

研修課題改善案

はじめに






本ドキュメントは、弊社の研修における問題点を改善するため、メンターを担当する者に共有される内容が記述されている。

メンターを担当する者は、本ドキュメントや作成された資料を参照しながら研修生の質疑応答や課題資料の改善に取り組んでいただきたい。

■ 資料作成環境

- 資料作成する際のエディタはVSCodeを使用。
- 作成するファイル形式はMarkdown形式。

```
example.md
```

- Markdown形式の記法については[こちら](#)  を参照。
- VSCodeで作成する際に導入するVSCodeの拡張機能は以下のものを使用。
 - [Markdown All in One](#) 
 - Markdown形式で記述するために必要なものが入ってる
 - [Markdown PDF](#) 
 - 作成したMarkdownファイルをPDFに書き出せる
 - [markdownlint](#) 
 - Markdownで記述する際の補完
 - [Markdown Preview Enhanced](#) 
 - Markdownファイルをプレビューする際にキレイに表示できる
- Markdown Preview Enhancedを使い、VSCode内でプレビュー表示して確認しつつ、適宜PDFに出力して出力結果（主に改ページの制御）を確認していく。

改善案

■ 資料作成について

- Markdownファイルに適用するスタイルは下記のものを使用。**Markdownファイルの冒頭に記述すること。**

```
<link href="https://use.fontawesome.com/releases/v5.0.6/css/all.css"
rel="stylesheet">
```

```
<style>
a::after {
  padding: 0 4px;
  content: "\f35d";
  font-family: "Font Awesome 5 Free";
  font-weight: 900;
  font-size: 0.8rem;
}
```

```
@media print {
  @page { margin: 0; }
  body { margin: 1.6cm; }
}
```

```
h2 {
  border-left: solid 6px #28a745;
  border-bottom: none !important;
  padding-left: 16px;
  height: 40px !important;
  line-height: 40px !important;
  font-weight: bold !important;
}
```

```
h3 {
  padding-left: -14px;
  border-bottom: dashed #ccc 1px;
}
```

```
h3:before {
  content: "■ ";
  color: #17a2b8;
}
```

```
th, td {
  border: solid 1px #666;
}
```

```
table {
  margin-bottom: 30px;
}
```

```
strong {
  color: #000;
}
```

```
hr {  
  margin-bottom: 30px;  
  border-color: #ccc;  
}  
  
blockquote {  
  background: none !important;  
  /* font-style: italic; */  
  color: #999;  
}  
</style>
```

- タイトルはh1(＃)で記述。

タイトル

タイトル

- 大項目はh2(##)で記述。

大項目

大項目

- 中項目はh3(###)で記述。

中項目

■ 中項目

- インデントはスペース2つ。
- PDF出力時に、一定の箇所で自動的に改ページされてしまい文章のキレが悪くなるので、以下のコードを適宜挿入することで改ページを制御すること。
 - 拡張機能のmarkdownlintに修正を求められるが無視して構わない。

```
<!-- 空の改行 -->  
<br>  
<div style="page-break-before:always"></div>  
<br>  
<!-- 空の改行 -->
```

- 問題文は、問題文と実行結果のサンプルと共に読むことで求める正解がわかるように記述するようにすること。
- 実行結果に動的なアクションがある場合は、実行結果のサンプル画像は複数枚用意すること。
- 課題で画像が必要な場合は、必要な画像は用意して課題と同じフォルダー内に配置して渡すこと。

■ 質疑応答について

- 研修生から質問があった場合、まずは質問の内容を確認。研修生がなにを求めているのかを把握し、メンターの中で解答を用意しておく。その後、研修生が自分で解答にたどり着けるように導く。答えは簡単には教えない。研修生自身で考え、自分で答えにたどり着いてもらえるように導き、研修生の自走力を養うような指導を行うこと。
 - エラーが発生して進めない、という場合はエラーの種類を説明し、エラーが発生している原因の見つけ方を教える、といった具合で解答の見つけ方を教える。
- 問題文が読み解けない、問題文の意図がわからない、最終的な実行結果がイメージできないといった場合は問題文の解説を行うこと。
- 確実に質問に対応できるように、全ての課題のサンプルコードを用意しておく必要がある。課題レベルのものであれば、帰社してメンターを担当することになってもコードを読むことで対応することができる。

■ メンターについて

- メンターは、最低でも1週間のうち2～3日以上以上の出勤による研修生への対応が必要であると考えられる。というのも、研修生というのは業界どころかプログラミング未経験者が多いために、Slackで質問しようにも質問そのものをうまく言語化すること自体が難しく、朝と夕方の通話だけでは課題の進みにブレーキをかけてしまっている可能性がある。
- 現状では、研修生同士の交流というのがあまりないため、メンター以外に頼れる人がいないということがある。
- 実際、メンターから声をかけるまで質問せず、わからなかった課題を飛ばして進めている研修生がいる。（オフィスにいるメンターより通話対応のメンターを信用している、人見知り、オフィスにいるメンターに声をかけづらい環境など、声をかけづらい原因は多々ある）そこで、午前午後に関らず1回ずつの通話だけでなく、適度な回数で研修生に進捗を確認し、疑問点や気になる点などを解説して課題に取り組む際のストレスを軽減させてあげるように努めるべきである。

各課題の改善案

■ 各課題に共通する改善案

- 実行結果のサンプル画像は、全てChrome最新バージョンで実行しスクリーンショットで保存したものを使用すること。

■ HTML課題

- 課題をクリアすることで得られる知識、技術について記載する。

▼ HTML課題：課題ごとの改善案

- 1-3
 - 禁止条件をもう少し詳しく書く。

■ CSS課題

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。
- 課題に使用するテキストはコピーで使えるように用意する。(野口氏作成のHTML課題準拠)

▼ CSS課題：課題ごとの改善案

- 2-1
 - 背景に設定する画像を用意しておく。

■ JavaScript課題

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。
- JavaScriptについての説明資料があるが、説明文とコードの見た目の違いが少なく見づらいため、コードを記述する際はコードブロックとして読みやすくする。

`` 拡張子
コード
``

- 課題では関数を用い、とあるが、機能説明のPDFにはJavaScriptの関数についての解説がないため、リファレンス系のサイトでの補足をするか、関数についてなど必要な情報を含めた解説する資料を作成する。

▼ JavaScript課題：課題ごとの改善案

- 1-1～2
 - 制御文という表現がわかりづらいため、代替できる表現を使用すること。

■ jQuery課題

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。
- **jQuery課題**と**jQueryテキスト**という2つのPDF資料があるが、**jQueryテキスト**の方にjQueryについての解説と課題が混在しているため、解説と課題を切り分けた資料を作成する。（**jQuery課題**には**datepicker()**の課題のみが記載されている）
- 実行結果のサンプル画像が小さいため、なるべく大きめのものを用意する。
- **jQuery課題**に記載されている解説では、説明文とコードの見た目の違いが少なく見づらいため、コードを記述する際はコードブロックとして読みやすくする。

`` 拡張子

コード

``

▼ jQuery課題：課題ごとの改善案

- **divでの要素の後に append で入力された色で文字、バックグラウンド色を追加する**
 - 実行結果のサンプル画像が2つしかないうえ、こういった操作による実行結果なのかが記載されていないため、操作の順番に応じたサンプル画像を複数用意する。
- **色々な種類のHTML、CSS、テキストをクリックで変更されることをサンプルで確認する**
 - サンプルがないためサンプルを用意する。

■ PHP基礎課題

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。

▼ PHP基礎課題：課題ごとの改善案

- **1-6**
 - 問題文の日本語が怪しい。数値300を数値365で除算させるような問題文に変更する。
- **1-8~9**
 - 条件以外の情報がないため、もう少しスムーズに課題に取り掛かれるような問題文にする。
- **1-12**
 - 問題文に**なぜ悪いか考え**とあるが、悪いコードの理由がすでに書いてある。間違いのコードだけ示して、**なぜ駄目なのか**と**修正したコード**を書かせるように修正する。
- **2-1~3**
 - 問題文の日本語がおかしく、課題の意図が伝わらないため正しい日本語で意図が伝わるように修正する。
- **配列課題、関数課題**
 - 配列及び関数についての説明がないので、配列について書かれてるリファレンスないし解説を記載する。

■ PHPクラス課題

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。
- クラスについての解説がないため、リファレンス系のサイトのURLか解説を記載する。

▼ PHPクラス課題：課題ごとの改善案

• クラス課題3

- メンバ関数という言葉に聞き馴染みがなく、現場ではメソッドと呼ぶ方が多いため、メソッドに書き換える。

• クラス課題6

- 問題の意図がわかりにくいため、何を求めているのかを記載する。また、複数の解答がある場合はその旨も記載する。

• クッキーとセッション課題全般

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。
- クッキーとセッションについての解説がないため、リファレンス系のサイトのURLか解説を記載する。

• クッキーとセッション課題5

- `session_register`はPHP5.3.0で非推奨、5.4.0で削除されたため、セッション課題の再考自体が必要。

• ファイル課題

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。

■ Bootstrap課題

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。
- 資料としてBootstrapナビゲーションテキストがあるが、ナビゲーションメニューについてしか記述されてないため、リファレンス系サイトを参照した方が早い。
- Bootstrapナビゲーションテキストのバージョンが古い可能性もあるため、資料を用意するのであれば、最新のものに切り替える。

■ DB課題

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。
- 前提条件がないため、CUIでの操作なのかGUIでの操作なのかが不明。前提条件を設定したうえで、問題文を修正。

▼ DB課題：課題ごとの改善案

- 1
 - テーブル定義を記載しているが、これがテーブル定義ですよ、といった一言くらいは必要。
- 2
 - 1と同じテーブルを作るとテーブル名が被ってしまいエラーとなる。名前を変えて全く同じテーブルを作成させるのもいいが、1で作成する2つのテーブルのうち片方をMySQLコマンドで作成させる方に変更した方がよい。
 - コマンドプロンプト自体がわからない場合もあるので、MySQLコマンドを使用するまでの段階的な解説は必要。。
- 3、4
 - 問題文の日本語がおかしく、課題の意図が伝わらないため正しい日本語で意図が伝わるように修正する。
- 5、6
 - 前提条件がないため、課題の意図が伝わらない。課題5ではインデックスの解説を記載。課題6では、課題のクリア条件を記載する。

■ DB連携課題

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。
- 課題下部にDB連携はPDOでの実装とする。できればDBのアクセスクラスを作成して呼び出す。とあるが、これは前提条件として設定する。
- 全般的に問題文が短すぎるかつ日本語が怪しいため、意図が伝わるように書き換える。
- 課題番号が数字じゃなくイロハニホヘトになっているので番号に統一する。

▼ DB連携課題：課題ごとの改善案

- 課題イ
 - 日本語がおかしいため、意図が伝わるように修正。また、CUIでの操作なのかGUIでの操作なのか不明なためその旨も記載する。
- 課題ロ
 - 20件insert文で追加するとあるが、どこまでのフィールドを追加するのかなどの条件がないため記載する。また、CUIでの操作なのかGUIでの操作なのか不明なためその旨も記載する。
- 課題ハ
 - トップ画面から検索メニューと登録メニューのツリーがあるが、トップ画面に検索メニューと登録メニューのリンクがあればいいので、リンクのツリー構造を修正する。
 - 検索条件や登録条件など、簡単な箇条書きでしか書いてないため、課題の詳細な条件を記載する。

■ レイアウト課題

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。

- 課題をクリアすることで得られる知識、技術について記載する。
- 前提条件を記述する。
- 参考にしてほしいリファレンス系のサイトを記載する。
- 資料と課題が混在しており非常に見づらいため、しっかりと切り分ける。
- 説明文とコードの見た目の違いが少なく見づらいため、コードを記述する際はコードブロックとして読みやすくする。また、CUIでのコマンドの入力なのかコードなのかハッキリ分けて記載する。

`` 拡張子

コード

``

- 常体と敬体が混在しているのでどちらかに統一する。（全ての課題全体の方針に合わせるとよい）
- 解説されているバージョンが古いので最新版に修正する。もしくは、バージョンに依存しない内容に留めるようにする。
 - 現在の最新バージョンではデフォルトでBootstrapが入っていないなど、資料作成当時とは大きく違う部分があったりする。
- 現在の課題資料に準拠してLaravelをインストールするのであれば、バージョンを指定してインストールさせるように解説を追記すること。
- ディレクトリの解説があるが不適切なインデントのせいで非常に読みづらいため、スクリーンショットないし見やすい書き方に修正する。
- とにかく読みづらい。