

# Code Style

YEGOR BUGAYENKO

Lecture #22 out of 24  
80 minutes

All visual and text materials presented in this slidedeck are either originally made by the author or taken from public Internet sources, such as web sites. Copyright belongs to their respected authors.



BRIAN KERNIGHAN

“The harder it is for people to grasp the intent of any given section, the longer it will be before the program becomes operational. Trying to outsmart a compiler defeats much of the purpose of using one. Write clearly — don’t sacrifice clarity for ‘efficiency.’”

— Brian W. Kernighan and Phillip James Plauger. *The Elements of Programming Style*. McGraw-Hill, Inc., 1974

## Which One Is Better?

```
1 int f(int n)
2 {
3     if (n == 1 || a < 2)
4         return 1;
5     int a = f (n-1);
6     int b = f (n-2);
7     return a + b;
8 }
```

```
1 int fibonacci(int n) {
2     if (n <= 2) {
3         return 1;
4     }
5     return fibonacci(n - 1)
6         + fibonacci(n - 2);
7 }
```

## My Favorite Style Checkers

- Qulice for Java (Checkstyle + PMD)
- ESLint for JavaScript
- Clang-Tidy for C++
- Pylint for Python
- Rubocop for Ruby
- PHP\_CodeSniffer for PHP
- rustfmt for Rust

# References

1974.

Brian W. Kernighan and Phillip James Plauger. *The Elements of Programming Style*. McGraw-Hill, Inc.,