

# 巨大なコードベースへ突撃するために

nsfisis (いまむら)

PHP カンファレンス福岡 2023 前夜祭 (非公式)

# 自己紹介

---

nsfisis (いまむら)



@ デジタルサーカス株式会社

はじめに

# 前置き

---

- 巨大なコードベースとは
  - 数十万から数百万行のソースコード
- 主な対象
  - 巨大なコードベースを読んだことがない方
- 話さないこと
  - 特定の言語向けの how-to (デバッガの使い方など)

# アジェンダ

---

1. ソースコードに突撃する前にできること
2. ソースコードを効率的に読む
3. 巨大なコードベースを読むときの心構え

**ソースコードに突撃する前に  
できること**

# 読む前にできること

---

読まなくても分かるなら、  
読まないに越したことはない

# 読む前にできること

---

- 公式ドキュメント
  - あれば
- イシュートラッカー
  - 誰かが報告しているかも
- リリースノート
  - 実はもう直っているかも



**ソースコードを効率的に読む**

# ソースコードを取得する

---

- ローカルに clone してすぐに読めるようにする
- `--depth=1` つきで shallow clone する (Git の場合)
  - 新しめの Git なら `git scalar` も (巨大リポジトリに最適化した Git の拡張)
- ブランチ .. タグを自分が読みたいものにする

# 構成を把握する

---

- ディレクトリ構成を把握する
- 全体のアーキテクチャを知る

典型的なパターンを知っていると有利

# 該当個所を見つける

---

コードリーディング 3 種の神器

# 該当個所を見つける

---

コードリーディング 3 種の神器

- git grep

# 該当個所を見つける

---

コードリーディング 3 種の神器

- `git grep`
- `git log`

# 該当箇所を見つける

---

コードリーディング 3 種の神器

- git grep
- git log
- vim

# 該当箇所を見つける

---

コードリーディング 3 種の神器

- git grep
- git log
- vim (neovim でも可)



# 調査・デバッグ

---

- テストコードを調べる
- 脳内でトレースする
- 言語・ライブラリ特有の落とし穴を重点的に読む

巨大なコードベースを読むときの心構え

すべての調査過程を記録せよ

開いたファイル、 読んだ関数、 立てた仮説、 etc