

ポートフォリオ キックオフ

はじめに

1ヶ月目の基礎学習

2ヶ月目の応用課題、チーム開発 お疲れ様でした！

いよいよ3ヶ月目である、**ポートフォリオ制作**へ突入です！

より一層気を引き締めて学習を進めていきましょう！

アジェンダ

1. ポートフォリオ制作とは
2. アプリケーションの基本構成
3. スケジュールと提出物
4. 諸注意
5. 良い成果物にするために
6. 企業はPFのどこをみるか

1. ポートフォリオ制作とは

1. ポートフォリオ制作とは

ポートフォリオとは
あなたの実力を伝えるための制作物
のことです

- ❑ 転職活動の際に企業へ見せるための、アプリケーションを開発します。
- ❑ 構想、設計、実装、デプロイまで全て自身で取り組みます。
- ❑ こんなサービスあったら便利だなと思うアプリケーションを開発しましょう。
- ❑ 今まで学んできたことの集大成です！

以降、ポートフォリオは「PF」と記載します。



2. アプリケーション開発で 利用する技術の基本構成

2. アプリケーション開発で利用する技術の基本構成

ここで言う基本構成とは、DWCとしてポートフォリオ制作で指定させていただく言語・フレームワーク・開発環境などを指します。

世の中には様々な言語、フレームワークが存在しますが
2ヶ月間の学習成果をフル活用できるよう、ある程度の技術要素については
指定をさせていただいております。

メンターによる技術サポートについても
基本構成をベースに対応させていただきます。

2. アプリケーション開発で利用する技術の基本構成

フロントエンド

- **HTML/CSS**
 - Bootstrap
 - Sass
 - テンプレートエンジン(slim,hamlなど)
- **JavaScript**
- **JavaScriptライブラリ**
 - jQuery
 - カリキュラムで学んだもの・チャレンジ要素一覧(後述)に記載のあるものを参考にしてください。
 - 難易度について確認が必要な場合は、メンターや運営部までご相談ください。

2. アプリケーションの基本構成

バックエンド

- **Ruby on Rails 5.x.x**

- 最新版は6ですが、6はまだ情報が少なく利用されている現場も多くなく、ネットにあるRails関連の記事や情報も、5が多いです。
- 6をインストールしてネットの情報をそのまま使うと事故が起きる可能性が高いです。

- **Gem**

- カリキュラムで学んだもの・チャレンジ要素一覧に記載のあるものを参考にしてください。
- 難易度について確認が必要な場合は、メンターや運営部までご相談ください。

- **外部API**(Google Maps API, ぐるなびAPIなど)

赤字：必須構成

2. アプリケーションの基本構成

インフラ

- **AWS(Amazon Web Service)**：本番環境として使用
 - **EC2**：アプリケーションのデプロイに使用
 - **RDS**：アプリケーションのデータベースとして使用
 - **S3**：アプリケーションのストレージとして使用
 - **Route53**：DNSの設定に使用
- **Vagrant**：ローカル開発環境として使用
 - カリキュラムにおいて、既に構築済みのものを使用

赤字：必須構成

3. スケジュールと提出物

3. スケジュールと提出物

締切	提出物	備考
10/19 23:59	テーマ（詳細 + 背景）+ チャレンジ要素一覧	レビュー有 / 公序良俗に反しないもの
10/26 23:59	WBS	レビュー有 / 設計後に提出
10/25 10:00	PF実装キックオフ	諸注意・デイリースクラム
10/31 23:59	テストデプロイURL	提出のみ
10/31 23:59	進捗報告(1回目)	提出のみ
11/5 23:59	進捗報告(2回目)	提出のみ
11/10 23:59	進捗報告(3回目)	提出のみ
11/13 23:59	<ul style="list-style-type: none">自己評価/振り返り完了報告	提出のみ
11/14,15	発表会	※提出方法については追って説明します。

3. スケジュールと提出物 ~テーマ,チャレンジ要素一覧を作成/提出~

締切	提出物	備考
10/19 23:59	テーマ（詳細 + 背景） + チャレンジ要素一覧	レビュー有 / 公序良俗に反しないもの
10/26 23:59	WBS	レビュー有 / 設計後に提出
10/25 10:00	PF実装キックオフ	諸注意・デイリースクラム
10/31 23:59	テストデプロイURL	提出のみ
10/31 23:59	進捗報告(1回目)	提出のみ
11/5 23:59	進捗報告(2回目)	提出のみ
11/10 23:59	進捗報告(3回目)	提出のみ
11/13 23:59	<ul style="list-style-type: none">自己評価/振り返り完了報告	提出のみ
11/14,15	発表会	※提出方法については追って説明します。

3. スケジュールと提出物 ~テーマ,チャレンジ要素一覧を作成/提出~

- ❑ まずは**PFのテーマ**を作成しましょう。
 - ❑ アプリケーションのREADMEに記載していただきます。
 - ❑ テーマの雛形は <https://is.gd/jOU16S> です。
 - ❑ <>で囲まれている箇所に記載してください。<>も削除しましょう。

```
# <ここにアプリ名を入力>

## サイト概要
<...>

### サイトテーマ
<...>
```

- ❑ 「開発環境」で、使わない要素や追加要素があれば編集しておきましょう。
- ❑ 外部の素材を使用するときは、サービス名とURLを明記しましょう。
- ❑ こういうサイトがあると便利だなというイメージを文章化しましょう。

3. スケジュールと提出物 ~テーマ,チャレンジ要素一覧を作成/提出~

- 次に**チャレンジ要素一覧**を作成しましょう。
 - 雛形 (<https://ux.nu/AipeD>) をコピーしてご利用ください。
(ファイル名は必ず「**チャレンジ要素一覧**」とすること)



- あらかじめ実装方法を調べ、方法の目処が立ったものは☑をいれてください。(G列)
- 作成後、README.mdにリンクを貼りましょう。(閲覧権限にお気をつけください)

3. スケジュールと提出物 ~チャレンジ要素一覧補足~

- チャレンジ要素一覧はこちらの中から選択してください。

<https://ux.nu/CXwCB>

- PF作成における、
最低限押さえていただきたい技術と技術要素の優先順位/学習難易度が記載されています。
- 一覧は過去の受講生様の取り組みや企業様からの評価を元に作成しました。
PFに「HIGH」「MIDDLE」のものをそれぞれ2~4ほど盛り込むのが目安です。
- 学習コストを目安に、作成する機能の優先順位付け(HIGH, MIDDLE)や、機能にはない非機能（環境変数化や、テストコード等はサービスの機能のため、非機能といいます）のチャレンジもしてみてください。

※ 専門技術講座を受講されている方は作成した機能一覧のファイル名にコース名の記載をお願いします。

例：チャレンジ要素一覧(クラウド教養コース), チャレンジ要素一覧(AI教養コース)

※ 「AI教養コース」を選択されている方は、PFに組み込む想定 of AI機能の概要も追加してください。

3. スケジュールと提出物 ~チャレンジ要素一覧補足~

作成したテーマとチャレンジ要素一覧はGitHubの**README.md**に記載し、
リポジトリURLを提出してください。

PFはそのリポジトリを運用していきましょう！

3. スケジュールと提出物 ~チャレンジ要素一覧補足~

★注意点★

- ❑ 難易度のわからない技術に関しては**必ずメンターに相談**してください。
 - ❑ 過去、特定の機能や技術を作り込むことにこだわりすぎたために、**ほとんどPFが作れなかったケース**もあります。
- ❑ コピペで機能を実装せず、README.mdやリファレンスを確認し、**理解して使う**こと。
 - ❑ 企業によってはGitHubを確認します。
 - ❑ **制作者自身で書いたコードかどうかを簡単に見抜きます**。
- ❑ メンターが全ての技術を把握している訳ではないことをご了承ください。
 - ❑ ご存知の通り、技術の範囲は広く、それぞれの得手不得手や、これまで触れてきた技術のバックグラウンドによって対応範囲は異なります。

3. スケジュールと提出物 ~チャレンジ要素一覧補足~

以下の技術に関しては現在の皆さんの技術力と相談となります。

- 実装に大幅な工数がかかるため。
- 実装できても、技術の理解が不十分のままになってしまうことが懸念されるため。

- **ActionCable**
- **Docker**
- **Vue/React等のJSフレームワーク**

また、上記以外にもチャレンジする技術として不安を感じるものがあれば
必ずメンターに相談してください。

3. スケジュールと提出物 ~PF制作で最も重要なこと~

□ PF制作で最も重要なこと

PF制作は自由テーマではありますが、
目的はこれまで身につけたスキルを展開し、
そこから更に発展させて企業に自身のスキルをアピールするものです。

そのため、好きなサービスを作ることが目的ではなく、

期日までに完成（正しく動作）させること、

正しいコードを書くこと、技術を理解して扱うことが大切になります。

3. スケジュールと提出物 ~PF制作で最も重要なこと~

現在の自身の技術力に不安のある方や、

2ヶ月目の応用課題が十分に進められなかった方へ

□ 前提

- 2ヶ月目の応用課題の達成がday3まで。もしくは、Bookers2(課題3)に理解できていない箇所がある。

□ 参考テーマ

- 自身の力でBookers2の機能をベースとして実装し、そこへ機能追加をするテーマにしましょう。
- Bookers2はSNSサービスの基本機能を持っています。
- SNSサービスとして、何を交流・情報交換するサービスなのか？「好きな写真」「好きな料理」等、これまでのご自身の経験・経歴などを元に何かの情報を投稿・共有するサービスをテーマにしてもよいと思います。

3. スケジュールと提出物 ~PF制作で最も重要なこと~

□ 実装する機能例

- 実装する機能は以下の優先順位で取り組まれると、着実に進められます。

Bookers2相当機能+デプロイ+リーダブルコード

- ユーザ機能
- コメント機能
- いいね機能
- デプロイ
- リーダブルコードの意識
(インデント、変数名を正しく書く)

発展機能

※デプロイ完了後、更に技術力を上げるため、以下を取り入れるとよいです。

- Rspec
- Rubocop
- JavaScript(jQuery)
- 開発スキルアップからピックアップして実装

3. スケジュールと提出物 ~PF制作で最も重要なこと~

□ 前提

- 2ヶ月目の応用課題の達成がday6まで。もしくは、Bookers2(課題3)が理解できており、修了していること。

□ 実装する機能例

- 実装する機能は以下の優先順位で取り組まれると、着実に進められます。

Bookers2相当機能+デプロイ+リーダブルコード

- ユーザ機能
- コメント機能(非同期通信を使用)
- いいね機能(非同期通信を使用)
- ランキング機能
- 検索機能
- デプロイ
- リーダブルコードの意識 (インデント、変数名を正しく書く)

発展機能

※デプロイ完了後、更に技術力を上げるため、以下を取り入れるとよいです。

- Rspec
- Rubocop
- JavaScript(jQuery)
- 開発スキルアップからピックアップして実装

3. スケジュールと提出物 ~テーマ/チャレンジ要素一覧のセルフチェック~

テーマ/機能一覧の提出前に、以下項目を満たしているか確認しましょう。
すべて満たしていたら提出しましょう！

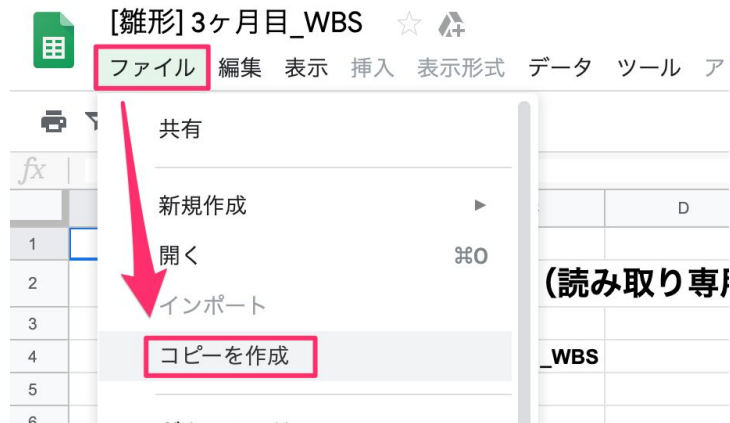
- ✓ : PFのRailsアプリケーションのREADME.mdに記載していますか？
- ✓ : README.mdに<>は残っていませんか？必要でなければ削除しましょう。
- ✓ : チャレンジ要素一覧の閲覧権限は「リンクを知っている全員が閲覧可」になっていますか？
- ✓ : チャレンジ要素一覧のファイル名に「のコピー」は残っていませんか？
- ✓ : チャレンジ要素一覧の「参考サイトやイメージが湧くアプリがあればそのURLを記入ください」
列にカリキュラムのURLや応用課題のURLを記載していませんか？
(受講している皆さんにお渡ししている資料は基本的に外部公開を禁止しています)

3. スケジュールと提出物 ~WBS~

締切	提出物	備考
10/19 23:59	テーマ（詳細 + 背景） + チャレンジ要素一覧	レビュー有 / 公序良俗に反しないもの
10/26 23:59	WBS	レビュー有 / 設計後に提出
10/25 10:00	PF実装キックオフ	諸注意・デイリースクラム
10/31 23:59	テストデプロイURL	提出のみ
10/31 23:59	進捗報告(1回目)	提出のみ
11/5 23:59	進捗報告(2回目)	提出のみ
11/10 23:59	進捗報告(3回目)	提出のみ
11/13 23:59	<ul style="list-style-type: none">自己評価/振り返り完了報告	提出のみ
11/14,15	発表会	※提出方法については追って説明します。

3. スケジュールと提出物 ~WBS~

- ❑ WBSはテーマ/チャレンジ要素一覧が**合格後に雛形を共有**します。
雛形のコピーを作成し、使用してください。
- ❑ コピー方法についてはメニューの[ファイル]を選択し、[コピーを作成]をクリックします。
右の画像を参考に行ってください。
- ❑ ファイル名は必ず「**アプリケーション名_WBS**」としましょう。



3. スケジュールと提出物 ~WBSチェック項目~

WBS提出前に、以下項目を満たしているか確認しましょう。

すべて満たしていたら提出しましょう！

- ❑ ファイル名が「アプリケーション名_WBS」となっている
- ❑ 「プロジェクト概要」が書かれている
- ❑ 「ポートフォリオテーマ」が書かれている
- ❑ デプロイ(テストデプロイ)は「AWS設定、nginx, puma設定、アプリケーション修正」の3項目がある
- ❑ デプロイ(テストデプロイ)が3日以上設けられている
- ❑ テストが0.5~1日設けられている
- ❑ WBSの体裁が整っている(日付が合っているか、誤字脱字がないか)
- ❑ 無理のない範囲の日程となっている
 - ❑ 以下の機能は1日での実装としないこと。必ず2,3日のバッファを設ける。
 - ❑ Rails(多言語化、予約)
 - ❑ JavaScript(インクリメンタルサーチ、非同期通信、録音機能、ガントチャート、データ可視化)
 - ❑ その他(例外処理、バッチ、N+1問題、S3)
- ❑ タスクの確認(機能とレイアウトでタスクが分けられている)

3. スケジュールと提出物 ~PF実装キックオフ~

締切	提出物	備考
10/19 23:59	テーマ（詳細 + 背景） + チャレンジ要素一覧	レビュー有 / 公序良俗に反しないもの
10/26 23:59	WBS	レビュー有 / 設計後に提出
10/25 10:00	PF実装キックオフ	諸注意・デイリースクラム
10/31 23:59	テストデプロイURL	提出のみ
10/31 23:59	進捗報告(1回目)	提出のみ
11/5 23:59	進捗報告(2回目)	提出のみ
11/10 23:59	進捗報告(3回目)	提出のみ
11/13 23:59	<ul style="list-style-type: none">自己評価/振り返り完了報告	提出のみ
11/14,15	発表会	※提出方法については追って説明します。

3. スケジュールと提出物 ~PF実装キックオフ~

開催予定日 : 10/25(日) 10:00~12:00

※詳細は月生チャンネルで連絡致します。

PF制作をスムーズに取り組めるように、

運営部から皆さんへ是非取り組んで欲しいことをご紹介します！



3. スケジュールと提出物 ~テストデプロイ~

締切	提出物	備考
10/19 23:59	テーマ（詳細 + 背景） + チャレンジ要素一覧	レビュー有 / 公序良俗に反しないもの
10/26 23:59	WBS	レビュー有 / 設計後に提出
10/25 10:00	PF実装キックオフ	諸注意・デイリースクラム
10/31 23:59	テストデプロイURL	提出のみ
10/31 23:59	進捗報告(1回目)	提出のみ
11/5 23:59	進捗報告(2回目)	提出のみ
11/10 23:59	進捗報告(3回目)	提出のみ
11/13 23:59	<ul style="list-style-type: none">自己評価/振り返り完了報告	提出のみ
11/14,15	発表会	※提出方法については追って説明します。

3. スケジュールと提出物 ~テストデプロイ~

- ❑ PFは企業が確認できるように、ネット上に公開する必要があります。
- ❑ アプリケーションをネット上に公開する作業を通称、デプロイと言います。
- ❑ 転職活動をするために、この作業は必須です。
(企業の方が実際にPFを閲覧・操作して確認する場合があるため)
- ❑ デプロイ作業は工程が複雑で、初心者にとってはエラーが起こりやすく苦戦します。
- ❑ 制作の中盤で行うこのテストデプロイを通して、早期の内に慣れておきましょう！



3. スケジュールと提出物 ~テストデプロイ~

提出内容: **デプロイしたアプリケーションのURL**

※レビューなし/提出のみ

- ❑ 自身のPFを一度デプロイしてみましょう。
 - ❑ 必ず自身のPFを使用して下さい。
- ❑ デプロイを行う際はAWS（Amazon Web Service）を利用するので、AWSアカウントの登録が必要です。
 - ❑ 登録にはカード情報が必要ですが、クレジットカード以外にもデビットカード、バンドルカード等でも登録が可能です。



3. スケジュールと提出物 ~進捗報告~

締切	提出物	備考
10/19 23:59	テーマ（詳細 + 背景） + チャレンジ要素一覧	レビュー有 / 公序良俗に反しないもの
10/26 23:59	WBS	レビュー有 / 設計後に提出
10/25 10:00	PF実装キックオフ	諸注意・デイリースクラム
10/31 23:59	テストデプロイURL	提出のみ
10/31 23:59	進捗報告(1回目)	提出のみ
11/5 23:59	進捗報告(2回目)	提出のみ
11/10 23:59	進捗報告(3回目)	提出のみ
11/13 23:59	<ul style="list-style-type: none">自己評価/振り返り完了報告	提出のみ
11/14,15	発表会	※提出方法については追って説明します。

3. スケジュールと提出物 ~進捗報告~

下記の3点をご報告いただきます。

- 進捗状況（遅れ/予定通り/前倒し）
- 一週間を振り返って
- 次週までの目標

報告内容を元にサポートさせていただくこともあるので、**忘れずに必ず提出**してください！

「遅れているから報告できない...」ではありません！
報連相は開発現場で働く上で、**基本中の基本**です。

3. スケジュールと提出物 ~自己評価/振り返り、完了報告~

締切	提出物	備考
10/19 23:59	テーマ（詳細 + 背景） + チャレンジ要素一覧	レビュー有 / 公序良俗に反しないもの
10/26 23:59	WBS	レビュー有 / 設計後に提出
10/25	PF実装キックオフ	諸注意・デイリースクラム
10/31 23:59	テストデプロイURL	提出のみ
10/31 23:59	進捗報告(1回目)	提出のみ
11/5 23:59	進捗報告(2回目)	提出のみ
11/10 23:59	進捗報告(3回目)	提出のみ
11/13 23:59	<ul style="list-style-type: none">自己評価/振り返り完了報告	提出のみ
11/14,15	発表会	※提出方法については追って説明します。

3. スケジュールと提出物 ~自己評価/振り返り、完了報告~

PFが完成したら、下記を送信いただきます。

- **自己評価/振り返り**

技術要素やアピールポイントなど

- **完了報告**

デプロイURLや提出前の確認事項など

間に合いそうにない場合は、必ず事前にご連絡ください。
無断で納品を遅らせるようなことは、開発現場ではありえません。

3. スケジュールと提出物 ~発表会~

締切	提出物	備考
10/19 23:59	テーマ（詳細 + 背景） + チャレンジ要素一覧	レビュー有 / 公序良俗に反しないもの
10/26 23:59	WBS	レビュー有 / 設計後に提出
10/25	PF実装キックオフ	諸注意・デイリースクラム
10/31 23:59	テストデプロイURL	提出のみ
10/31 23:59	進捗報告(1回目)	提出のみ
11/5 23:59	進捗報告(2回目)	提出のみ
11/10 23:59	進捗報告(3回目)	提出のみ
11/13 23:59	<ul style="list-style-type: none">自己評価/振り返り完了報告	提出のみ
11/14,15	発表会	※提出方法については追って説明します。

3. スケジュールと提出物 ~発表会~

発表会で自身のPFを発表していただきます。

日程：11/14,15

（詳細は後日、月生チャンネルにてアナウンスさせていただきます）

13日のデプロイ期限を納期とし、
アプリケーションの実装はもちろん、本番でしっかりと
発表できるところまで意識して取り組みましょう！！



3. スケジュールと提出物 ~提出方法~

ポートフォリオ連絡フォーム

<https://forms.gle/a4rvfcHzEC5zdQ2o8>

- ❑ 制作中に要求される提出物はこちらのフォームから送信してください。
- ❑ フォームはブラウザなどのブックマークに入れておきましょう！



ポートフォリオ連絡フォーム

*必須

氏名(フルネーム) *

苗字と名前の間にスペース無し

回答を入力

受講開始年月 *

例) 2018年1月生の場合: 201801 (6桁で入力)

回答を入力

教室 *

選択 ▼

次へ

Google フォームでパスワードを送信しないでください。

3. スケジュールと提出物 ~提出方法~

ポートフォリオ連絡フォーム

ポートフォリオ連絡フォーム

ポートフォリオ制作

連絡内容を選択して下さい。

- ☐ ポートフォリオ作成内容について
- ☐ WBS提出フォーム
- ☐ テストデプロイURL提出フォーム
- ☐ 進捗報告
- ☐ ポートフォリオ完了報告
- ☐ ポートフォリオ制作_自己評価/振り返り

[戻る](#) [次へ](#)

Google フォームでパスワードを送信しないでください。

□ **該当する内容**を選択してください。

4. 諸注意

4. 諸注意

- ❑ メンターへの質問
- ❑ 提出物のレビュー返却時間と返却までにすること
- ❑ デプロイとGitHub
- ❑ スキル評価について
- ❑ 外部リンク・画像などの使用について
- ❑ Remoについて

4. 諸注意 ~メンターへの質問~

メンター対応時間



フェーズ1 : 13:00~22:00

フェーズ1,2 : 13:00~22:00

フェーズ3,4 : 19:00~22:00(月~木)

13:00~22:00(土、日)

※祝・祭日も各曜日に準ずる

※質問受付の締め切りは21:30となります。

質問対応範囲の早見表: <https://bit.ly/2LXPLL1>

4. 諸注意 ~メンターへの質問~

メンターへの質問時間が限られるため、時間を有効活用しましょう！

❑ メンターへ質問ができる時間

- ❑ 調べてもわからないことを質問してみましょう
- ❑ 理解が浅い箇所を実装しましょう

❑ メンターがいない時間

- ❑ 自分で調べながら実装していきましょう
- ❑ 質問できる時間が近づいてきたら質問したい箇所の学習を進めましょう

4. 諸注意 ~メンターへの質問~

「基本構成」の内容を、メンターの質問サポート範囲としております。

「基本構成」以外のご質問につきましては、原則として問題の切り分け方や調べ方などの、アドバイスをお伝えします。アドバイスを参考に、ご自身で解決していただく形になります。

現役エンジニアのメンターは「基本構成」の技術をベースとして採用しています。「基本構成」以外の対応をする場合、非常に時間がかかる、もしくは解消しないことがあるため、上記の対応とさせていただきます。あらかじめご了承ください。

4. 諸注意 ~提出物のレビュー返却時間と返却までにすること~

返却時間

当日17:00までに提出されたものは、最低、翌々日の13:00までに返却

※休校日である金曜日は日数に含みません

以下の提出物(テーマ/チャレンジ要素一覧)のレビュー返却には時間を要します。
時間を有効活用するために、レビュー返却まで以下を進めてみましょう。

- ❑ 実装する機能について調べてみましょう
 - ❑ 機能の実装方法についてREADME.mdやリファレンス、記事を探してみましょう
- ❑ デザインを参考にしてみたいサイトを探してみましょう
 - ❑ 「WEBデザイン 参考」などと検索し探してみましょう
- ❑ 応用課題やカリキュラム(主にRails)の復習をしてみましょう
 - ❑ Railsに関してわからないことが多い人は復習もしてみましょう

4. 諸注意 ~デプロイとGitHub(1/2)~

❑ 必ずデプロイまで終えること

- ❑ 職務経歴書にデプロイしたPFのリンクを記載します。
- ❑ **メンターがサポートできる3ヶ月目以内**に終えること。
- ❑ WCO：デプロイを学ぼう
- ❑ **専門技術講座の修了条件となります。受講の方は厳守してください。**
 - ❑ PF完成することで3ヵ月目の開校日をすべて出席とする。
PFの完成とは、**デプロイが終了し、PFが動作していることを指す。**

❑ GitHubにも気を配る

- ❑ 就職活動の際、GitHubを見られることがあります。
- ❑ **コミットメッセージなどはわかりやすく設定すること。**

4. 諸注意 ~デプロイとGitHub(2/2)~

□ Gitのコミットはこまめに行うこと

- **怠ると、ソースコードを失う可能性があります。**
- 実際に、コミットを1度もしていなかった受講生が、誤って「rmコマンド（ファイル・ディレクトリの削除コマンド）」をPFのソースコード全体に対して実行してしまい、**1ヶ月かけて作ったアプリケーションを失った**、という事象がありました。コミットをしていれば、元に戻せたはずです。
- Gitへのコミットは**ゲームのセーブ機能**のようなものです。頑張りをリセットしてしまわないように、こまめに行いましょう。

4. 諸注意 ~スキル評価について~

短期集中講座の方へ

転職活動の際に必要な、スキル評価を3ヶ月目終了後実施します。
制作したPFにアクセスし、評価するため以下についてご注意ください。

- ❑ **学習期間から10日間はデプロイしたPFを稼働状態**にしてください。
※稼働状態=EC2,RDSインスタンス等が起動され、PFがネット上に公開されている状態
- ❑ 稼働状態の継続にあたり、AWSが有料枠に変更された場合は**料金が発生する可能性**があります。
(料金についてはご自身で常にご注視ください。不明点は運営部までご連絡ください)

スキル評価が終了した際は運営部から連絡いたします。
以後は皆さんの判断でデプロイを停止・稼働を行ってください。
専門技術生は4ヶ月目も学習となりますので、スキル評価は別タイミングとなります。

4. 諸注意 ~外部リンク・画像などの使用について(1/3)~

<前提>

外部の情報やリソースの利用する場合、後続のガイドラインを参考に自身の責任のもと、利用するようお願いいたします。

第三者の権利を侵害した場合や、権利侵害が発覚し就職活動へ影響が出た場合であっても、弊社は責任を負いかねますので、予めご了承ください。

4. 諸注意 ~外部リンク・画像などの使用について(2/3)~

画像の取扱い

❑ 利用が許可される場合

- ❑ 著作権フリー画像や著作権者の利用許諾を得られた場合
 - ❑ 著作権フリー画像のサイトポリシーや著作権者の利用条件に沿った形式であること
 - ❑ 出典を明記すること
- ❑ ご自身で撮影、若しくは作成された画像であること

❑ 利用が許可されない場合

- ❑ Web上にアップロードされている他人の画像を無断でダウンロードし、再公開（サーバーアップロード）すること

4. 諸注意 ~外部リンク・画像などの使用について(3/3)~

リンクの取扱い

- ❑ リンク先サイトのポリシーに従い、利用可否を判断した上で利用
 - ❑ リンクを掲載する行為自体が、著作権法その他法令に違反することは原則ありません。
 - ❑ リンク先サイトのポリシーを確認し適切に利用してください。

画像の取扱い (直リンクを利用する場合)

- ❑ リンク先サイトのポリシーに従い、利用可否を判断した上で利用
 - ❑ 出典を明記すること
 - ❑ リンク同様、画像の直リンク自体が法令違反となることは原則ありませんが、リンク先サイトのポリシーを確認し適切に利用してください。

APIの取扱い

- ❑ API連携の際は、APIの公開元のポリシーに従い利用可否を判断した上で利用

4. 諸注意 ~Remoについて~

Remoは学習フェーズIII以降も開放しております。

※座席の変更はありません。

注意事項

- ① Zoomでメンターへ質問する際は、Remoのマイクをミュートすること
- ② 相手の時間を尊重しつつ、他の受講生とコミュニケーションをとりましょう
- ③ 3ヶ月目のRemoへの参加は任意です

4. 諸注意 (転職保証について)

▶ 【重要】転職保証について

※下記の条件に該当する場合、転職保証が外れる可能性があります

- 学習フェーズ全期間
 - 期間中5日以上連絡が取れない
 - 当社が設定したイベントに無断で欠席
 - 学習フェーズIII
 - PFの成果物を期間内に完成、提出していない
- ※PFの完成とは、デプロイが終了し、PFが動作していることを指す
- ※デプロイの際は必ずAWSを利用すること

4. 諸注意 (専門技術講座を受講の方へ)

▶ 【重要】修了証発行条件

※学習フェーズIIIの学習期間にて、下記の条件を満たす必要があります

- 学習フェーズIII
 - PF完成することで3ヵ月目の課程を修了とする
 - PFの完成とは、デプロイが終了し、PFが動作していることを指す
- ※デプロイの際は必ずAWSを利用すること

5. 良い成果物にするために

5. 良い成果物にするために ~README.mdを詳しく書こう~

「GitHub README 書き方」などで検索すると、良い書き方が掲載されているサイトがHITします。

📖 README.md

README

Tooling

このサイトはバイク乗りのコミュニケーションを図るための投稿サイトです 基本機能は 新規投稿、その投稿にコメントができます。また都道府県での検索やカテゴリー別での検索結果表示もできます。一週間ごとのランキングを確認することもできます。

Dependency

ruby 2.5.5 Rails 5.2.3

Usage

ユーザー機能・新規投稿のリンクをクリックすると新しく投稿するための入力画面がでるのですべての欄を入力すると投稿することができます。メールアドレスがなくても投稿することができます。カテゴリーが必須です。投稿後は自分の投稿！

5. 良い成果物にするために ~PFにaboutページを作ろう~



参考サイト：

<https://www.satofull.jp/static/instruction01.php>

<https://drinx.kirin.co.jp/about/>

about ログイン 新規会員登録

名言-POSTの使い方

- ①当サイトは名言のSNSです。
- ②あなたが耳にしたその一言、実は名言かもしれません。
- ③発言者の区分が著名人の場合は、Wikipediaへリンクすることが可能です。
- ④会員登録することにより、名言の投稿やお気に入りリストの管理が可能です。
- ⑤一度投稿した名言は解説のみ編集可能です。
- ⑥発言者の職業、人名で名言の検索が可能です。

5. 良い成果物にするために ~なぜPFを作るのかを常に思考しよう~

PF制作で見えるもの

技術力以外にも、
その人の**考え・スタンス**がわかる



5. 良い成果物にするために ~なぜPFを作るのかを常に思考しよう~

- ❑ なぜその題材を選んだのか
- ❑ 選んだ技術で何を実現したいのか
- ❑ コードを書く時にどんな事に気を配っているのか
- ❑ 誰も見えやすいことを意識してるのか
- ❑ サービスの企画から設計、開発、ネット上での公開まで全体の工程を一人で理解出来ているか
- ❑ スクールで学んだことがそのまま出来るようになるのではなく、自ら考え、調べ、失敗し、解決して挑戦することが出来るか

5. 良い成果物にするために ~制作にあたって避けたいこと~

✗ 自分で扱えない・説明できない技術を使う

- ❑ チャレンジはOK。でも、自分が書いたコードの説明ができない
- ❑ 「Qiitaにそう書いてありました」は、何も身についていない

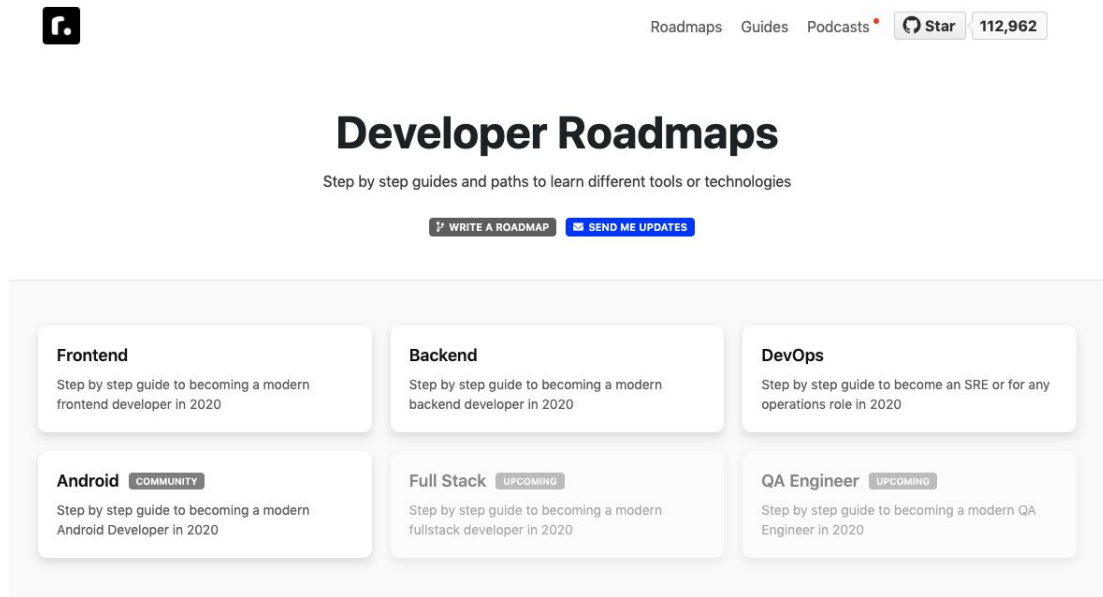
✗ 作ることが目的化し、技術習得・成長になっていない

- ❑ いろいろ詰め込んだ結果、まともに動いていないパターンがある
- ❑ テーマ < 何が出来るのか

✗ 他の人が見れない、見にくいコードになっている

- ❑ コードが汚くて、そもそも見る気になれない

[補足]良い成果物にするために ~扱える技術かどうか判断しよう~



自分で扱える技術かどうか。
自分で説明できる技術かどうか。

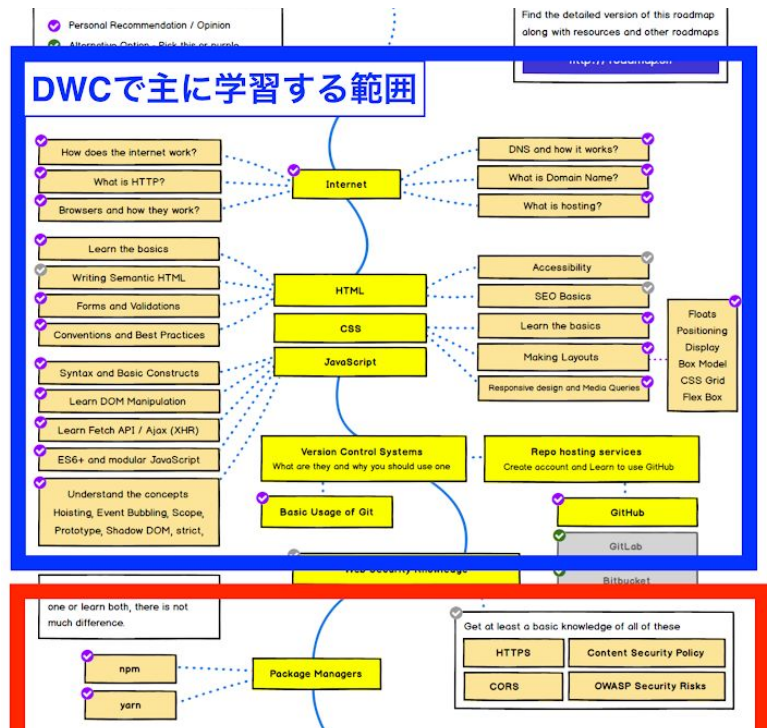
これらの参考となる資料として
”Developer Roadmaps”があります。

”Developer Roadmaps”は
エンジニアとしてのスキルを身につける
順番を上から順番にまとめた資料ですが
今、自分がどこまで理解しているのかを
把握するのにも役立ちます。

皆さんの自身の現在位置を確認し
今できる範囲のことを認識した上で
ポートフォリオ制作に臨みましょう。

出典: <https://roadmap.sh/>

[補足]良い成果物にするために ~扱える技術かどうか判断しよう~



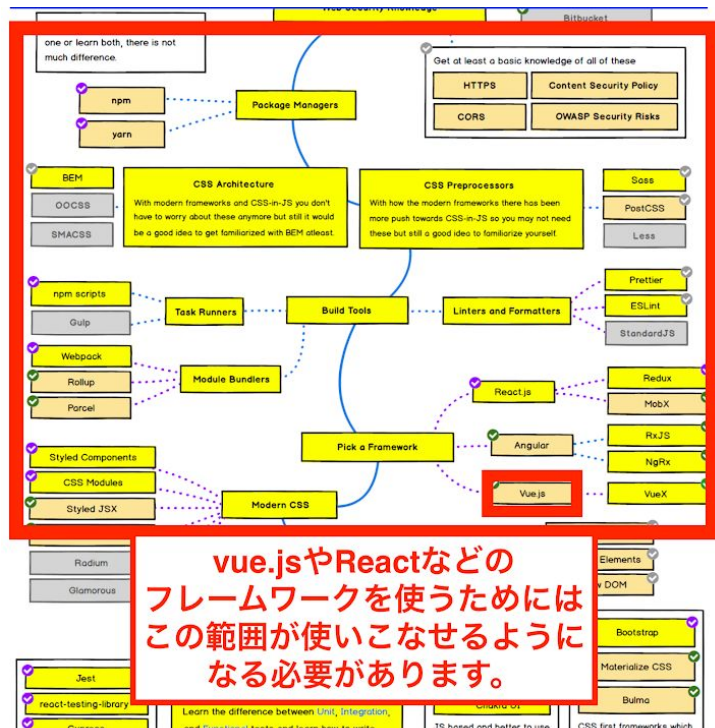
フロントエンドの技術について

参考: Developer Roadmaps > Frontend

今までみなさんは
HTML, CSSを中心に学びつつ
バージョン管理(Git)の使い方などを
習得してきました。

例えば、これからゼロベースで
vue.jsやReact.jsを学ぼうとした時に
追加で何を習得する必要があるのでしょうか？

[補足]良い成果物にするために ~扱える技術かどうか判断しよう~



vue.jsやReact.jsを学ぼうとした時に
前提として下記の知識が必要です。

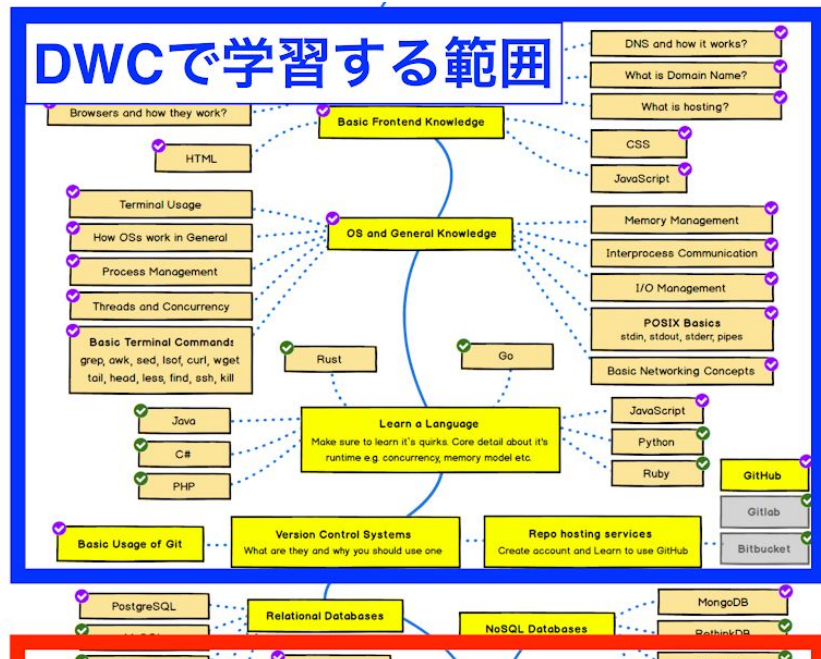
- Package Manager
- CSSの応用知識
- Build Tools

一口にvueやReactと言っても、必要な知識はそれ
だけではありません。

制作よりも学習に大きく時間を取られ
**結果、転職活動に必要なポートフォリオが
完成しない危険性があります。**

**これらの知識に興味があれば
ポートフォリオ制作後に学習**する
ようにしましょう。

[補足]良い成果物にするために ~扱える技術かどうか判断しよう~



バックエンドの技術について

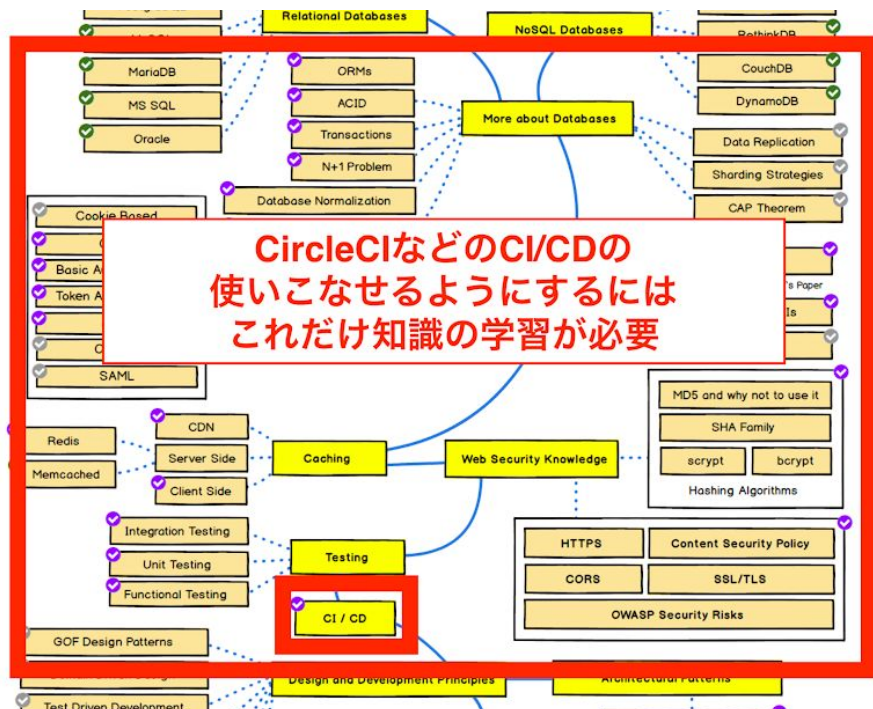
参考: Developer Roadmaps > Backend

フロントエンドの知識に加え、
Ruby on Railsでそれらを活用した
アプリケーションを製作
それに伴うターミナルでのコマンド操作を
習得してきました。

ポートフォリオ制作にあたり
追加でAWSのコンテンツを
キャッチアップしていただきますが
例えば、これに加えてゼロベースで
CircleCIやDockerを学ぼうとした時に
必要な学習量はどのくらいでしょうか？

出典: <https://roadmap.sh/>

[補足]良い成果物にするために ~扱える技術かどうか判断しよう~



例えば、CircleCIを学ぼうとした時に

前提として下記の知識が必要です。

- ・データベースに関するより深い理解
- ・API
- ・キャッシュ
- ・Webセキュリティ
- ・テスト

繰り返しになりますが、フロントエンド同様
制作よりも学習に大きく時間を取られ
**結果、転職活動に必要なポートフォリオが
完成しない危険性があります。**

※ Dockerについてはこの画像より下層にあり
必要とされる知識もより多いです。
気になる方は、配布された
出典元のサイトをご覧ください。

出典: <https://roadmap.sh/>

[補足]良い成果物にするために ~扱える技術かどうか判断しよう~

エンジニアとして働く上で「学び続ける」というスタンスは必須です。

しかし、「全く新しいものを学ぶ時期」と「アプリケーションを作る時期」を一緒にしすぎると、結果どちらも中途半端に終わってしまいます。

自分がどこまで理解しているかを考えず
いきなり背伸びして、新しい技術に飛びついても
結局、細部まで理解できず、基礎に立ち返ることになります。

まず「自分は今、どこまでのスキルを身に着けたか？」を的確に把握し
「次、どのスキルを身につけるべきか？」をしっかり判断することが重要です。
地に足の付いた学習をすると、中長期的には
頼られるエンジニアになることができます。(弊社エンジニア談)

背伸びをしすぎず、「どこまでなら背伸びできるか」を念頭に置いて、制作を行いましょう！

6. 企業はPFのどこをみるか

6. 企業はPFのどこをみるか

企業は、アプリケーションやソースコードから、以下を判断します。

「エンジニアとしてのスタンス」「一緒に開発に取り組めるエンジニアか？」

あなたが

- システム利用者（エンドユーザ）を考えているエンジニア
- チームメンバーに配慮できるエンジニア
- 今後の保守運用を考慮しているエンジニア

であることを証明するためにも、

次に説明する**3つの対策**をしましょう！

6.企業はPFのどこをみるか ~対策①~

対策①：エラーにより動作しない箇所をなくしましょう！

- ❑ システムは、作って終わりではありません。稼働後が本番です。
要は、「動いてナンボ」なのがシステム開発です。
「この人はリリース後の運用を考えられるエンジニアか？」
「この人はエンドユーザのことを考えているエンジニアか？」
を企業は見ています。
バグばかりのアプリケーションでは、その姿勢が伝わりません。
- ❑ テスト期間をしっかり考慮に入れ、バグを潰しましょう！

6. 企業はPFのどこをみるか ~対策②~

対策②：コメントを書きましょう！

- ❑ ソースコード内のコメントは、実際の挙動に影響がないため、ついつい記述が疎かになりがちです。
しかし、コードは自分以外がメンテナンスすることにも非常に多いです。
コードだけでは伝えづらい「なぜこの処理を書いたのか」という注釈が必要になります。
- ❑ 「コメント 重要性」「コメント 粒度」で検索してみましょう！

6. 企業はPFのどこをみるか ~対策③~

対策③：リーダブルコードを意識しましょう！

- ❑ リーダブルコードとは
 - ❑ 読みやすいコードのこと
- ❑ インデントや変数・関数名の分かりやすさ、DRYであるかどうか...
どれくらい意識できているでしょうか？
コメントと同様、いかに他者に配慮したコードを書いているかを企業はチェックします。
- ❑ 「リーダブルコード 要約」で必ず検索してみましょう！

6. 企業はPFのどこをみるか ~技術的観点以外~

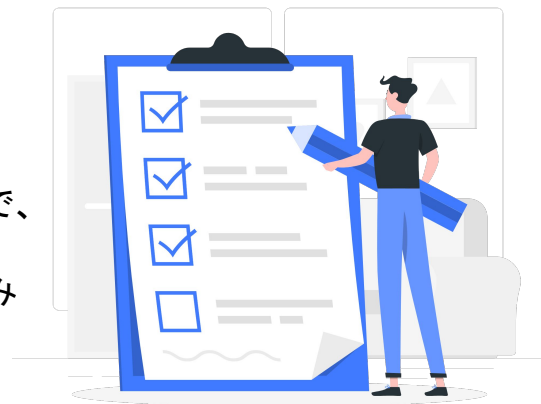
PFをみる企業は技術的観点はもちろんですが

- PFのテーマや取り組みや、こだわりによってパーソナリティなどをみています。

そのため、技術にこだわる以上に

使い手をイメージした作り込みができていますか

- エラーがないこと、テストデータでないこと、レイアウトが使いやすいかも重要なので、技術によりすぎて不完全な成果物になってしまうより
「技術がなくても誰にでも意識し取り組めば出来る点」に重きを置く事も意識
してください。





**こんなに学習に打ち込める期間は
もうないかもしれません。**

**意味のある1ヶ月間にできるかは
あなた次第です！**

あなたはどんなPFを作りますか？