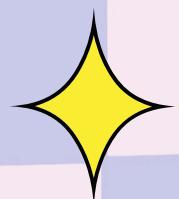
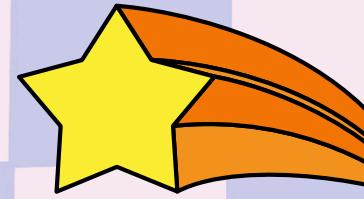
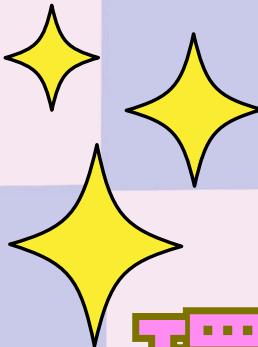
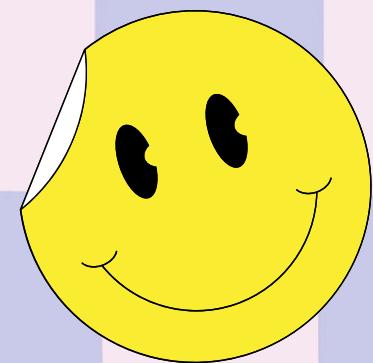
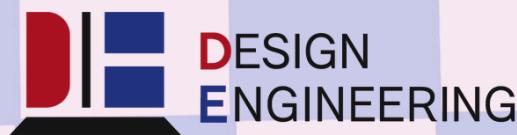
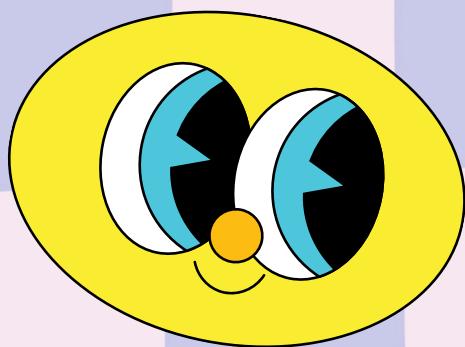


たった1つで
環境構築が秒速になる！

DOCKERの魔法



Dockerとは？

- Docker社が開発した、仮想環境を動かす仕組みのこと

- 仮想環境：一つのパソコンやサーバーの中で、
仮想的に構築した環境



- 船の上に「コンテナ」を載せているようなイメージ

- なぜコンテナなのか？ → 違うところに移すことが容易だから
 - 例えば、PHPという違う言語を入れたいと思ったら、「PHP」というコンテナを降ろして
「Ruby」というコンテナを積めば同じ環境で別の言語を使える

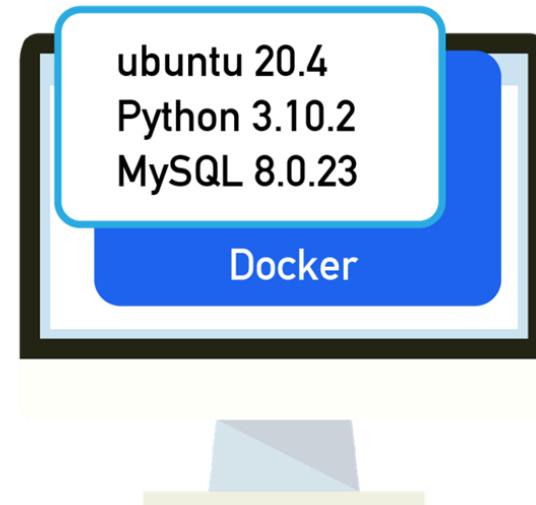
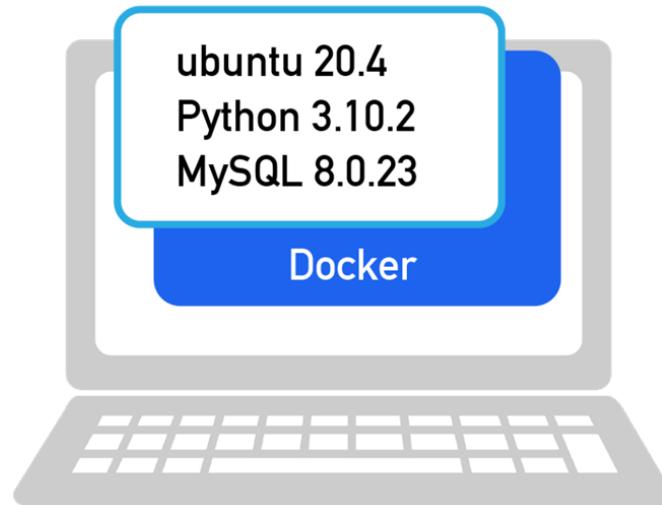
ポイント①：開発環境の足並みを揃えられる

- **いつでも、どこでも、同じ環境を再現できる**
 - インストールなど苦労して行なった環境構築を、もう一度別のPCでやるのはとても面倒…
 - Docker が入ってさえいれば Mac↔Windows も同じ環境が再現可能
 - **それぞれのPCの状態が違っても、メンバー全員が同じ仮想環境下で開発できる**
 - その環境は [Docker Hub](#) から取ってくるイメージを使う

開発環境の足並みを揃えられる (2)

OSの上に Docker を挟んでいるため、アプリケーションの構築環境は Mac でも Windows でも Linux でも全く同じ環境が担保され、正しい動作も保証されている

→ 「Macでは動いたけどWindowsでは動かない...」のような依存が起きず、どのコンピューター上でも同じ環境を作り出せる



ポイント②：環境構築の速さ

例：Python開発環境を用意する場合

Docker あり

- docker-compose.yml を用意し、バージョンやライブラリを入力
- コマンド一つで起動
(docker compose up)

Docker なし

- Pythonをインストール
- バージョンを合わせる（例：3.9.0）
- pipを設定
- 必要なライブラリを1つずつインストール
- DB（MySQL）をインストールして設定
- OSやパッケージの違いでエラー → 調整が必要

ポイント③：再利用・バージョン管理・配布が容易

- OSの指定、ミドルウェアのインストールや環境設定がコード化されている
 - コードの一部を書き換えての再利用が簡単
 - テキストファイルなのでGitで管理ができる
 - 軽量なテキストファイルだから他のメンバーに配布もしやすい。
→ コマンドを実行するだけで、共有された環境構築が簡単にできる

ポイント④：軽量かつ高速

- DockerコンテナはホストOSのカーネルを共有して動く
 - 従来のように「OSを丸ごと起動」する必要がない
- 従来の仮想マシン (Virtual Machine)
 - OSごとにインストール＆起動させる必要がある
 - リソース (CPU・メモリ) も多く消費
- Dockerコンテナ
 - ホストOS上で直接アプリを動かす仕組み
 - 起動は数秒、リソースも効率的に利用可能

Dockerを使う流れ

既存の環境を使う場合

Docker Hub (倉庫) → イメージ (設計図) → コンテナ (実物の環境)



自分の作った環境を渡す場合

Docker Hub (倉庫への共有) ← イメージ (設計図化) ← コンテナ (動いている環境)

Docker Hubとは？

- Docker社が提供しているDockerイメージ（コンテナ）の共有サービス
 - <https://hub.docker.com/>
 - さまざまなDockerイメージが公開されている
 - Docker Hubから任意のイメージを取得後、コンテナ化して様々なアプリケーションを利用可能
 - イメージの管理や共有が容易になり、開発プロセスの効率化とコラボレーションの促進が実現できる

Dockerイメージとは？

- コンテナを実行するために必要なものをひとまとめにしたパッケージ
 - コンテナで実行するアプリケーションとその依存関係、環境変数、ファイルシステムなどを含む
 - `Dockerfile`というテキストファイルを元に作成
 - `Dockerfile`は[Docker Hub](#)を使うことで、様々なイメージを組み合わせて作ることが可能
 - 持ち運びできるから、どこでも同じコンテナを作れる
- オブジェクト指向で言うところの「クラス」
 - どのような状態や振る舞いを持つべきかを定義する「設計図」のようなもの

Dockerコンテナとは？

- お互い影響や干渉をせず、隔離された実行環境を提供する技術
 - Dockerイメージが設計図だとすると、コンテナはその設計図から作成された実物
 - オブジェクト指向で言うところの「インスタンス」
- コンテナの特徴
 - 軽量
 - 高速
 - 隔離されている（他のコンテナやホスト環境に干渉しない）
 - 一時的に使える（使い捨てが可能）

参考資料

- クライン【KLeIn】 [Dockerを超ざっくり解説！【何となく理解するIT】](#)
- Pythonプログラミング VTuber サプー [【Docker入門】初心者向け！Dockerの基本を学んでコンテナ型の仮想環境を作ろう！](#)
- もんしょー|ITエンジニア個別指導【ITラボ】 [【入門者も理解】Dockerとはなにか？【専門用語少なめ】](#)
- エンジニア転職チャンネル【RUNTEQ公式】 [【初めてのDocker】コンテナってそもそも何なの？](#)
- @TakanoriVega [Docker\(コンテナ\)使って何が嬉しいんだっけ？ \(Qiita\)](#)
- カゴヤのサーバー研究室 [【入門】Docker Hubとは？概要と仕組み、基本的な使い方を解説](#)
- WebNAUT [はじめてのDocker～公式イメージで簡単にWordPressサイトをつくる方法～](#)



LINE公式アカウントにて
最新情報配信中！



無料で自由に使える
学習＆カフェスペース開放中！

変化を楽しみ、自分らしく未来へ。

デザインエンジニアリングは、
挑戦するエンジニアの一歩を応援する会社です。

“好き”や“ワクワク”をそのままキャリアに変え、
自分の可能性を信じて前へ進む人には、無限のチャンスが広がっています。

失敗も学びに変え、仲間と共に笑い、共に成長しながら、
毎日が少しずつ楽しくなる未来へ。
未経験でも大丈夫。あなたの最初の一歩を、心からお待ちしています！

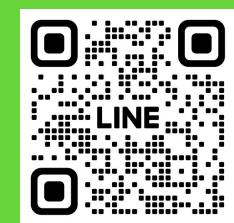


イベント・セミナー開催中！

カジュアル面談・エントリーは
こちらから！



URL: <https://design-engineering.jp/>



LINE ID:@749gaovb

Dockerとは？

- Docker社が開発した、仮想環境を動かす仕組みのこと

- 仮想環境：一つのパソコンやサーバーの中で、
仮想的に構築した環境



- 船の上に「コンテナ」を載せているようなイメージ

- なぜコンテナなのか？ → 違うところに移すことが容易だから
 - 例えば、PHPという違う言語を入れたいと思ったら、「PHP」というコンテナを降ろして
「Ruby」というコンテナを積めば同じ環境で別の言語を使える

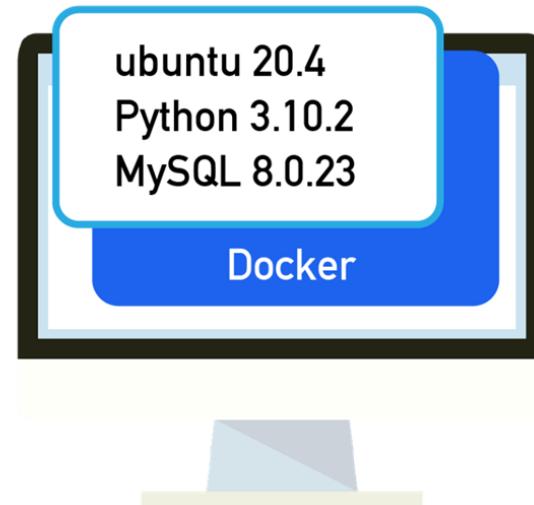
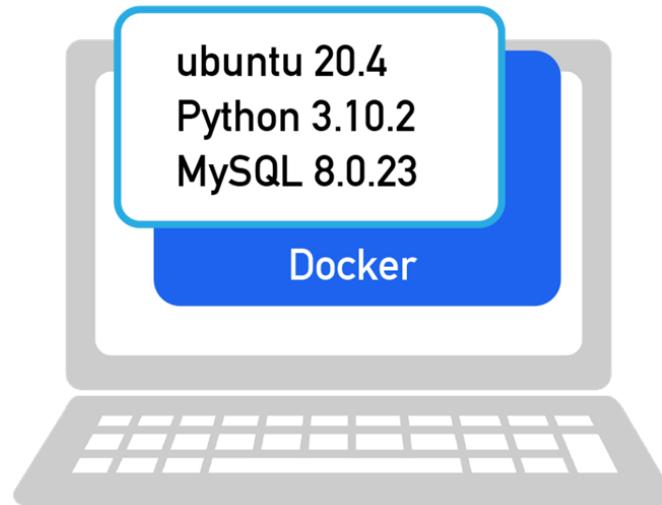
ポイント①：開発環境の足並みを揃えられる

- いつでも、どこでも、同じ環境を再現できる
 - インストールなど苦労して行なった環境構築を、もう一度別のPCでやるのはとても面倒…
 - Docker が入ってさえいれば Mac↔Windows も同じ環境が再現可能
 - それぞれのPCの状態が違っても、メンバー全員が同じ仮想環境下で開発できる
 - その環境は [Docker Hub](#) から取ってくるイメージを使う

開発環境の足並みを揃えられる (2)

OSの上に Docker を挟んでいるため、アプリケーションの構築環境は Mac でも Windows でも Linux でも全く同じ環境が担保され、正しい動作も保証されている

→ 「Macでは動いたけどWindowsでは動かない...」のような依存が起きず、どのコンピューター上でも同じ環境を作り出せる



ポイント②：環境構築の速さ

例：Python開発環境を用意する場合

Docker あり

- docker-compose.yml を用意し、バージョンやライブラリを入力
- コマンド一つで起動
(docker compose up)

Docker なし

- Pythonをインストール
- バージョンを合わせる（例：3.9.0）
- pipを設定
- 必要なライブラリを1つずつインストール
- DB（MySQL）をインストールして設定
- OSやパッケージの違いでエラー → 調整が必要

ポイント③：再利用・バージョン管理・配布が容易

- OSの指定、ミドルウェアのインストールや環境設定がコード化されている
 - コードの一部を書き換えての再利用が簡単
 - テキストファイルなのでGitで管理ができる
 - 軽量なテキストファイルだから他のメンバーに配布もしやすい。
→ コマンドを実行するだけで、共有された環境構築が簡単にできる

ポイント④：軽量かつ高速

- DockerコンテナはホストOSのカーネルを共有して動く
 - 従来のように「OSを丸ごと起動」する必要がない
- 従来の仮想マシン (Virtual Machine)
 - OSごとにインストール＆起動させる必要がある
 - リソース (CPU・メモリ) も多く消費
- Dockerコンテナ
 - ホストOS上で直接アプリを動かす仕組み
 - 起動は数秒、リソースも効率的に利用可能

Dockerを使う流れ

既存の環境を使う場合

Docker Hub (倉庫) → イメージ (設計図) → コンテナ (実物の環境)



自分の作った環境を渡す場合

Docker Hub (倉庫への共有) ← イメージ (設計図化) ← コンテナ (動いている環境)

Docker Hubとは？

- Docker社が提供しているDockerイメージ（コンテナ）の共有サービス
 - <https://hub.docker.com/>
 - さまざまなDockerイメージが公開されている
 - Docker Hubから任意のイメージを取得後、コンテナ化して様々なアプリケーションを利用可能
 - イメージの管理や共有が容易になり、開発プロセスの効率化とコラボレーションの促進が実現できる

Dockerイメージとは？

- コンテナを実行するために必要なものをひとまとめにしたパッケージ
 - コンテナで実行するアプリケーションとその依存関係、環境変数、ファイルシステムなどを含む
 - `Dockerfile`というテキストファイルを元に作成
 - `Dockerfile`は[Docker Hub](#)を使うことで、様々なイメージを組み合わせて作ることが可能
 - 持ち運びできるから、どこでも同じコンテナを作れる
- オブジェクト指向で言うところの「クラス」
 - どのような状態や振る舞いを持つべきかを定義する「設計図」のようなもの

Dockerコンテナとは？

- お互い影響や干渉をせず、隔離された実行環境を提供する技術
 - Dockerイメージが設計図だとすると、コンテナはその設計図から作成された実物
 - オブジェクト指向で言うところの「インスタンス」
- コンテナの特徴
 - 軽量
 - 高速
 - 隔離されている（他のコンテナやホスト環境に干渉しない）
 - 一時的に使える（使い捨てが可能）

参考資料

- クライン【KLeIn】 [Dockerを超ざっくり解説！【何となく理解するIT】](#)
- Pythonプログラミング VTuber サプー [【Docker入門】初心者向け！Dockerの基本を学んでコンテナ型の仮想環境を作ろう！](#)
- もんしょー|ITエンジニア個別指導【ITラボ】 [【入門者も理解】Dockerとはなにか？【専門用語少なめ】](#)
- エンジニア転職チャンネル【RUNTEQ公式】 [【初めてのDocker】コンテナってそもそも何なの？](#)
- @TakanoriVega [Docker\(コンテナ\)使って何が嬉しいんだっけ？ \(Qiita\)](#)
- カゴヤのサーバー研究室 [【入門】Docker Hubとは？概要と仕組み、基本的な使い方を解説](#)
- WebNAUT [はじめてのDocker～公式イメージで簡単にWordPressサイトをつくる方法～](#)



LINE公式アカウントにて
最新情報配信中！



無料で自由に使える
学習＆カフェスペース開放中！

変化を楽しみ、自分らしく未来へ。

デザインエンジニアリングは、
挑戦するエンジニアの一歩を応援する会社です。

“好き”や“ワクワク”をそのままキャリアに変え、
自分の可能性を信じて前へ進む人には、無限のチャンスが広がっています。

失敗も学びに変え、仲間と共に笑い、共に成長しながら、
毎日が少しずつ楽しくなる未来へ。
未経験でも大丈夫。あなたの最初の一歩を、心からお待ちしています！



イベント・セミナー開催中！

カジュアル面談・エントリーは
こちらから！



URL: <https://design-engineering.jp/>



LINE ID:@749gaovb