

# Django, Ruby on Rails, Spring 비교 그리고 사용방법

## 목차

---

- 1 서론
- 2 Django
- 3 Ruby on Rails
- 4 Spring
- 5 비교 분석
- 6 간단한 프로그래밍 예제
- 7 해당 웹서버 프레임워크를 사용하는 웹사이트
- 8 향후 전망

# 서론

현대 웹 개발에서 프레임워크는 개발 생산성과 코드 품질을 결정하는 핵심 요소이다. 웹 프레임워크는 웹 애플리케이션을 효율적으로 빠르게 개발하는 데 사용할 수 있는 소프트웨어로, 대부분의 웹 애플리케이션에 공통적으로 필요한 기능들을 재사용 가능한 모듈 형태로 제공한다. 본 보고서에서는 현재 가장 널리 사용되는 세 가지 웹 프레임워크인 Django, Ruby on Rails, Spring 을 다각도로 비교 분석하고자 한다.

## Django

Django 는 Python 언어로 작성된 고수준의 웹 프레임워크다. 2003 년 로렌스 저널 월드 신문을 만들던 웹 개발팀의 내부 프로젝트로 시작되어 2005 년 오픈소스 프로젝트로 공개되었다. Python 의 간결하고 직관적인 문법을 기반으로 하여 “배터리 포함(batteries-included)” 철학을 따르며, 웹 개발에 필요한 대부분의 기능을 기본적으로 제공한다

Django 는 MVC 패턴을 기반으로 한 MTV(Model – Template - View) 아키텍처를 사용한다. 주요 특징으로는 강력한 ORM(Object-Relational Mapping) 시스템, 자동으로 구성되는 관리자 화면, 우아한 URL 설계, 자체 템플릿 시스템, 캐시 시스템 등이 있다. Django 의 가장 큰 장점은 빠른 개발 속도로, 기본 CRUD 를 빠르게 개발할 수 있는 viewset 을 제공하고, 다양한 파이썬 라이브러리 사용이 가능하다.

Django 의 주요 단점으로는 레이어의 분리가 제대로 이루어지지 않아 코드 재사용성이 떨어지고, 객체 지향 프로그래밍 학습에는 효과적이지 못하다는 점이 있다. 또한 ORM 성능이 다소 떨어지며, 가상환경 관리의 번거로움과 배포 시 WSGI 서버 설치의 복잡성도 단점으로 지적된다

## Ruby on Rails

Ruby on Rails 는 Ruby 언어로 작성된 웹 애플리케이션 프레임워크로, 웹 애플리케이션을 빠르고 효율적으로 개발할 수 있도록 도와주는 강력한 오픈 소스 프레임워크이다. Rails 는 “Convention over Configuration(규칙이 설정보다 낫다)”라는 핵심 철학을 가지고 있어, 개발자가 명시적인 설정을 많이 하지 않아도 미리 정해진 규칙에 따라 많은 부분을 자동으로 처리한다.

Ruby on Rails 의 주요 특징으로는 MVC 패턴, Active Record ORM, Gem 생태계, Convention Based Routing 등이 있다. Rails 의 가장 큰 장점은 높은 생산성과 빠른 개발 속도로, 반복적인 작업을 자동화할 수 있고 보안성이 높다. 또한 확장성이 높아 다양한 라이브러리와 플러그인을 이용할 수 있으며, Ruby 언어의 간결하고 직관적인 문법으로 인해 쉽게 개발할 수 있다.

Rails 의 단점으로는 적은 수의 개발자가 사용하는 언어이므로 지원되는 라이브러리와 플러그인의 수가 다른 언어와 비교해서 상대적으로 적다는 점이 있다. 또한 높은 생산성 대신 실행 속도가 느리고, 초기 설정이 복잡하며 초기 동작 속도가 느리다는 평가도 있다.

## Spring

Spring Framework 는 Java 플랫폼을 위한 오픈 소스 애플리케이션 프레임워크로, Java 언어 기반의 프레임워크이다. Spring 은 객체 지향 프로그래밍 언어인 Java 의 강력한 장점을 살려낼 수 있도록 개발된 프레임워크로, 좋은 객체 지향 애플리케이션을 개발할 수 있도록 도와주는 프레임워크이다.

Spring Framework 는 핵심 기술(Spring DI 컨테이너, AOP), 웹 기술(Spring MVC, WebFlux), DB 접근 기술(Transaction, JDBC, ORM), 기술 통합, 테스트 기능 등 다양한 기능들이 하나로 이루어진 프레임워크이다. Spring 의 가장 큰 장점은 역할과 레이어의 분리가 제대로 이루어져 DI 개념으로 의존 주입이 자동화 관리되고, 객체 지향 프로그래밍에 최적화되어 있다는 점이다.

Spring 의 주요 단점으로는 초반 러닝 커브가 존재하여 DI, Bean 등의 개념이 처음 접하는 사람에게는 어려울 수 있고, build 시에 오랜 시간이 걸린다는 점이 있다. 또한 gradle 에 dependency 를 추가할수록 build 시간이 증가하며, serverless 환경에서는 사용이 어려워 인프라 비용이 많이 든다는 단점도 있다

## 비교 분석

### 구조 비교

프레임워크	아키텍처 패턴	언어	ORM 지원
Django	MTV	Python	Django ORM
Ruby on Rails	MVC	Ruby	Active Record
Spring	MVC + 계층형 아키텍처	Java	JPA / Hibernate

Django 는 MTV 패턴을 사용하여 Django 에서는 View 를 Template, Controller 를 View 라고 부르지만, 용어만 다를 뿐 개념은 MVC 패턴과 동일하다. Ruby on Rails 와 Spring 은 전통적인 MVC 패턴을 따르며, Spring 의 경우 추가적으로 Service, Repository 계층을 분리하여 더욱 체계적인 구조를 제공한다

### 학습 난이도

학습 난이도 측면에서 Django 는 Python 의 간결한 문법과 “배터리 포함” 철학으로 인해 상대적으로 학습하기 쉽다. Ruby on Rails 는 Convention over Configuration 원칙으로 설정보다는 규칙을 따르면 되어 초기 학습이 용이하지만, 복잡한 프로젝트에서는 학습 곡선이 높아진다. Spring 은 초반 러닝커브가 가장 높으며, DI, Bean, AOP 등의 개념을 이해해야 하므로 학습 난이도가 높다.

### 커뮤니티 및 생태계

Django 는 대규모의 활발한 커뮤니티를 가지고 있으며, 가장 활발하고 탄탄한 공식 문서를 제공한다. Ruby on Rails 는 규모가 크고 활발한 커뮤니티를 가지고 있지만, 상대적으로 개발자 수가 적다. Spring 은 현재 한국 백엔드 개발을 잠악하고 있으며, 한국 블로그 및 참고 사이트의 깊은 개발 영역들은 모두 Spring 을 기반으로 제공되어 백엔드 학습에 매우 유리하다.

### REST API 지원

모든 프레임워크가 REST API 를 잘 지원한다. Django 는 Django REST Framework(DRF)를 통해 RESTful API 서버를 쉽게 구축할 수 있으며, ORM 및 non-ORM 에 대해 모두 Serialization 기능을 제공한다. Ruby on Rails 는 Convention Based Routing 을 통해 RESTful API 디자인 원칙을 따르도록 권장한다. Spring 은 Spring MVC 를 통해 강력한 REST API 지원을 제공하며, Spring Boot 를 사용하면 더욱 간편하게 API 를 개발할 수 있다.

## 확장성 및 성능

성능 측면에서 Django 는 일반적으로 Flask 나 FastAPI 같은 다른 프레임워크에 비해 속도가 느리며, 모놀리식 설계로 인해 확장성이 제한될 수 있다. Ruby on Rails 는 높은 생산성 대신 실행 속도가 느리다는 평가가 있지만, 확장성은 양호하다. Spring 은 Java 의 성능 특성상 상대적으로 우수한 성능을 제공하며, 엔터프라이즈급 애플리케이션에 적합한 확장성을 가지고 있다.

## 간단한 프로그래밍 예제

### Django 예제

```
# models.py
from django.db import models

class Book(models.Model):
    title = models.CharField(max_length=200)
    author = models.CharField(max_length=100)
    published_date = models.DateField()
    isbn = models.CharField(max_length=13)

# views.py
from django.shortcuts import render
from .models import Book

def book_list(request):
    books = Book.objects.all()
    return render(request, 'books/book_list.html', {'books':
books})
```

### Ruby on Rails 예제

```
# app/models/book.rb
class Book < ApplicationRecord
  validates :title, presence: true
  validates :author, presence: true
end

# app/controllers/books_controller.rb
class BooksController < ApplicationController
  def index
    @books = Book.all
  end
end
```

## Spring 예제

```
// Book.java
@Entity
public class Book {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String title;
    private String author;
    // getters and setters
}

// BookController.java
@Controller
public class BookController {
    @Autowired
    private BookService bookService;

    @GetMapping("/books")
    public String listBooks(Model model) {
        model.addAttribute("books", bookService.getAllBooks());
        return "books/list";
    }
}
```

## 각 웹서버 프레임워크를 사용하는 웹사이트

Django 를 사용하는 대표적인 웹사이트로는 Instagram, Mozilla, Disqus, Bitbucket 등이 있다. Instagram 은 널리 사용되는 사진 및 동영상 공유 플랫폼으로 Django 로 개발되어 백엔드와 데이터 저장소를 구동하며, Mozilla 는 Firefox 웹 브라우저를 개발하는 조직으로 Django 를 사용하여 애드온 사이트와 지원 플랫폼을 비롯한 여러 웹 애플리케이션을 구축한다.

Ruby on Rails 를 사용하는 유명한 서비스로는 GitHub, Airbnb, Shopify, Twitter, Twitch, 깃허브, 킥스타터 등이 있다. 국내에서는 당근마켓, 카카오 T, 왓챠, 프로그래머스 등에서 사용되고 있다. 이러한 대규모 서비스들이 Rails 를 사용한다는 것은 Rails 의 확장성과 안정성을 증명하는 좋은 예시이다

Spring Framework 는 현재 한국에서 가장 널리 사용되는 백엔드 프레임워크로, 대부분의 대기업과 금융기관에서 엔터프라이즈급 애플리케이션 개발에 사용되고 있다. Spring 은 안정적이고 확장 가능한 구조로 인해 대규모 시스템 구축에 적합하며, 국내 IT 기업들의 주요 선택지가 되고 있다.

## 향후 전망

2025 년 웹 개발 트렌드에서는 AI 와 웹 개발의 통합이 더욱 가속화될 것으로 예상되며, 노코드(No-Code)와 로우코드(Low-Code) 플랫폼이 웹 개발 생산성 측면에서 더욱 주목받을 것으로 전망된다. 또한 AI 개발 환경이 더욱 빠르게 보급되어 전반적인 사용자 경험과 웹 개발 프로세스 자체를 혁신하는 방향으로 나아갈 것이다

Django 는 Python 의 지속적인 인기와 AI/머신러닝 분야의 성장으로 인해 계속해서 널리 사용될 것으로 예상된다. 특히 데이터 기반 웹 애플리케이션과 API 개발에서 강세를 보일 것으로 전망된다. Ruby on Rails 는 스타트업과 빠른 프로토타이핑이 필요한 프로젝트에서 여전히 선호될 것이며, 개발 생산성 측면에서의 장점이 지속적으로 부각될 것이다. Spring 은 엔터프라이즈 시장에서의 강력한 위치를 유지하며, 특히 한국 시장에서는 계속해서 주도적인 역할을 할 것으로 예상된다.

## 참고 문헌

- AWS. Django 란 무엇인가요? <https://aws.amazon.com/ko/what-is/django/>
- Ruby on Rails Docs, <https://rubyonrails.org/docs>
- Django Docs, <https://docs.djangoproject.com/en/5.2/>
- Spring Docs, <https://spring.io/guides>
- 재능넷. Ruby on Rails vs. 다른 웹 프레임워크: 비교 분석.  
<https://www.jaenung.net/tree/11052>
- 요즘 IT. 2024 년 회고와 2025 년 웹 개발 트렌드 전망.  
<https://yozm.wishket.com/magazine/detail/2910/>