Lightning Talk

N Code Labo LT 大会 2024 年 9 月

自己紹介

- 名前：鎌田 一輝（かまた かずき）
- 生業：Web アプリケーションエンジニア
- よく使う技術：Angular(TypeScript), Django(Python)

今回のテーマは「非同期処理」

非同期処理？なにそれおいしいの？

ノリだけ掴む async-await

↓こういうコードが何をやっているか、なんとなくわかるようになる(かも)

```
async function displayUsers() {
  const users = await fetchUsers();
  console.log(users);
}
```

同期処理・非同期処理ってなに？

同期処理

- コードを１行ずつ実行 → 完了したら次の行に移る(順次実行)
- 普通にコードを書いたらこうなる

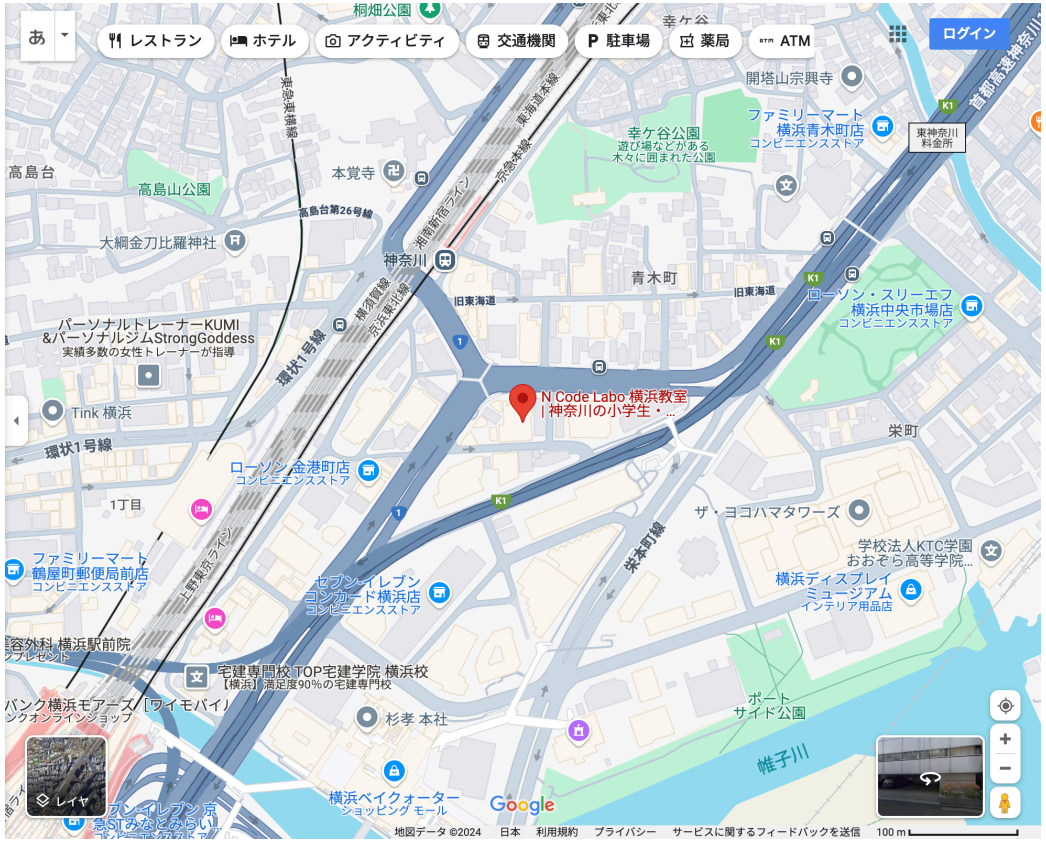
非同期処理

- **実行した処理の完了を待たずに** 次の行に移る
- 専用の書き方をする必要がある

非同期処理の例

Google Map

地図情報を非同期に取得するため、
マップ移動、ランドマークの選択などが
ブロックされない



async

関数を非同期にするキーワード

- 時間がかかる処理を非同期で実行することが多い
  - HTTP リクエスト、ファイル読み書き、DB クエリ発行 etc...
- 待ち時間で他の処理を実行できて効率が良い！

Promise(など)

非同期処理の結果を表すオブジェクト

- 主に、非同期関数が戻り値として返す
- 「実行中(pending)」「成功(fulfilled)」「失敗(rejected)」のいずれかの状態を持つ
  - 「成功」の場合は、結果を取得できる
  - 「失敗」の場合は、例外が投げられる

await

非同期処理の完了を待機するキーワード

- 同期処理っぽく処理の完了を待てる
- 異なる点は...
  - どの処理を並行で実行するか、待つか選べる
  - **いい感じの待ち方** をしてくれる
    (再描画や各種イベントはブロックしない)

非同期処理を実装してみよう

サンプルコード

```
function fetchUsers() {
  return new Promise((resolve, reject) => {
    fetch("https://ip:port/api/v1/users")
      .then((response) => {
        if (!response.ok) {
          throw new Error("Failed to fetch users");
        }
        return response.json();
      })
      .catch((error) => {
        console.error(error);
      });
  });
}

async function displayUsers() {
  const users = await fetchUsers();
  console.log(users);
}

displayUsers();
```

サンプルコード(デモ用)

```
function fetchUsers() {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      resolve(["Tetsuya", "Akihiro", "Tatsuo", "Masamune"]);
    }, 2000);
  });
}

async function displayUsers() {
  const users = await fetchUsers();
  console.log(users);
}

displayUsers();
```

非同期処理はもう怖くない！！！