General Relativity Applied to Coding Styles

General Relativity is not “everything is relative”. On the contrary, it is all about what does not change under certain types of transformation.

Jean-Louis Leroy - jl@leroy.nyc

Reading vs Writing and Changing

- *Code readability matters above all*, they say.
- Wake up guys! (and the 5 gals!)
- Nobody has ever been paid to read code.
- We are paid to *write* code.
- Reading code is just a means to an end.
- Writeability matters!
- *Transformability* matters even more!
- Let’s see what Einstein has to say about it...

Main Equation of Transformation Friendly Styles

- Let S(*code*) be true if *code* is formatted according to the rules of style S.
- Let T(*code)* be a transformation of code.

$$S(\mathit{code}) \Rightarrow S(T(\mathit{code}))$$

“copy/move” transformations

- If *code* is correctly indented, moving it or copying it to a correctly indented location results in correctly indented code.
- “Great Wall” rules violate the Main Equation.
  - “No line of code shall extend beyond the 80th (or 79th, or 72nd) column”
- Replacement: a line of code should not be longer than (e.g.) 40 characters *excluding indentation.*

“lengthening” transformations

- Changing the length of an identifier, or adding arguments to a function call, should not cause style violations.

```
double solve(double a,
             double b,
             double c) {
  // ...
}
```

```
double solve_quadratic(double a,
             double b,
             double c) {
  // ...
}
```

fix indentation ☹

```
double solve(double a,
    double b,
    double c) {
  // ...
}
```

```
double solve_quadratic(double a,
    double b,
    double c) {
  // ...
}
```

move on, do useful work ☺