

# ～フレームワーク～

フレームワークはプログラミングでアプリ開発をする際に、土台として提供されているもの。

アプリ開発をする際にもものすごく時間がかかってしまいます。そこでアプリ開発をするために必要なものをまとめて提供してくれるものを使って、スムーズに開発を行って行くことがスタンダードな方法です。

「よく使われるライブラリを集めたもの」として考えられる。

- ・ フレームワークのメリット  
開発時間を短くできる  
大人数開発にも参加しやすくなる
- ・ フレームワークのデメリット  
コーディングスキルが比較的向上しづらい  
java の深い理解が得られにくい  
フレームワークによる脆弱性が避けられない

## フレームワークは目的で選ぶ

フレームワークはそれぞれ「得意とする分野が違う」又最低限を知っていればすぐに使い始められる軽量なフレームワークもあれば、多機能だけれどある程度学習に時間を要するものもある。

開発者の知識や開発を担当する機能に応じて考えるもの

## java フレームワークおすすめランキング

3 位 : JSP(Java Server Faces)

2 位 : Play Framework

1 位 : Spring Framework

# ~JSF~

JSFはjavaEE(java Platform,Enterprise Edition)という環境で動作する標準的なコンポーネント思考（機能ごとに部品として扱う考え方）のユーザーインターフェイスフレームワーク

JSF は javaEE に含まれているので新しくライブラリなどをプロジェクトに追加しなくても使える。

一言で言うとユーザーインターフェイス（ユーザーが操作する見た目部分）を簡単に作ることができるフレームワークで、**Struts**（ストラッツ）と同じ**MVC**モデルを採用したフレームワークです。

- ・ 特徴

- 開発チームとデザインチームで分業できる

# ~Play Framework~

Play Framework とは java と scala で作られた WEB フレームワークです  
使うために必要なスキルは、java の基礎知識

- 特徴

「高速開発」

Play Framework は軽量なフレームワークで動作も早い、また「MVC」という設計に基づいて開発される

「リソース消費が少ない」

CPU やメモリーに負担が少なく、少ないリソースでアプリの動作をさせることができる

「高い拡張性」

CSS の拡張言語である SCSS や、javaScript の拡張言語である Coffee スクリプトに始まりリアルタイム通信やテストに関するものまで意図通り揃っている

WEB システム・WEB サービスに向いている

# ~Spring Framework~

Spring Framework は java の WEB フレームワークで、WEB アプリを作ることができる。

Spring Framework では DI(Dependency Injection)と呼ばれる仕組みが導入されており、日本語では（依存性の注入）と訳される。

以前までは変更が弱く、変更が入るたびにかなりの時間を費やす必要があったが、依存性のあるプログラムは外部から取り出せるようになっているので、一つ一つのプログラムを独立にできる。そのため、変更する場合も最低限の変更で開発が楽になる。

また **Spring MVC Test** という専用のプログラムを使うことができる。これは一連の処理を終えるかどうかをテスト（結合テスト）することができ、馬具の混入を最低限に抑え、安全に開発進めることができる。

使うために必要なスキルは、java の基礎知識

- ・ 特徴

「拡張性が高い」

フレームワークの基本的な機能の大部分が「インターフェイス」として提供されているため、必要な機能だけを外部から使うことができる

「保守性が高い」

AOP(Aspect Oriented Programming)というプログラムを用いることができる。これは、クラスには「本質的な処理」のみを書いて、「本質的ではない処理（共通化できる処理）」を別を書くこと

主に必要な処理をそうじゃない処理を分けることで、コードを分かりやす

くする効果がある。それによって、何かバグや不具合があった場合にも原因を特定しやすくなる。

「再利用性が高い」

上記の（AOP）の仕組みにより、共通のプログラムをまとめやすくなるので再利用することが容易になる。

WEB システム・WEB サービスに向いている