

# システム開発 ストレートフラッシュ 共有ノート (Docker編)

---

バックエンド開発に必要な知識や情報を共有するファイルです。

## 目次

---

1. [Dockerとは](#)
2. [docker-composeとは](#)

## Dockerとは

---

Dockerはコンテナ型の仮想環境を作成・配布・実行するためのプラットフォームです。

開発においては環境の差異をDockerが吸収してくれるので、開発メンバーが使用するPCのOSが違うといった場面でもDockerを使うことでみんなが同じ環境で開発できます。(Windowsだとライブラリ入ったけどMacだと入らない！みたいな問題を解決してくれます)

## よく使うDockerコマンド

---

`docker images` --コンテナイメージ一覧を表示

`docker rmi IMAGE ID` --指定したコンテナイメージの削除

`docker start CONTAINER ID` --指定したコンテナの起動

`docker restart CONTAINER ID` --指定したコンテナの再起動

`docker stop CONTAINER ID` --指定したコンテナの削除

`docker rm CONTAINER ID` --指定したコンテナの削除

`docker logs CONTAINER ID` --指定したコンテナのログの出力( `-f` オプションをつけるとリアルタイムでログを出力できる )

`docker ps` --起動しているコンテナの情報の出力( `-a` オプションをつけると起動していないコンテナを含めて表示 )

`docker exec -it CONTAINER ID bash` --コンテナの中に入る

# Docker Composeとは

---

複数のコンテナを一括で管理できるツールです。

API用のコンテナとDB用のコンテナをまとめて管理したい場合などに使用します。

## 実装の流れ

---

- 1: サービスごとのDockerファイルを定義
- 2: アプリケーションを構成するサービスを docker-compose.yml ファイル内に定義
- 3: アプリケーションのビルド&起動

## よく使うDocker Composeコマンド

---

`docker-compose build` --アプリケーションのビルド

`docker-compose up -d` --アプリケーションの起動( `-d` オプションをつけるとバックグラウンドで処理されます)

`docker-compose down` --アプリケーションの停止

`docker-compose restart` --アプリケーションの再起動

`docker-compose logs -f` --アプリケーションのログ出力( `-f` オプションをつけるとリアルタイムでログを出力できる)