ChatGPT

画像のダウンロードと実行

自分でイメージを作成する前に、既存のイメージを探して使ってみましょう。

Web開発者にとって必要なツールといえば、当然のことながらWebサーバーです。

画像をダウンロードする方法を学びます。まずは任意のコマンドプロンプトを開いて、次のように入力します：

```
docker pull php
```

ここで出てくるキーワードは次のとおりです：

- `docker`：Dockerコマンドを実行することを示します。
- `pull`：Gitと同じようにイメージをダウンロードするためのコマンドです。
- `php`：ダウンロードしたいイメージの名前です。

さまざまなダウンロード可能なイメージは以下のサイトで確認できます：

https://hub.docker.com

デフォルトでは、PHPは `latest` というタグを使用します。これは最新バージョンをダウンロードするという意味です。しかし、特定のバージョンを指定することもできます。

```
docker pull php:8.2-apache
```

イメージ名の後の「:」は「タグ」と呼ばれ、より詳細なバージョンの指定が可能です。

所有しているイメージは次のコマンドで確認できます：

```
docker images
```

不要なイメージは次のようにして削除します：

```
docker rmi php:latest
```

これで、PHPとApacheサーバーを含むイメージが準備できましたが、これを使用するコンテナはまだありません。コンテナを作成するには以下のように記述します：

```
docker run --rm -p 8080:80 php:8.2-apache
```

新しい構文の説明

以下はコマンド内の新しいオプションの説明です：

- `run`：コンテナを起動することを示します。
- `--rm`：コンテナの停止時に自動的に削除されます。
- `-p`：コンテナがアクセス可能なポートを指定します。
- 最後に指定するのは、コンテナがコピーすべきイメージの名前です。

もし `pull` をせずに `run` を実行した場合でも、Dockerは自動的にイメージを取得してくれます。

ここで指定しているポート「8080:80」は、2つのポートを表しています。最初の「8080」は仮想マシン側のポートで、これを使ってアクセスします。2番目の「80」はコンテナ側のポートです。

つまり、`localhost:8080` にアクセスすると、コンテナ内のポート80にリダイレクトされるということです。現時点ではファイルが何もないため、403 Forbidden エラーになります。

サーバーの停止方法

サーバーがコマンドラインを占有している場合は、`Ctrl + C` で停止できます。それ以外の場合、次のコマンドを使用します：

```
# 現在動作中のコンテナを見る
docker ps
# コンテナを停止する
docker stop コンテナ名
# コンテナを再起動する
docker start コンテナ名
# コンテナを削除する
docker container rm コンテナ名
```

`docker ps` に `-a` オプションを付けることで、停止中のコンテナも含めて全て表示することができます。

自分のファイルを使う方法

ここまででも十分ですが、やはり自分のファイルをコンテナ内で使いたいところです。

方法は3つあります：

1. **自分自身のイメージを作る**：本番環境に最適ですが、開発中は変更のたびにイメージを再作成する必要があり不便です。

2. **ボリュームを作成する**：仮想マシン内にフォルダを作り、複数のコンテナで共有可能です。ただし、こちらも更新が手間です。

3. **ディレクトリをマウントする（バインドマウント）**：自分のPC上のディレクトリをコンテナのフォルダに結び付けることで、即時反映が可能です。

ディレクトリのマウント

コマンドラインで、マウントしたいディレクトリ（ここでは　"01-dev"）に移動し、以下のコマンドを実行します：

```
docker run -d -p 8080:80 -v ${PWD}:/var/www/html --name exempleWeb php:8.2-apache
```

ここでの新しい構文は次のとおりです：

- `-d`：ターミナルをコンテナに縛らずバックグラウンドで実行します。
- `-v ${PWD}:/var/www/html`：現在のフォルダをコンテナ内の `/var/www/html` にマウントします。
- Windows CMD を使っている場合は `${PWD}` の代わりに `%cd%` を使用してください。
- `--name`：コンテナに任意の名前を付けます。省略すると自動で名前が付きます。

これで `localhost:8080` にアクセスすれば、自分のサイトが表示されるようになります。

PHP拡張のインストール

PHPの標準インストールには多くの拡張が含まれていますが、すべてではありません。必要な拡張を追加するには、コンテナのコマンドラインに接続する必要があります。

```
docker exec -it コンテナ名 bash
# または bin/bash
```

- `exec`：コンテナ内でコマンドを実行します。
- `-it`：コンテナにインタラクティブに接続します。
- `bash`：シェルを起動します。

接続後は、次のようにしてアップデートを行うと良い習慣です：

```
apt update && apt upgrade
```

アップデートが完了したら、必要な拡張を追加します：

```
# データベース接続用：
docker-php-ext-install pdo_mysql

# PHPで画像を生成するため：
```

```
apt install -y libfreetype6-dev
docker-php-ext-configure gd --with-freetype=/usr/include/freetype2/
docker-php-ext-install gd
```

次のステップでは、独自のイメージを作成していきます。