

・ フレームワークを使う意義

データの入出力やエラー処理、画面遷移などの基本的な仕組みを提供する。これは、いつも同じようなロジックを記述する必要がなくなるという「開發生産性」のみではなく、「誰が書いても」均質なコードを保証できる。

標準化ドキュメントを工数をかけてメンテナンスする必要もなく、チーム同士でモジュール(部品)間のインターフェンスを確認し合う必要もない。「フレームワークが提供するルール」に従うのみで良い。

・ S t r u t s (正式名称 : Apache Struts)

■特徴

Javaで最も有名なフレームワーク。Apache ソフトウェア財団によって開発されている Web アプリケーションフレームワーク。2000 年に初期バージョンが開発された。

MVC モデルを採用している。

画面の変更であれば、View である画面(JSP)の修正だけで済むという開発効率向上のメリットがある。JSP では「<%= %>」で囲まれるコードを埋め込む必要があるが、Struts では、独自のタグライブラリにより Java 言語のコードを混在させる必要がない。

チームで Web アプリケーションを設計すると、個々のエンジニアの設計方法や、考えが違うため、実装方法に多少の差異がある Web アプリケーションが構築されてしまいがちだが、Struts はその問題を解消することが可能である。

カスタムタグの利用や、Action クラスと struts-config.xml での画面遷移の管理、バリデータ(入力チェック)機能の提供が挙げられる。

・ J S F (Java Server Faces)

■特徴

JSF は 2004 年に開発された Java ベースの Web アプリケーションフレームワークで、JavaEE の仕様にも採用されている Java の標準フレームワーク。

MVC モデルを採用したフレームワークだが、Struts のように URL によってアクションが決まる「アクションベースフレームワーク」ではなく、表示する画面に対応するサーバーサイドのクラスにより、処理が行われる「コンポーネントベースフレームワーク」という違いがある。

また、MVC モデルの View(表示)には JSP で実装されるケースが多いが、JSF は「XHTML」という XML 形式の HTML で実装される。JSP はサーブレットの処理なしでは正しく表示されないが、XHTML の場合はブラウザでそのまま表示されるので、デザインの確認がしやすいフレームワークという特徴も持っている。

・ SpringFramework

■特徴

RodJohnson により開発された DI(DependencyInjection)コンテナと呼ばれる、環境に依存する機能を外部から提供可能とする方法が特徴的なオープンソースのフレームワーク。

Spring はオブジェクトの関連付け、生成、登録などを管理する「Core モジュール」、JavaBeans へのアクセス機能を提供する「Context モジュール」、MVC モデルでの開発を提供する「WebMVC モジュール」など 7 種類のモジュールで構成されている。

また、Java プラットフォームで動作する様々なアプリケーションに適用することができるフレームワーク。