

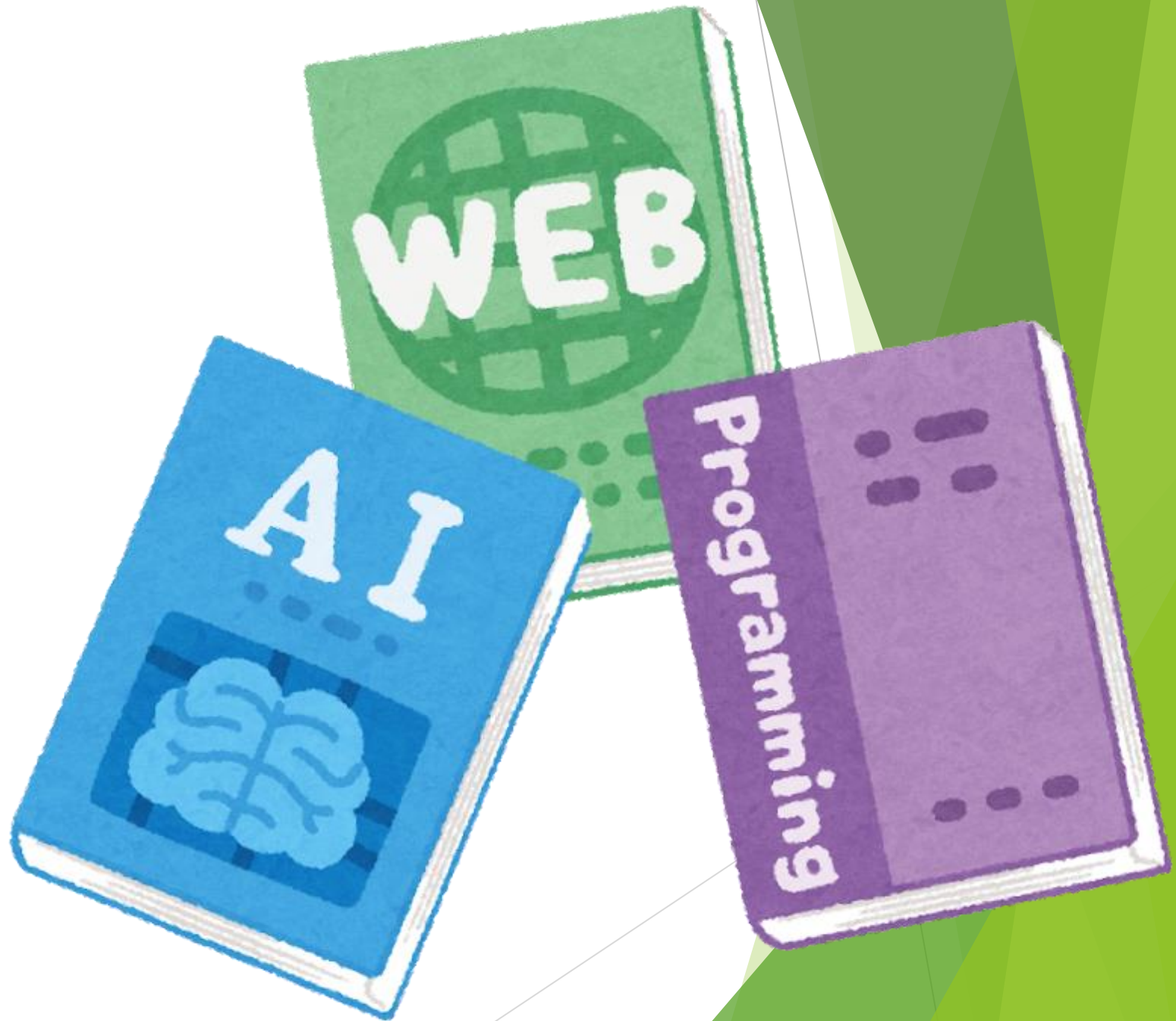
# 自己紹介プレゼンテーション

～プログラミング経験、どんなエンジニアになりたいか～

※タイトル含め全17ページ

# 目次

- ▶ 自己紹介
- ▶ プログラミング経験
  - ▶ 制作したアプリ
    - ▶ アプリの詳細
  - ▶ まとめ(プログラミング経験)
- ▶ どんなエンジニアになりたいか
  - ▶ そのために考えていること



# 自己紹介

# 自己紹介

## ▶ 名前

▶ ○○

## ▶ 出身地

▶ ○○

## ▶ 出身学校

▶ ○○

▶ ○○

## ▶ 趣味

▶ 自作パソコン,ゲーム,プログラミング,ホテル巡り(格安),アロマ



# プログラミング経験

# プログラミング経験

## ▶ Java

- ▶ 使用歴2年
- ▶ コーディング自体が一番得意
- ▶ Java初学者参考書が一通りわかるレベル、フレームワークを学んでいきたい！



## ▶ C#

- ▶ 使用歴2年
- ▶ デスクトップアプリに使用(.NET)
- ▶ 調べたら機能は追加できるレベル、Visual Studioの仕様理解を頑張りたい！



## ▶ Python

- ▶ 使用歴0.5年
- ▶ Web開発に使用(Flask)(Streamlit)
- ▶ 基礎文法と簡単なアプリ作成ができるレベル、ライブラリを覚えたい！



- ▶ GitHub
  - ▶ <https://github.com/imunkonow/>
- ▶ ブログ(ポートフォリオ掲載用)
  - ▶ <https://imunkonow.wordpress.com/>

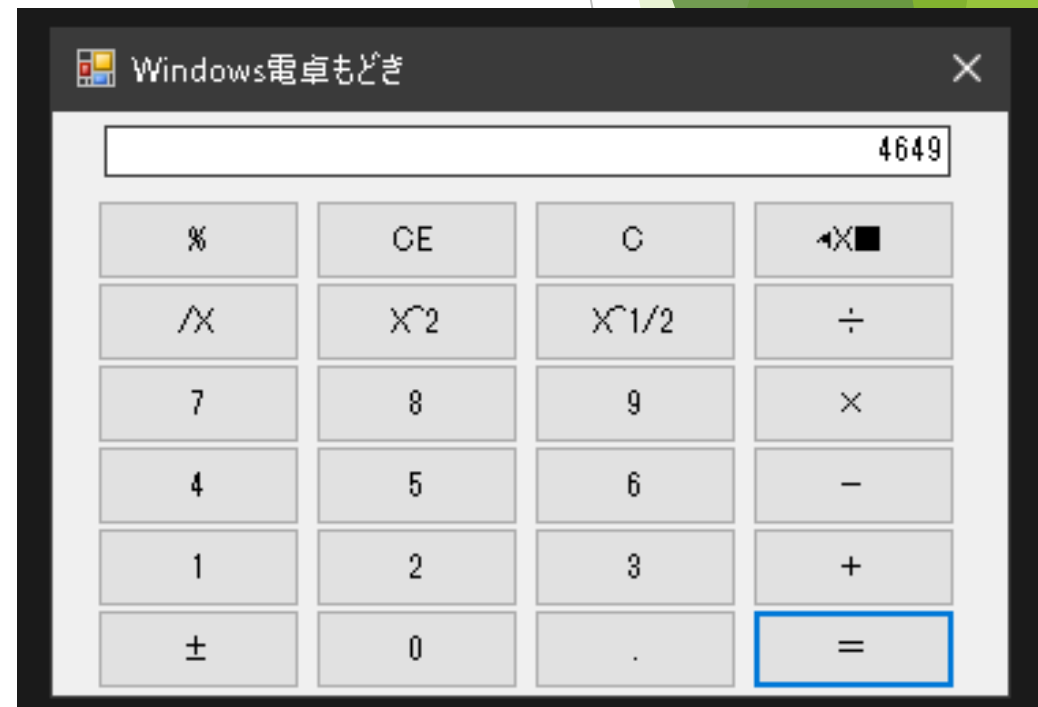


# プログラミング経験

## -電卓(デスクトップアプリ)-

授業「何でもいいからWindowsの標準電卓を再現せよ」

- ▶ 使用技術
  - ▶ .NET(C#)
- ▶ 技術選定の理由
  - ▶ ドラッグアンドドロップで直感的に開発ができる。
  - ▶ C言語,Javaを学んでいたことから、学習コストが低いため。
  - ▶ JavaのGUIライブラリのサポートが終了しているため。
- ▶ 学んだこと
  - ▶ 同じ処理の多さに苦労したことから、メソッドの使い方、オブジェクト指向の大切さを学んだ。





# プログラミング経験

## -ストップウォッチ(デスクトップアプリ)- 1/2

授業「何でもいいからアナログストップウォッチを製作せよ」

### ▶ 悩んだ、苦労した点

- ▶ “アナログストップウォッチ C#” と検索しても、  
デジタルストップウォッチしか出てこない(情報が少ない)

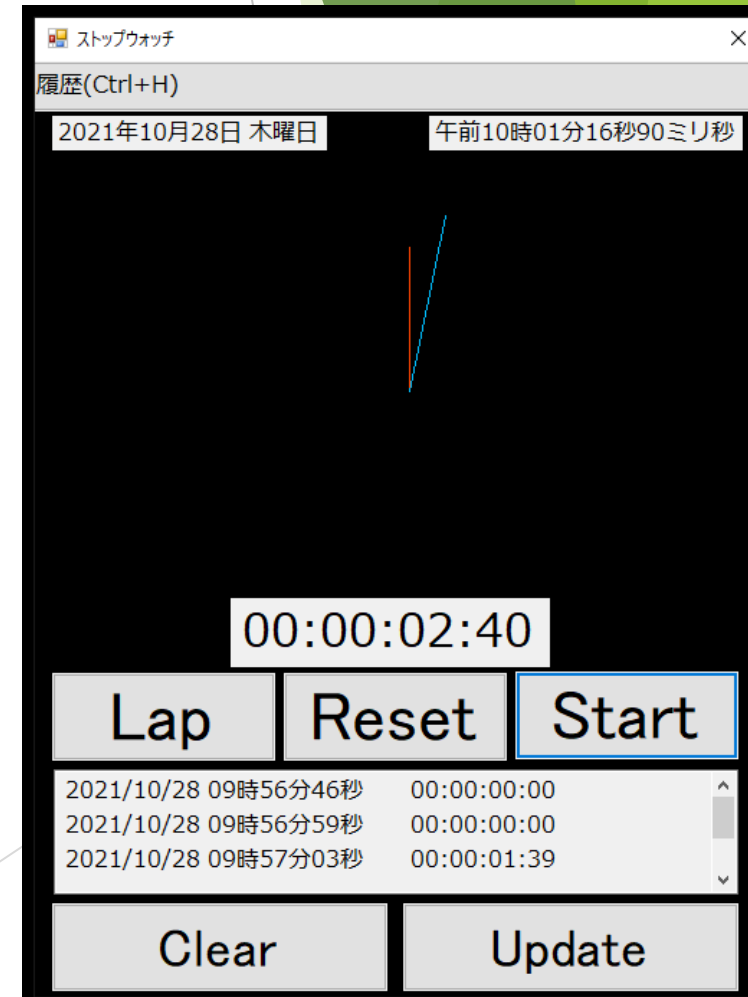


### ▶ どうやって乗り越えたか

- ▶ “アナログストップウォッチ C#”としてではなく、  
“デジタルストップウォッチ C#” “アナログ時計 C#”  
と課題を細分化することで、組み合わせで解決した。

### ▶ 学んだこと

- ▶ オブジェクト指向の考え方、課題を細分化することを学んだ。



# プログラミング経験

## -ストップウォッチ(デスクトップアプリ)- 2/2

### 仕様

- ▶ 記録を保存。アプリを閉じてでもOK
  - ▶ .txt 形式で保存
- ▶ アナログ&デジタルに対応
- ▶ ショートカットキーに対応

**仕様書**

The screenshot shows the application interface with several windows and annotations:

- history.txt - メモ帳 (Notepad):** Displays a list of saved records with timestamps and durations.
 

2021/10/28 09時56分46秒	00:00:00:00
2021/10/28 09時56分59秒	00:00:00:00
2021/10/28 09時57分03秒	00:00:01:39
- ストップウォッチ (Stopwatch):** The main application window.
  - 履歴 (Ctrl+H):** A sub-window showing the current date and time: 2021年10月28日 木曜日, 午前10時01分16秒90ミリ秒.
  - Buttons:** Lap, Reset, Start, Clear, Update.
  - Annotations:**
    - 計測値を1周60000ミリ秒で表示 (Display measurement value in 60000 milliseconds per lap)
    - 経過時間の分は任意 (Lap time in minutes is optional)
    - 秒 (Seconds)
    - 分 (Minutes)
    - 計測値をミリ秒表示 (Display measurement value in milliseconds)
    - 計測停止時は計測を開始 (Start measurement when stopped)
    - 計測中は計測を停止 (Stop measurement during measurement)
    - アプリケーションが終了しても履歴は消えない(ファイル経由) (History is not deleted even after application ends via file)
    - 保存した計測値をリスト表示 (Display saved measurement values in list)
    - なんらかの形で更新された場合に更新して表示 (Update and display when updated in any way)
    - 保存していた履歴をすべてクリアする (Clear all saved history)
- 履歴 (H):** A window showing a list of saved records, similar to the Notepad window.
  - Annotations:**
    - アプリケーションが終了しても履歴は消えない(ファイル経由) (History is not deleted even after application ends via file)
    - 保存した計測値をリスト表示 (Display saved measurement values in list)
    - Update button
    - Clear button
    - なんらかの形で更新された場合に更新して表示 (Update and display when updated in any way)
    - 保存していた履歴をすべてクリアする (Clear all saved history)

# プログラミング経験

## -ToDoリスト(Webアプリ)- 1/2

趣味 「ログイン機能やCRUDを用いたWebアプリを作りたい」

### ▶ 使用技術

- ▶ Flask(Python)
- ▶ BootStrap(CSS)
- ▶ HTML
- ▶ デプロイ(クラウドサービスへのアップロード)

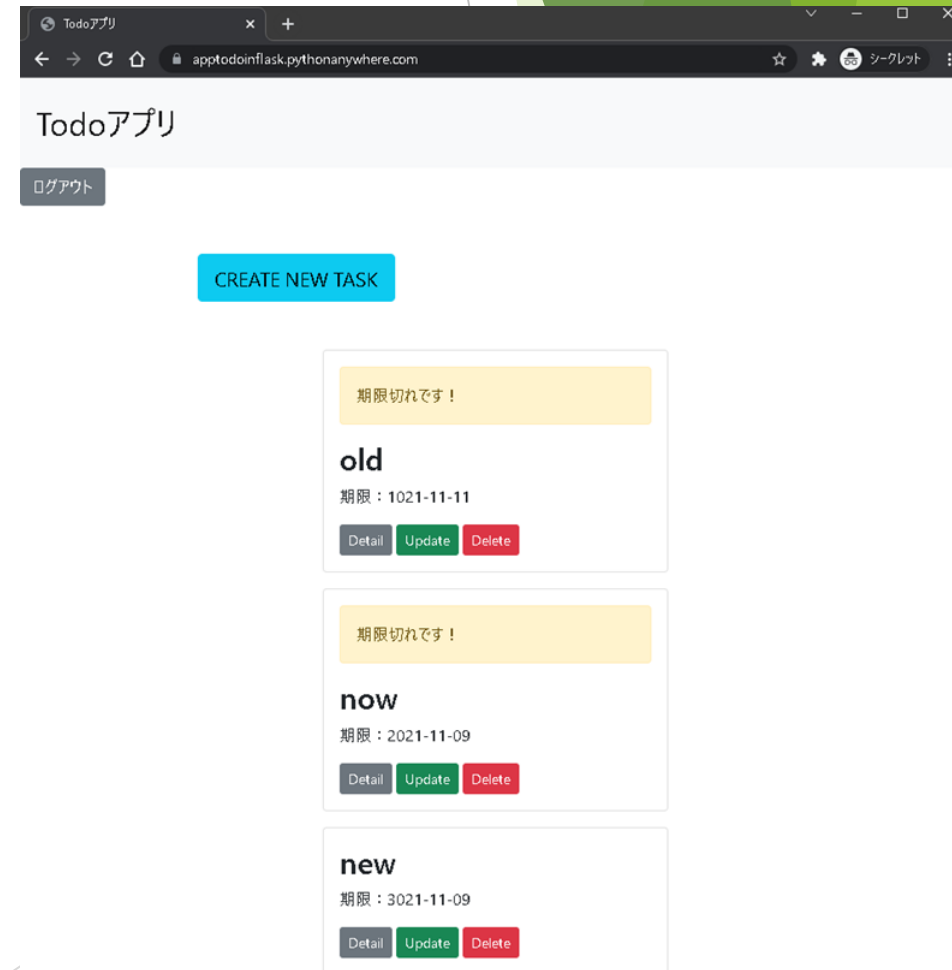
### ▶ 技術選定の理由

- ▶ Python及びWebフレームワークの学習目的。
- ▶ 個人開発&個人レベルなので、大規模向けのDjangoではなく、小規模向けのFlaskを選定。
- ▶ デザインに疎いため、定番であるBootStrapを採用。

### ▶ 学んだこと

- ▶ 初めてのWebアプリ製作だったので

全部



# プログラミング経験

## -ToDoリスト(Webアプリ)- 2/2

### 仕様

- ▶ ログイン、ユーザー登録に対応
- ▶ データベース機能対応(SQLite3)
  - ▶ CRUD(作成、読み込み、更新、削除)
- ▶ 期限切れ通知に対応

### これから追加したい機能

- ▶ ユーザーごとのToDoデータベースを作成
- ▶ UI(ユーザーインターフェース) の改善




# プログラミング経験 -スパムメール(Webアプリ)-

- ▶ 開発経緯
  - ▶ 授業  
「チームでPythonを使用したWebアプリ製作」
- ▶ 私のポジション
  - ▶ **リーダー**(他メンバー全員Python知らない)
- ▶ ポジションの苦労
  - ▶ チームをまとめることが**大変**
    - 暇にさせない  
(遊んでて良いんだという空気にさせない)
    - できるだけリソースを無駄にしないように**工夫**  
(アイデアや意見、簡単な部分を任せる等)

Document x +

← → ↺ 🏠 [spammailinpython.pythonanywhere.com](http://spammailinpython.pythonanywhere.com) 🔍 ☆ ⚙️ 🛡️ シークレット



# 畑良佑専用スパムメール

To :

Subject :

Message :

Times :

[Let's Spam](#)



# プログラミング経験

## -まとめ-

### ▶ 言語

- ▶ Java C# Python (他にも C言語,Ruby,JavaScript,PHP,SQL,Linux,AWSなど)

### ▶ 学んだこと

- ▶ メソッドの使い方
- ▶ オブジェクト指向
- ▶ 課題の細分化
- ▶ チームをまとめる大変さ



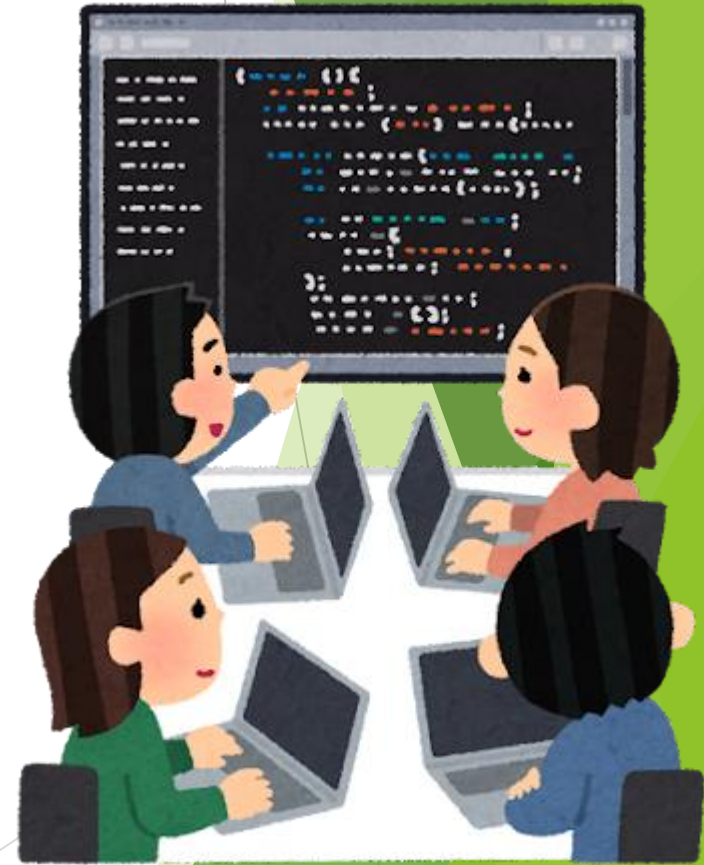
どんなエンジニアになりたいか



# どんなエンジニアになりたいか

## 上流工程から下流工程まで何でもできる フルスタックエンジニアになりたい

- ▶ なぜ上流工程に携わりたいのか
  - ▶ 学校ではプログラミングだけでなく、要件定義、機能設計などの上流工程を体験したことで、**IT全体の仕事に興味をもった**から。
  - ▶ **エンドユーザーとの声の近さ、人との関わりを大切にしたい**ため。
- ▶ 上流工程に携わるために必要だと考えていること
  - ▶ 下流工程での**経験が必要**だと考えている
    - ▶ 設計段階や他の人に使ってもらうことを意識した開発が必要





# どんなエンジニアになりたいか

## -そのために考えていること-

- ▶ 具体的にこれから何をするのか(考えていること)
  1. ネイティブアプリの開発、アプリストアへのデプロイ
    - ▶ PC,WEBと経験したので、モバイルも経験し、公開したい
  2. CMSを使わないWebサイトの製作
    - ▶ Wordpressなどのツールを使わずに1からの製作をし、HTML,CSS,JavaScriptへの理解を深める。
- ▶ 入社後に考えていること
  - ▶ 少しでも早く上流工程に携わるために、下流工程を一人前にする
    - ▶ 休日の勉強や、毎日の情報収集を欠かさない。
  - ▶ フロントエンド,バックエンド,アプリの全部に携わってみたい

