

コーディングスタイルの理想と現実

music.jp®のリファクタリング before/after

株式会社エムティーアイ 宮地 存





取締役副社長COO 1980.08.06 175.5cm/71.5kg 茨城県古河市出身 専修大学経営学部卒

Twitter: @koba









イベント 2

73

メンバー

■ お問い合わせ

コーダー飲み 12回目 バレンタインDay

② 2015-02-15 (日) 17:00 - 20:00



http://goo.gl/b9kERF





スマートフォンやタブレットなどで利用できる モバイルサイトやアプリを通じて、

音楽配信、健康情報、電子書籍、気象情報など、

日々の生活に密着したサービスを

お客さまに提供する

モバイル・コンテンツ企業です。









音楽と電子書籍が一つのサイトで購入できる イーコマースサイト。

全てのコンテンツはPC/スマート/タブレット など、購入したデバイス以外でも楽しめる クラウドサービス。

音楽や電子書籍など全200万タイトルを 配信している会員サービスです。

リファクタリング

リファクタリング (refactoring) とは コンピュータプログラミングにおいて、 プログラムの外部から見た動作を変えずに ソースコードの内部構造を整理すること。

本日の流れ

- それまではどうやっていたのか?
- 何をしたのか?
- どう変わったのか?
- その後

それまでは どうやっていたのか?

運営体制

事業戦略室	3名
システム開発部	40名
システム運用部	25名
音楽配信事業部	9名
音楽配信サービス部	10名
プロフダクト企画部	15名
コンテンツ統括部	10名
ユーザーセンターデザイン部(UCD)	12名
合計	124名

- UCDは他部署とは独立
- 月に20件前後のPDCA(バナー差し替えなどは除く)

カオスなHTML, CSS

- 1サイト内で3サイトある感じくらいにカオス
- 開発の遅延、デグレの頻出の発生
- ●増える残業時間

宮地さんからの依頼

- 効率的な開発フローの構築
- HTMLはそのままで なんとかできないか

何をしたのか?

- 2013年夏 (今の状態の1個前)
- 150枚くらいのHTMLをテンプレート としてリファクタリング
- 代表的なUIの改修
- スタイルガイドの導入
- 今はGitの導入も

Compass懐かしいですね

- CSSはScss + Compassで
 - Compassでウォッチ
 - ※今はGrunt
- スタイルガイドの導入
 - StyleDocco
 - ※今はKSS

スタイルガイド



はじめに

MDJ Store のコーディングを行う方全員を対象としたスタイルガイドです。 すべてに目を通し、不明な部分があれば管理者ま でご連絡ください。

StyleDocco

- ルートの「htdocs」ディレクトリに移動し「styledocco -n "styledocco" --preprocessor "sass --compass" Content/scss_
- モジュールを追加、変更した場合には必ずリアルタイムに StyleDocco を更新します
- どうしてもリアルタイムに更新できない場合でも、見出しと「後で更新」などのテキストを記載し、後々に更新をおこたるこ とのないように工夫してください

HTML

モジュールやテンプレートなる要素の閉じタグ前にコメント記述します

```
<div class="mod-text">
[内容]
<!-- .mod-text --></div>
```

- その他、不要なコメントの記述はひかえてください
- Retina ディスプレイなど、高精細ディスプレイ用の画像を img 要素で埋め込む場合には width / height 属性どちらかを指 定します
- article、section など、HTML5 のセクショニング・コンテンツはアウトラインが崩れない範囲の利用にとどめてください
- ページに不要な CSS / JavaScript の記述、読み込みは行わないでください

CSS

- class 名はハイフンつなぎです、キャメルケースなどは使用しないでください(既存のモーダルやカルーセルなど、 JavaScript に大きな影響を与える箇所は除外します)
- font-size を指定する場合は rem 単位を使用します、html 要素の文字サイズは 10px になっています
- ベンダブリフィックスの必要なプロパティ、data URI、CSSスプライトは Compass を利用します
- 各モジュールは「mod-モジュール名」の class に格納されます

by music.jp

- 0.0: Overview
- 1: 見出し
- セクションおよびセクション内で使用するモジュール(文字装飾、テーブル、ボックス)
- セクションの装飾(背景色、仕切り線・影)
- 4: ボタン
- 5: アイテムリスト(横)/プレイリスト/メニューリスト
- 6: アイテムリスト(縦)
- 7: 画像とテキストの横並び
- 8: アイコン
- 9: アイコンラベル
- 10: もっと見る
- 11: アコーディオン
- 12: 余白 / 左右、中央寄せ
- 13: バナー
- 14: フォームパーツ
- 15: カルーセル
- 16: ステップ
- 17: ニュース
- 18: ページ内リンクボタン
- 19: 急上昇ワード
- 20: ランキング
- 21: MIXIN (CSSスプライト)
- 22: コピーガード
- 23: ヘッダー (タブ/検索窓)
- 87: 会員プレミアム訴求に関するcssです。

はじめに

このページはportal SP版のスタイルガイドです。 KSSというスタイ また、このスタイルガイドでは、通常のruby版のものではなく、noc

生成はgruntで行います。 CUIでプロジェクトのルートに移動した後

arunt dev

というコマンドを叩くと生成されます。

プロジェクトのルートにある Gruntfile.coffee に設定がありま

仕様

HTML/CSSの記述ルール

- 不要なコードはコメントアウトせずに消去してください。必要が
- img要素には width 属性だけを指定すれば height 属性は不要で
- class 名はハイフンつなぎです。キャメルケースなどは使用しない
- 試しで記述した不要なプロパティは必ず削除してください。
- font-size を指定する場合は rem 単位を使用します。html 要素(
- ベンダプリフィックスの必要なプロパティ、data URI は、comp good example syntax:

```
<figure><img src="path/to/file" width="320"></figure
<div class="mod-btns btns-coin">
<a href="#"></a>
<!-- /.mod-btns --></div>
```

scssの記述ルール

- ! と { の後には1つ分のスペースを入れます。
- 過度はネスト(入れ子)は不用意な肥大化に繋がりますので避け
- インデントはタブひとつ(スペース4つ分)とします。
- mixinは基本的に新たには作りません。
- 複数人での運用のため、過度なextendは避けてください。
- プロパティの並び替えについては、gruntで自動化されていますが good example syntax:

```
.mod-btn {
   border: 1px solid #0f0;
   font-size: 1 2rem:
```

シンプルなSCSSの構成 _tmpl-名前 .scss 各種.scss _mod-名前 .SCSS

シンプルなclassのルール

```
.mod-lyric {
   .lyric-title { ... }
   .lyric-header { ... }
   .lyric-body { ... }
}
```

モジュール内のclassは 「モジュール名-」で開始

引き締める部分は引き締めた

- mixinは管理者の許可を得て
- extendは禁止
- イレギュラーな余白は
 - <hr class="mod-margin height-15">

どう変わったのか?

どのようにして 社内の意を得たのか?

反対意見も

- バックエンダーから反対の声
- 開発費用どんだけかかると思ってんの

どのようにして 保守したのか?

← → C dskd.jp/archives/54.html





















2014/12/10 10:30

僕はCSSを見殺しにした

category: Advent Calendar, CSS

CSS Architecture Advent Calendar 2014の10日目。

それまではけっこう頑張っていた。スタイルガイドも作っていた。デザイナーと コミュニケーションをとり、拡張性のあるパーツを作っていった。新しく触る人 にも読み方や使い方を説明できるようにしていた。

崩壊はUIデザイナーがいなくなった時に始まった。汎用ボタンは使われなくなっ た。決まったルールのデザインエッセンスはなくなった。要素間の空白は誰かの 感覚で変わった。

なぜ止めることができなかったのか。それは、デザインの改修が少しずつ行われ たからだと思う。その改修はいつのまにか始まり、いつ終わるとも決まっていな かった。あらゆるパターンが同居するデザインをCSSは管理できない。改修途中 でも平気でブランチが切られていく。デザイナーがやりたい時にやりたいだけ変 えると知ったとき、僕はCSSを管理するのをやめた。

すでに完成形まで共通化されたCSSでは影響範囲が大きすぎた。それでも既存の ルールセットに手を加えたのは、「いままでの共通パーツを使いまわせる」とい う思いこみがあったからだった。結果的に全ページユニークなデザインになった のを見ると、デザインが変わると決まった瞬間に新しいCSSファイルとセレクタ を作った方が楽だったのだろう。これは明確に僕の予測ミスだし、途中でもそれ に気づいて管理しなおすべきだった。



深まる溝

- モジュールにないものはすべて断った
- デザイナーさんとの仲は最悪に

代表的な数値の変化

- 滞在時間
- ・コンバージョン
- ページビュー数
- ユニークユーザー数

• • • •

スタッフの稼働時間は 改善しましたか?

Aさん

リニューアル前	リニューアル後	
136.73時間	105.19時間	
147.39時間	128.26時間	
105.38時間	89.44時間	
108.37時間	111.25時間	
497.87時間	434.14時間 OF	% FF

Bさん

リニューアル前	リニューアル後	
145.92時間	148.65時間	
172.83時間	117.08時間	
162.34時間	128.94時間	
125.17時間	149.45時間	
606.26時間	544.12時間 Ol	% =F

Cさん

リニューアル前	リニューアル後	
129.75時間	85.25時間	
132.25時間	118.50時間	
138.75時間	95.25時間	
105.25時間	145.50時間	
506.00時間	12° 444.50時間 OF	% F

PDCAサイクルは 改善されましたか?

- リードタイムの短縮により、 より良いサービスをより早く
- 成果物安定により、制作スケジュールが予測しやすい
- アジャイルな制作フローの確率

運用コストは 下がりましたか?

検証にかかるコスト比較

	before	after
外注管理	107時間	9時間
コーディング	9時間	9時間
その他管理	3.5時間	9時間
外部修正	9時間	6時間
検収・品質チェック	105時間	18時間
仕様確認	13時間	15時間
内部修正	34時間	12時間
合計	284時間	78時間

73% OFF

その後

- music.jp® PCサイトへの展開
- 他サービスへの展開

ありがとうございました

https://twitter.com/koba