



Code

コード・リーディング

Reading

オープンソースから学ぶソフトウェア開発技法

Diomidis Spinellis 著 / (株)トップスタジオ 訳
熊岡文雄 / 平林俊一 / 釜つもとゆきひろ 監訳



コードは小説より面白い!

コードを小説のように読んで、独創的な筋書きや思いもよらぬ規則性を楽しみましょう。一見非常に複雑なものをコンパクトでエレガントな構造で実現しているものを見つけることもあれば、小粒ながら貴重な宝石のようなコードを見つけることもあります。

優れたコードを読むことを習慣付けた人は、間違いなく、より優れたコードを書くようになります。

コードは小説より面白い!

コードを小説のように読んで、独創的な筋書きや思いもよらぬ規則性を楽しみましょう。一見非常に複雑なものをコンパクトでエレガントな構造で実現しているものを見つけることもあれば、小較ながら貴重な宝石のようなコードを見つけることもあります。

優れたコードを読むことを習慣付けた人は、間違いなく、より優れたコードを書くようになります。

(; つ Д С)ゴ シゴ シ

(° Д°)え ?

わかったつ・も・りになる!

CodeReading

わかった つ・も・り とは？

- 概略だけ取り出してわかったつもりになってドヤ顔してもらおう事を想定して作っています。
- 正確性よりもわかりやすさを重視して作っています。
- この内容を聞いても、本当の実力はつきません。このスライドを出発点として各自勉強してみてください。



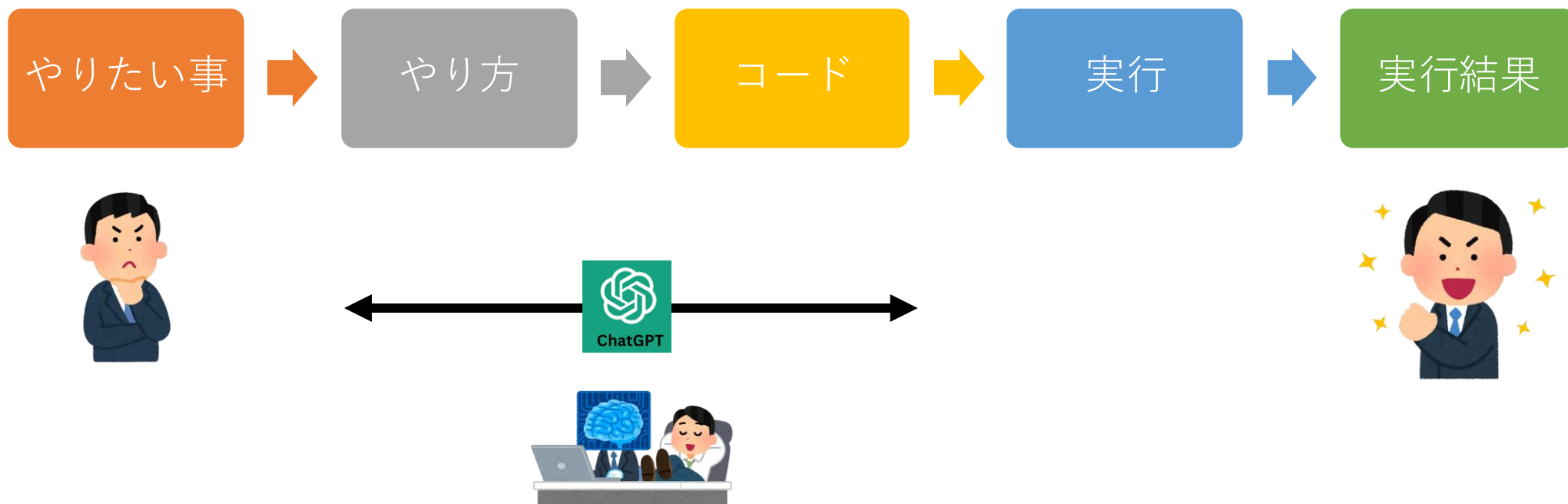
本日のお品書き

- **読む/書くではゲームのルールが違う**
- CodeReadingルール
- コードとはまだ見ぬ誰かへのお手紙



読む/書くではゲームのルールが違う

- コードを**書く時**は以下のようにします。



読む/書くではゲームのルールが違う

- コードを**読む時**は以下のようにします。



Q: 女の子が男の子を見てドン引きしています。男の子は何がしたかったか教えてください。

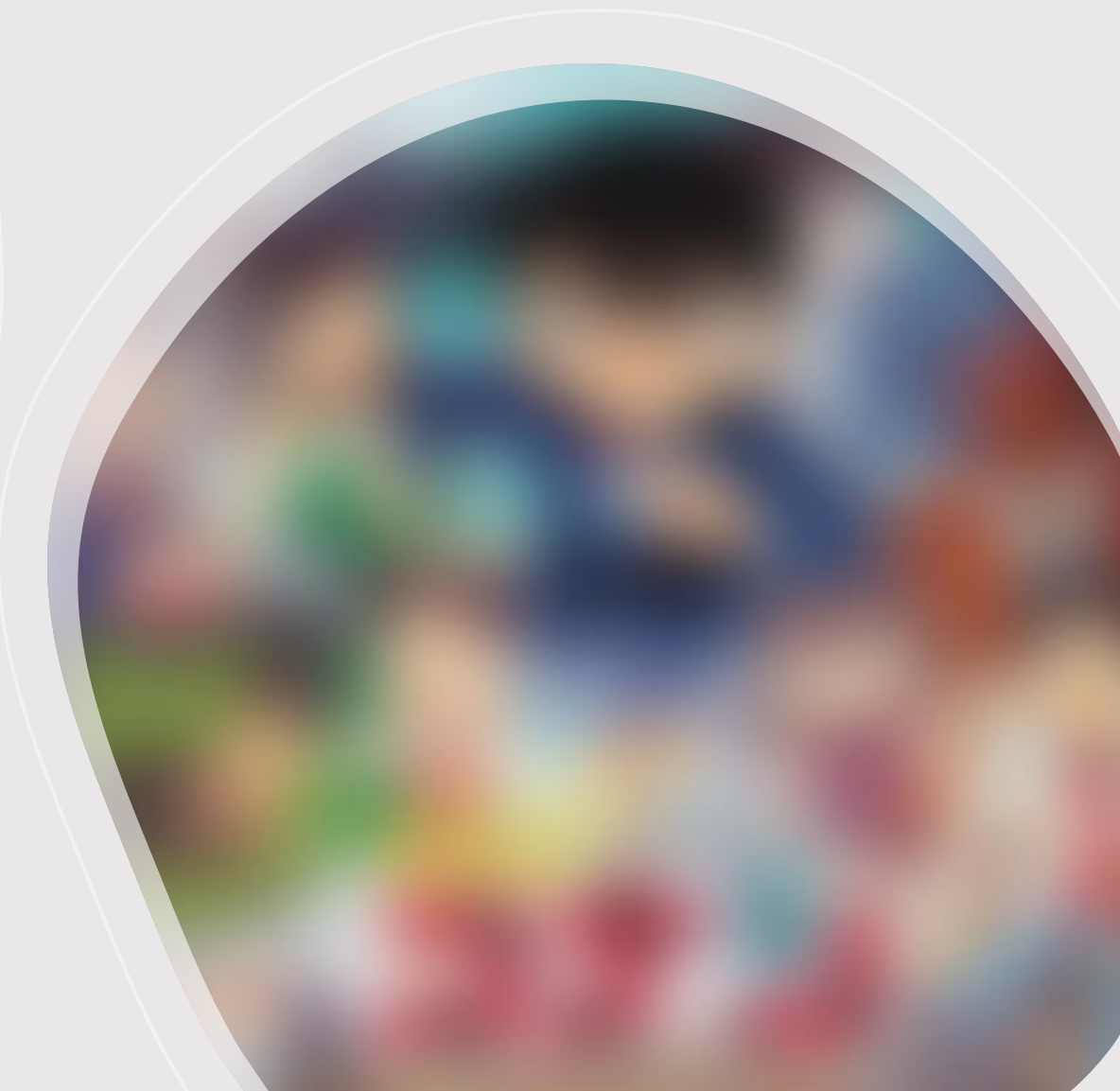
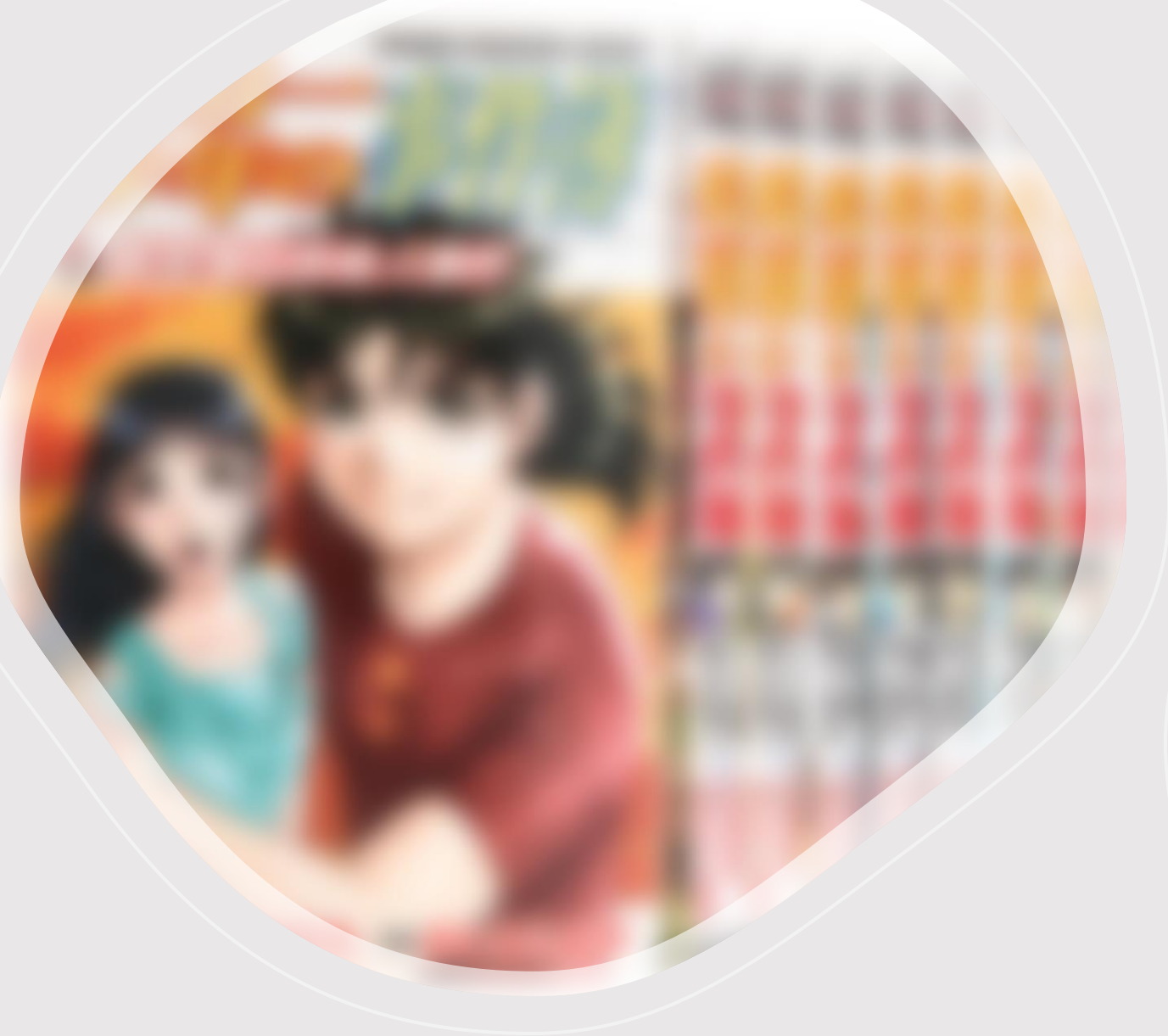
整理すると

コードを書く

- やりたい事からやり方を考えるゲーム

コードを読む

- やり方と結果からやりたい事を**推測する**ゲーム



やろうとしている事は、この人たちに近いのです。
(死人は出ないヨ)



最も重要なことは

細部に惑わされずに、
全体像と状況証拠から
コードが実現したい
本当の意図をつかみ取る

こと

本日のお品書き

- 読む/書くではゲームのルールが違う
- **CodeReadingルール**
- コードとはまだ見ぬ誰かへのお手紙



CodeReadingルール その1：はじめにコード を読んではいけな い

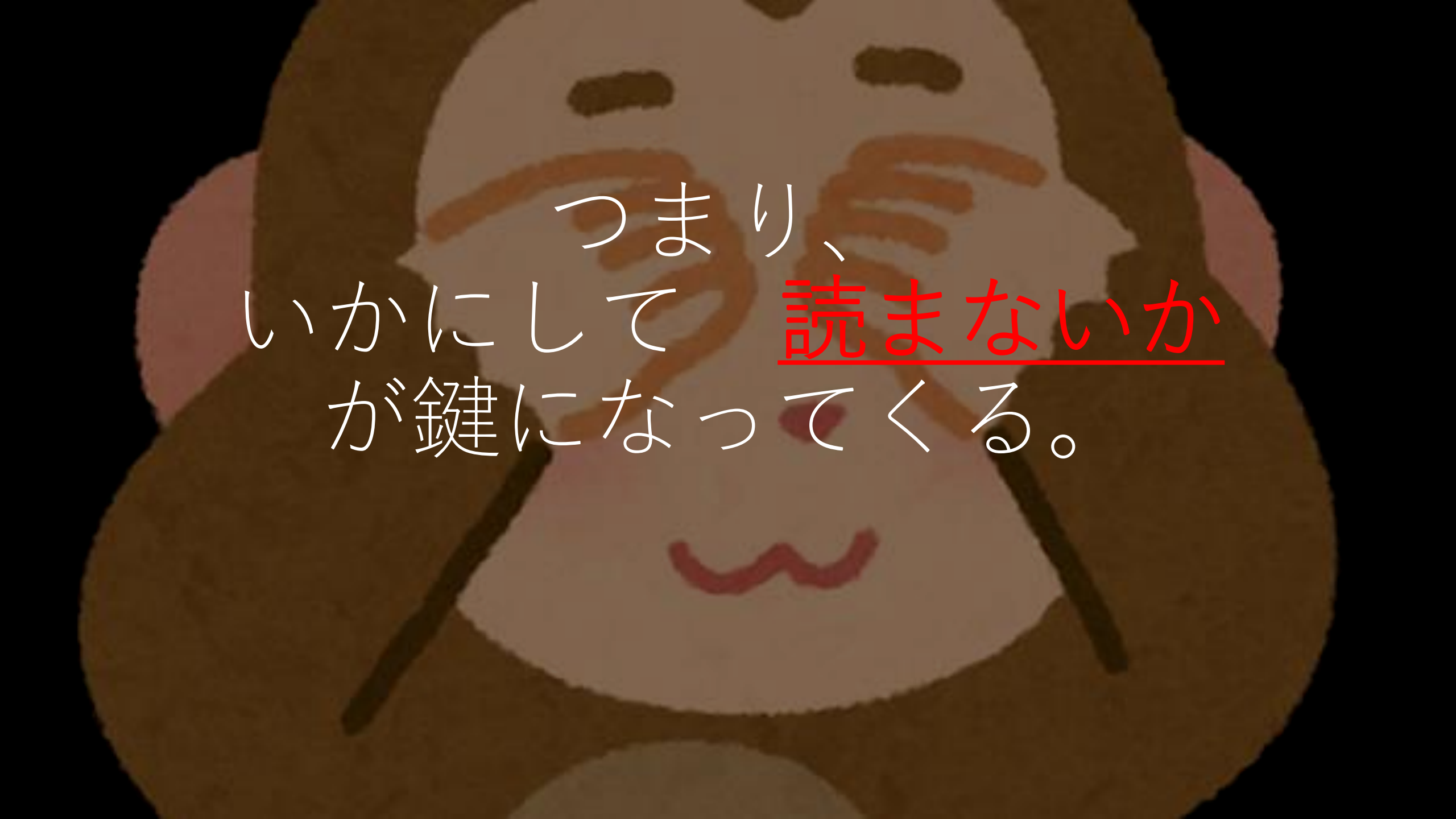
-
- 何も考えずにコードを読んではいけません。
 - コードを読むのはどうしても分からないときだけ。最後の最後です。



CodeReadingルール その2: はじめにコード を読んではいけな い

- コードを読むという事は時間がかかる事です。
- なぜなら、コードはやり方が書いてあるだけでやりたい事は書いてないからです。
- コードを読むという事は自分から意図的に泥沼に落ちるといことです 😞





つまり、
いかにして 読まないか
が鍵になってくる。

CodeReadingルール その3： システムの概要を調べよう!

コードを技術的に調べる前に、

- ・ 関係者からの聞き取り
- ・ ドキュメントの読み込み
- ・ 動かしてみても挙動を調べる

などを通して「**このシステムは何が目的で作られていて、何が出来るのか?**」という事を調べておきます。

技術者の場合、技術的に解決することを優先しがちですが、前提となる事実関係をしっかりと捉えることが最優先です。



CodeReadingルール その4：フォルダ構成は重要

はじめに読むべきファイルを絞り込みます。プロジェクトのフォルダ構成は非常に重要であり、フォルダを見て、読むべき場所をしばり混んでいきます。

各フレームワーク、言語ごとに「推奨すべきフォルダ構成」というものが定義されている事が多いです。ご自分のプロジェクトに関して、調べてみてください。

プロジェクトを始めるとき、オリジナリティーに富んだ「ぼくの考えた最強フォルダ構成」とかは後から読む人が困惑するので避けましょう。



CodeReadingルール その5：名前から妄想しよう

📌 お題は不同！ Qiita Engineer Festa 2023で記事投稿！

@rokumura7 (haxpig) in 株式会社うるる
投稿日 2023年07月18日

「良い名前付け」の参考サイトまとめ

📌 プログラミング, 英語, 命名規則, 名前, 命名

おはようございますこんにちわこんばんわ。どうもぶたです。

以前、チーム内で「変数や関数の名前に妥協したくないなー。どうしたら上手く命名できるんだろう？やっぱ英語の勉強？」という話になったので、今回は**名前付け**、**命名**についてまとめます。
とは言え、自分自身多くの記事やドキュメント、書籍などに助けられているので、ほぼ紹介記事になります。

ただ、順番には気がつけた方がいいと個人的には思っています。
何事もそうですが、**なぜやるのか**を知ってから**どうやるのか**を学ぶべきかな、と。

例えば、「この単語とこの単語はニュアンスが違う」「そんな単語存在しないよ」「単数と複数が間違ってる」
そんなレビューを受けたことがある人もいると思います。僕も言われたことがあります。
そういった内容の記事もたくさんあります。僕も読み込んでいますしストックして参照できるようにしています。

それはそれで有用ですし、是非意識していきたいことです。
ただ、そもそも、「**良い**」名前付けとは何なのか、**なんで「良い」名前付けをするのか**、という点を忘れてはいけないと思うのです。

読むべきフォルダが見つかったら、次に名前から内容を妄想します。

ファイル名、名前空間、クラス名、関数名、と絞り込みを行い対象を絞り込みます。

怪しいクラス、関数、処理をみつけたとしてもその場で読み進めてはいけません。

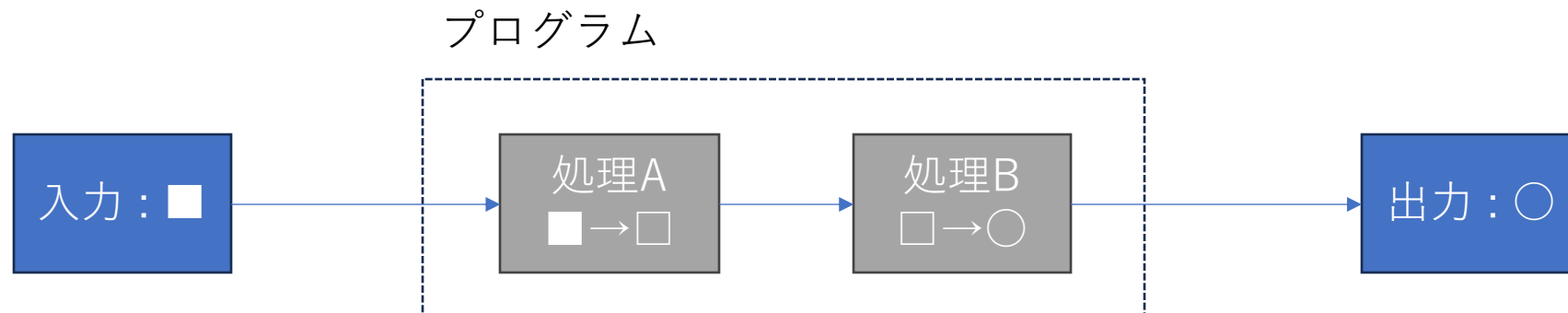
まずは、その場所を「**怪しいところリスト**」を作ります。
リストには簡単なメモをつけておくとよいでしょう。

重要なことは「**怪しいところの分布と関係**」をつかむことです。

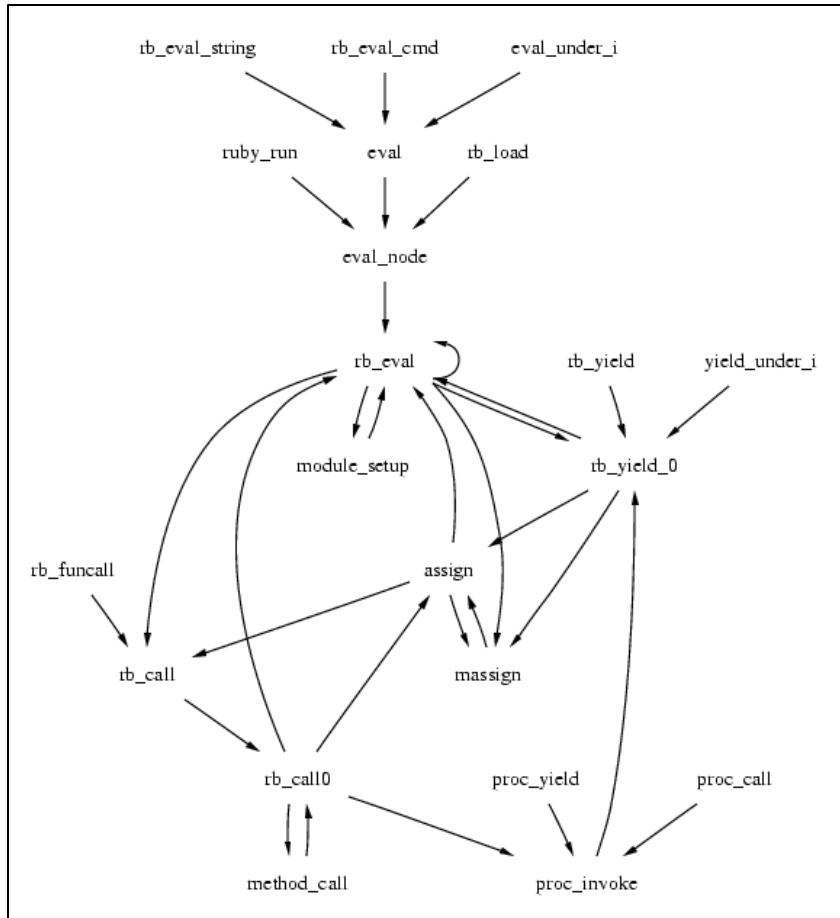


前提の復習：プログラムとは？

- プログラムとはデータを変換する手順のことです。
- 入力と出力があり途中に何段かの変換処理があります。
- 中身を調べる前に、各段の入り口と出口で何をやっているか知ることが全体像を把握する場合にとっても重要になります。



CodeReadingルール その6：関係を絵にする：コールグラフ



あやしいリストにある関数の呼び出し前、呼び出し後を図にします。

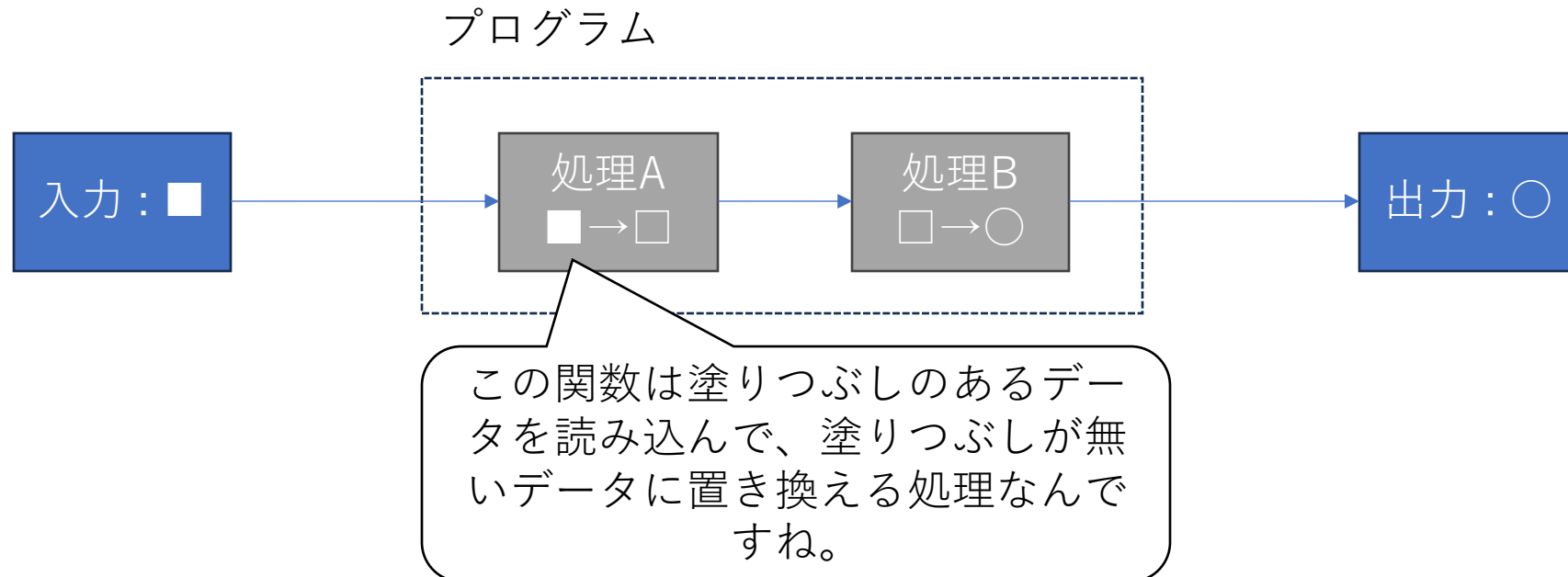
このような図の事を「コールグラフ」と言います。

コールグラフを使って処理の位置関係を調べます。

作成してくれるツールもあります。

CodeReadingルール その6：関係を絵にする：データフロー

- コールグラフの作成が出来たら、関数の呼び出しと戻り値としてどのようなデータが処理間で受け渡されているか調べます。
- これにより、これから読もうとしている処理の全体像が見えてきます。



CodeReadingルール その7:ブロックで読む

```
--
69 int pwd_main(int argc, char **argv) MAIN_EXTERNALLY_VISIBLE;
70 int pwd_main(int argc UNUSED_PARAM, char **argv UNUSED_PARAM)
71 {
72     char *buf;
73
74     if (ENABLE_DESKTOP) {
75         /* TODO: assume -L if $POSIXLY_CORRECT? (coreutils does that)
76          * Rationale:
77          * POSIX requires a default of -L, but most scripts expect -P
78          */
79         unsigned opt = getopt32(argv, "LP");
80         if ((opt & 1) && logical_getcwd())
81             return fflush_all();
82     }
83
84     buf = xrealloc_getcwd_or_warn(NULL);
85
86     if (buf) {
87         puts(buf);
88         free(buf);
89         return fflush_all();
90     }
91
92     return EXIT_FAILURE;
93 }
```

なんか、入力値のチェックかな?

処理の実態関数みたいだな。

画面にフラッシュしているな。

いよいよ、コードを読みます。
ここでも、漫然と最初からコードをよんではいけません。
コードをブロック単位に分けて何をやっているか推測して読むべき場所を絞り込みます。

そして、本当に自分が知りたい塊を見つけたら、そこを時間をかけて丁寧に読みます。

本日のお品書き

- 読む/書くではゲームのルールが違う
- CodeReadingルール
- **コードとはまだ見ぬ誰かへのお手紙**



コードとはまだ見ぬ 誰かへのお手紙

- コードを読むことということは推理でした。
- コードを書く人は、読む人が読みやすいコードを意識して書きましょう!
- それは、まだ見ぬ誰か(未来の自分を含む)へのお手紙なのです。
- だからそこ、**論理的に丁寧にわかりやすく、読む相手の事を考えて書きましょう!**

