam(2007-2021)] is the way to go. One can create their Formatting File to follow internal Coding Styles. Give it a search.

In general, Code Formatting tools can be found on the go. Type in *[language] formatter* in your editor of choice. Chances are, you will find a decent option.

4.2 Checking Coding Style in GitHub

To make sure every member of a team follows the coding style, GitHub actions allow one to run a coding style checker. Thus preventing the merge if the coding style is not respected. An example (I have not tried) can be found here: python-style-check.

5 Conclusion

Pfew, it has been a long yet interesting journey! The reader now knows what a coding style is as well as coding conventions. Throughout the journey, multiple pillars of a good code quality have been explained and demonstrated. Finally, some tools and tips were given to help the reader through their quest for greater code quality.

It is now apparent that most advanced programmers use some form of coding style, that is: deliberately or not. It is important to note that: a standardised coding style; is more beneficial and more important than having it your way. It is forthwith interesting to acknowledge the limits of the coding style. While a requisite for a clean and efficient codebase, it is not a one-stop solution. Rather, it subscribes and takes part in a broader solution such as the abovementioned agile methodology.

Lastly, Don't be a robot.



Figure 1: Robot vs Human

References

[PEP-8(2013)] PEP-8. Pep 8 – style guide for python code. *www.python.org*, 2013. URL `https://www.python.org/dev/peps/pep-0008/`.

[JavaScriptCodingStyle(2021)] JavaScriptCodingStyle. The javascript language - code quality. *javascript.info*, 2021. URL `https://javascript.info/coding-style`.

[Martin(2008)] Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Pearson, 2008.

[Pankowski(2017)] Marek Pankowski. How to organize sql queries when they get long. *learnsql.com*, 2017. URL `https://learnsql.com/blog/5-tips-managing-long-sql-queries/`.

[JavaCodeConventionsARCHIVE(1999)] JavaCodeConventionsARCHIVE. Javase codeconventions-indentation - archive. *www.oracle.com*, 1999. URL `https://www.oracle.com/java/technologies/javase/codeconventions-indentation.html`.

[Malik(2019)] Usman Malik. Introduction to python sql libraries. *realpython.com*, 2019. URL `https://realpython.com/python-sql-libraries/#mysql`.

[SOLOWAY and EHRLICH(1984)] Elliot SOLOWAY and Kate EHRLICH. Empirical studies of programming knowledge. *www.ics.uci.edu*, 1984. URL `https://www.ics.uci.edu/~redmiles/inf233-FQ07/oldpapers/SollowayEhrlich.pdf`.

[van Heesch(1997-present)] Dimitri van Heesch. doxygen - documenting the code. *www.doxygen.nl*, 1997-present. URL `https://www.doxygen.nl/manual/docblocks.html`.

[ClangTeam(2007-2021)] The ClangTeam. Clangformat. *clang.llvm.org*, 2007-2021. URL `https://clang.llvm.org/docs/ClangFormat.html`.