Refactoring CSS

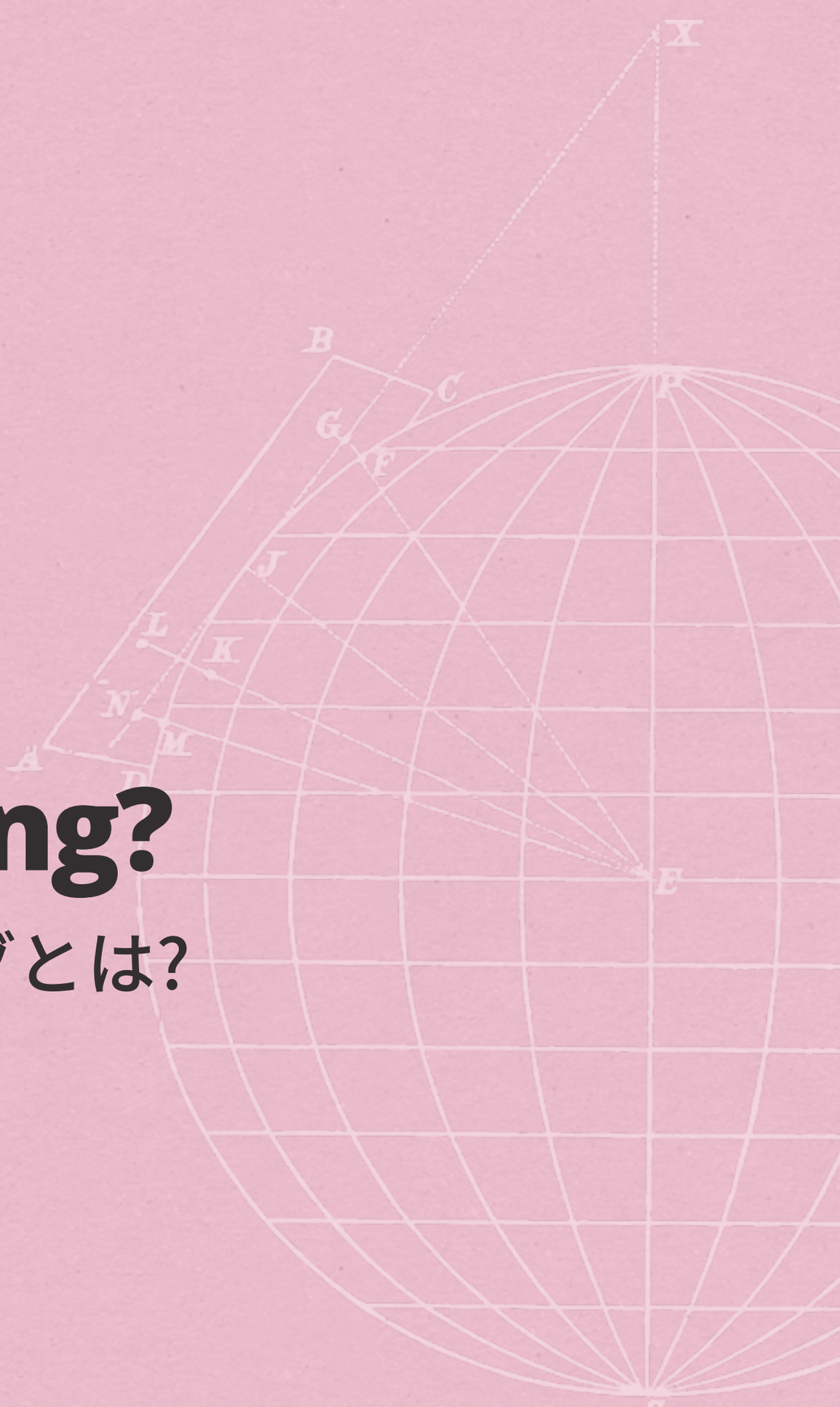
管理されたカオスへの道のり

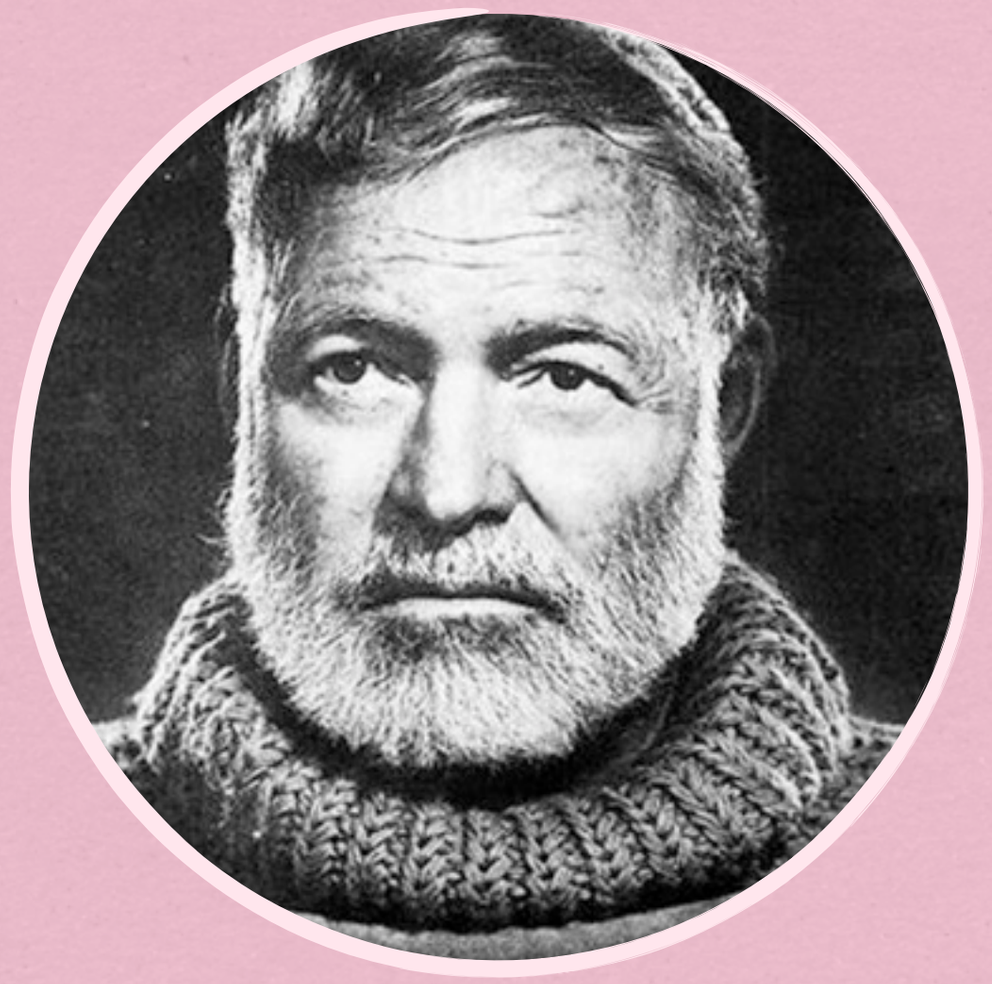
Yuya Saito / Nikkei
Hiroki Tani / tsukuruba
2017.02.25 @ Inside Frontend #1

Act I: Setup

What is, Why Refactoring?

リファクタリングとは?
なぜ必要なのか?





Ernest Hemingway

Novelist

“

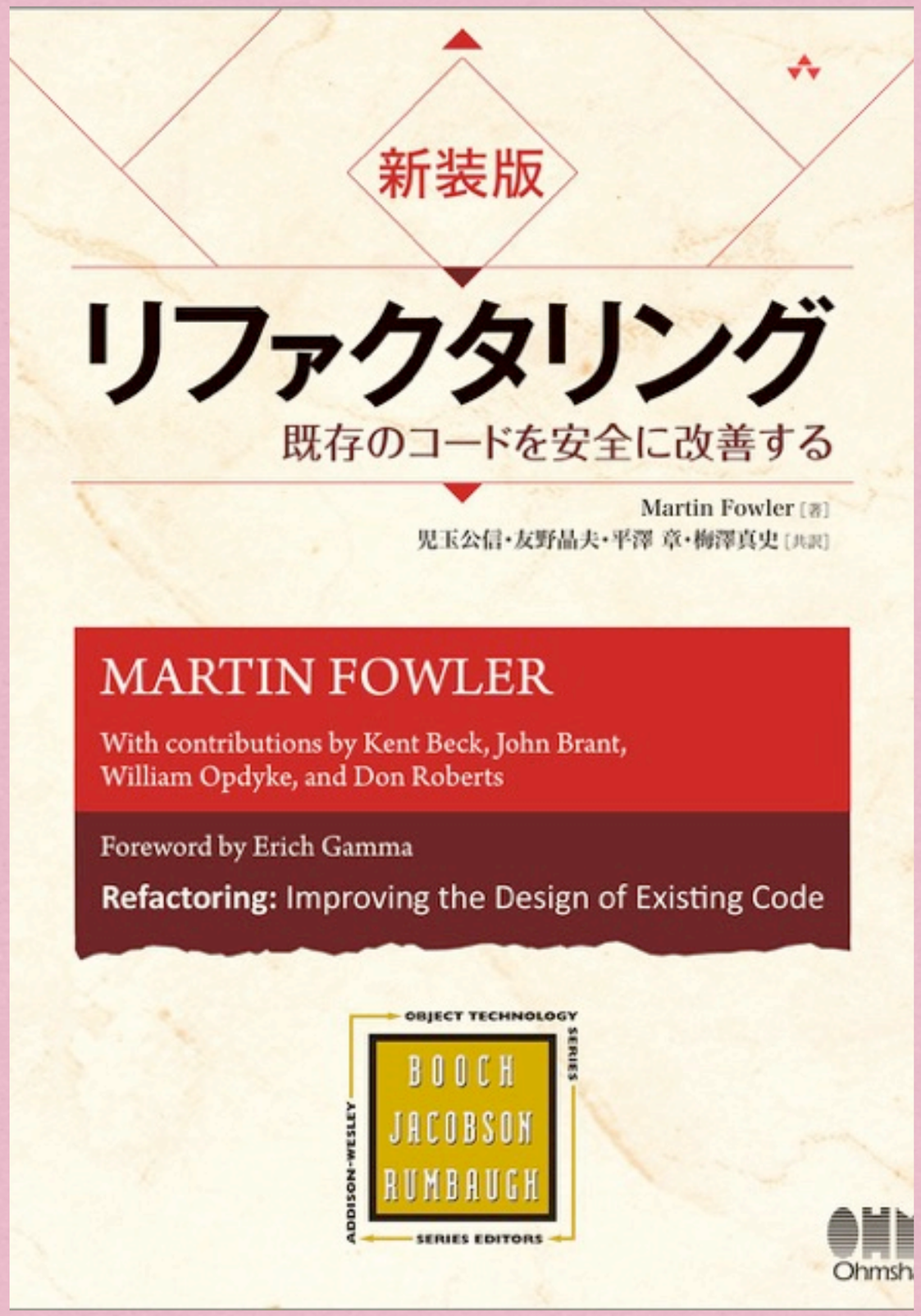
一番最初の下書きはどんな場合でも
イマイチなものだ。

— **Ernest Hemingway**

リファクタリング

by Martin Fowler

amzn.asia/6ohAyxd



> リファクタリングとはソフトウェアの観測できる外部的振る舞いを変えずに、内部構造を改善するための統制されたテクニックである。
>
> — **Martin Fowler**

保守しやすく

拡張しやすい

CSSにおける
リファクタリングには
難しい場面が多くある

UIの変更頻度と変更速度

影響範囲の広さ

テストを書くことが難しい

ROI...?

Act II: In Details

Refactoring When?

リファクタリングは何時するのか?

Always be Refactoring

機能実装前、
バグ修正時

コメントを残す

Code Smells

Rule of Three

1, 3, _, _, _

1, 3, 5, _, _

Refactoring CSS doesn't mean rewriting selectors w/ the n number of dashes and underscores. Focus on what's between the curly braces.

— Mark Otto

Continuous Refactoring

技術的負債の返済時

Refactoring

How?

リファクタリングはどうやるのか?

リファクタリング中は
それだけをする

問題は小さく砕いて解決する

By Brad Frost, Atomic Design

元のソースコードは
消さない

.RF_

```
outline: 1px dotted red;
```

```
all: initial;
```

```
.c-nav-primary {
  font-size: 1rem;
  font-family: sans-serif;
}

.c-nav-primary__item {
  ...
}
```

```
/* Refactoring ...  */
.RF_c-nav-primary {
  all: initial;

  /* all: initial stops inheritance */
  font-size: 2rem;
}
```

jsbin.com/sakatab/edit?html,css,output

File ▾ Add library Share HTML CSS JavaScript Console Output Account Blog Help

HTML ▾

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>Demo: all:initial and .RF</title>
</head>
<body>

<nav class="c-nav-primary RF_c-nav-primary">
  <a href="#0" class="c-nav-primary__item">I'm a navigation item</a>
  <a href="#0" class="c-nav-primary__item">I'm a navigation item</a>
  <a href="#0" class="c-nav-primary__item">I'm a navigation item</a>
  <a href="#0" class="c-nav-primary__item">I'm a navigation item</a>
</nav>

</body>
</html>
```

CSS ▾

```
.c-nav-primary {
  font-size: 1rem;
  font-family: sans-serif;
}

.c-nav-primary__item {
  color: black;
  background: cornsilk;
  display: inline-block;
  padding: 0.5rem;
  text-decoration: none;
  width: 100%;
}

.c-nav-primary__item + .c-nav-primary__item {
  margin-top: 0.5rem;
}

/* Refactoring... */
.RF_c-nav-primary {
  all: initial;

  /* all: initial effectively stops inheritance */
  font-size: 2rem;
}
```

Output Run with JS Auto-run JS

I'm a navigation item

I'm a navigation item

I'm a navigation item

I'm a navigation item

studiomohawk.jsbin.com/sakatab/edit?html,css,output

Bin info

ドキュメント

ドキュメンテーション

≠

スタイルガイド

```
// pattern2
// XL, L        ⇒ M                          ⇒ S
// +------------+ +-TITLE------+ +-------------+ +-------------+
// | IMG        | +-TITLE------+ | IMG         | | IMG         |
// | (OPT)      | +-META-------+ |             | | (OPT)       |
// +------------+ +-EXCERPT----+ +-------------+ +-------------+
// +-TITLE------+ +-0-4 LINE---+                 +-TITLE------+
// +-TITLE------+                                +-TITLE------+
// +-META-------+                                +-META-------+
// +-EXCERPT----+                                +-EXCERPT----+
// +-0-4 LINE---+                                +-0-4 LINE---+
```

```
// nui-card-scale--pattern3
// | breakpoint | font-size(px) | font-size(rem) | line-height |
// |------------|---------------|----------------|-------------|
// | default    | 14            | 0.875          | 1.50        |
// | M          | 18            | 1.125          | 1.33        |
// | XL         | 21            | 1.313          | 1.33        |
```

Refactoring Tips

リファクタリングのコツ

技術的負債の
ご利用は計画的に

技術的負債

≠

古いコード

技術的負債

≠

汚いコード

戦略的で、意識的な
『ショートカット』

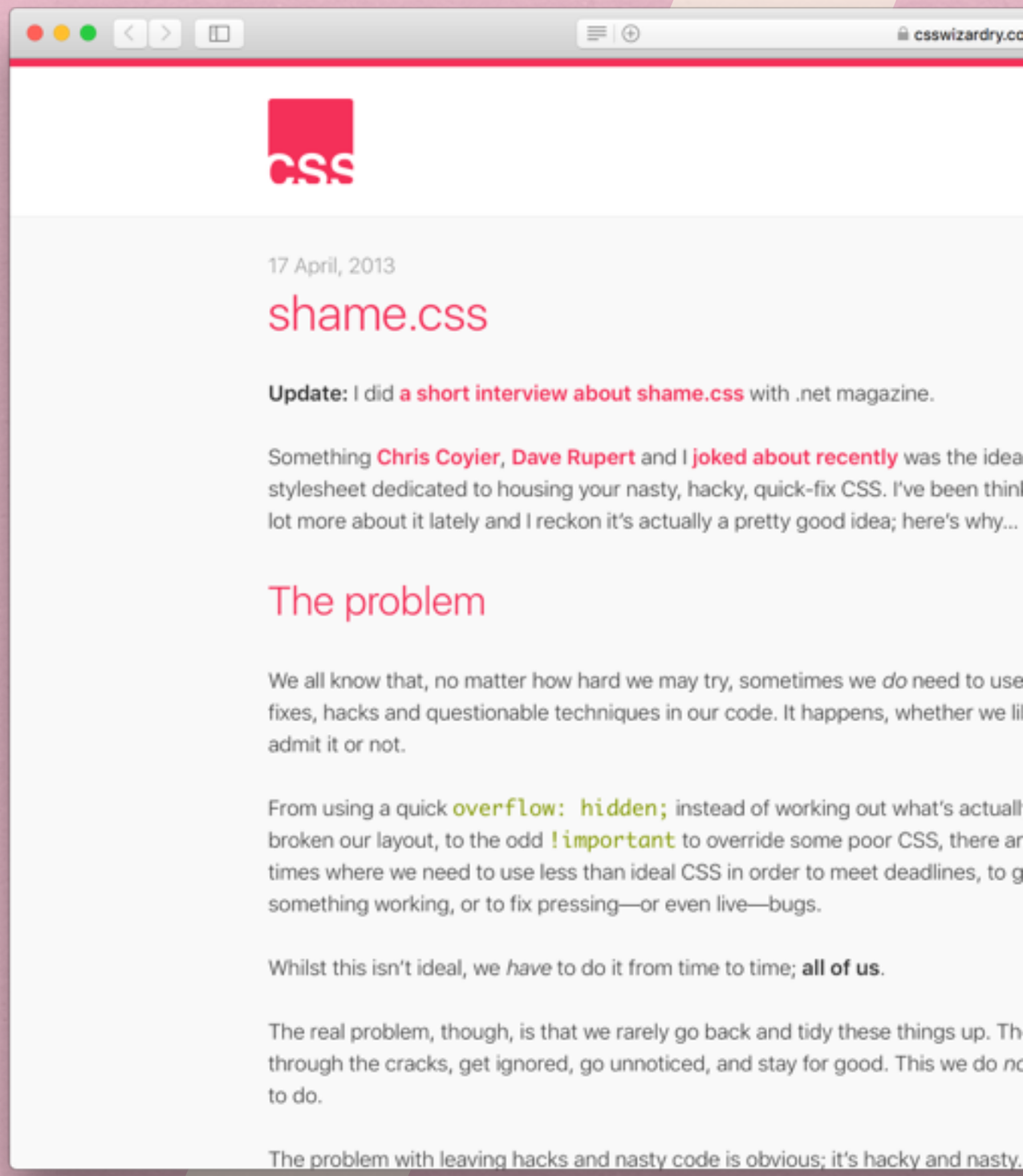
短期的な目標だけではなく
中長期的な視点を持つ

負債を明確にする設計

shame.css

by Harry Roberts

csswizardry.com/2013/04/shame-css



コメントを確実に、
そして詳しく記載する

- コードベースのどこに関連しているのか?
- なぜ必要だったのか?
- どうやって問題を解決しているのか?
- 負債返済の条件

命名規則:

- _ (アンダースコア)

(ちゃんと)

コメントを残す

Act III: Conclusion

残った疑問

- 気がつかない間に負債となって
  しまったものたちをどうするのか?
- リファクタリングの最大の武器、
  テストについて
- リファクタリングと政治

Ask US Anything

結びに

リファクタリングとは、
ひらめきを冷徹な頭で見直すこと。

The Long Goodbye

by Raymond Chandler

amzn.asia/hCsYlF4

自分で自分に仕掛ける罠ほど
たちの悪い罠はない。

— Raymond Chandler

Thank You!

Refactoring CSS

管理されたカオスへの道のり