

なにこれ

ここ 1、2 週間ほどテスト関連の話をよく聞いていて、そういえば Front-end ってどんなテストやっているのか全然知らなかったから調べてみた。

モチベーション

- フロントエンドのテストについて知りたい
- 最近の動向を知りたい

テストの話

フロントエンドの単体テスト？

単体テストはアルゴリズムのロジックに対して行う時に有用。

→ 結合度の高いコードは結合テストをしっかりとやるのが効果的。

密結合なものは結合テストをする。

<https://postd.cc/a-response-to-why-most-unit-testing-is-waste/>

単体テストが書けるような設計・実装をする

- 例えばボタンを追加するとして、とりうる内部表現は無数にある
 - テスト対象の表現が一意に定まらない
- 他の開発者のために自分が書いたコードが要求を満たしているか検知する手段としての単体テスト
- GUI 開発は「ユーザがどう感じたか（UX）」に集約される
 - UX は単体テストの範疇外
- 単体テストが書ける = 不要な依存がなくモジュール同士の境界面が自明になってる
 - 単体テストが書けるような設計を目指すとい

<https://mizchi.hatenablog.com/entry/2018/03/13/214755>

UI 変更に耐えうるテストは難しい

フロントはロジックの変更よりも UI 変更の頻度の方が多い。

→ ボタンのデザインを変えたりするとテストのリファクタリングがはじまる

<http://w3q.jp/t/9438>

ビジュアル要素のテスト

- スナップショットテスト (html / 仮想 DOM)
 - html の比較: 仮想 DOM→HTML→diff をとる
 - 仮想 DOM の比較: 仮想 DOM→diff をとる

内部実装を変更するとテストに通らなくなる & 見た目（ブラウザへの描画結果）のデグレがわからん

<https://meetup-jp.toast.com/1550>

まとめ

- ロジックをできるかぎり切り出す
 - 切り出したロジックに対しては単体テストをする
- デザイン崩れを検知するにはビジュアルリグレッションテスト
 - デザインを変更するとテストの修正も必要になるから程々に

悩んだら読むと良さそうなサイト

- <https://zenn.dev/seya/scraps/6f930e359d6a7c>
 - フロントエンドテストの情報がまとまっている
- <https://speakerdeck.com/twada/>
 - テスト界隈で有名な人らしい。TDD の翻訳者
- <https://postd.cc/frontend-testing-is-for-everyone/>
 - 色々な種類のテストと、テスト範囲、メジャーなツールが紹介されている

ほいでは ٩(◁)و