

웹서버 컴퓨팅

AD 프로젝트

20203109

윤영광

Django

- 사용 언어

Python

- 등장 배경, 역사

Django 는 2003 년 Lawrence Journal-World 신문사의 웹 개발팀이 웹사이트를 개발하고 유지보수하기 위하여 개발했다. 원래 신문사의 웹사이트는 PHP 기반으로 쓰였는데, 당시 PHP 버전은 객체지향을 지원하지 않았고 규모가 큰 웹사이트에는 적합하지 않다고 생각했기에 평소에 관심을 가지고 있던 Python 으로 Django 를 개발하게 되었다. Django 는 2005 년 오픈소스로 공개된 이후 수많은 프로젝트에 사용되고 있으며 소규모 ~ 대규모 프로젝트까지 다양한 환경에서 활용되고 있다.

- 장점

Django ORM 은 테이블을 파이썬 클래스로 표현함으로써 복잡한 SQL 쿼리를 작성하지 않고 데이터베이스를 쉽게 조작할 수 있다. 그리고 자동으로 생성되는 관리자 인터페이스를 제공함으로써 웹사이트를 쉽게 관리할 수 있다. 또한 CSRF 보호, SQL 인젝션 방지 등 여러 기능을 내장함으로써 보안이 뛰어나다.

- 간단한 프로그래밍 예제

```
from django.http import HttpResponse

def hello(request):
    return HttpResponse("20203109")
```

- 해당 웹서버 프레임워크를 사용하는 웹사이트 Instagram, Pinterest, Mozilla ...

- 향후 전망 (어떤 프레임워크가 많이 사용될 지)

Django 는 python 을 기반으로 하고 있으며 python 이 데이터 과학, 인공지능 등 지속적으로 여러 분야에서 사용됨으로써 Django 역시 앞으로 수요를 유지할 가능성이 높다. 그리고 Django 는 활발한 커뮤니티를 가지고 있어서 신규 개발자들이 쉽게 접근할 수 있다. 또한 과거부터 현재까지 점진적으로 업데이트를 해오면서 안정성과 최신 트렌드를 모두 잡고 있으며, 앞으로도 현대적인 웹 개발에 맞게 발전할 것이다.

[참고]

* 등장 배경, 역사: [https://www.codedoks.com/ko/docs/1-1-](https://www.codedoks.com/ko/docs/1-1-django%EC%9D%98-%EC%97%AD%EC%82%AC%EC%99%80-%EC%B2%A0%ED%95%99/)

[django%EC%9D%98-%EC%97%AD%EC%82%AC%EC%99%80-%EC%B2%A0%ED%95%99/](https://www.codedoks.com/ko/docs/1-1-django%EC%9D%98-%EC%97%AD%EC%82%AC%EC%99%80-%EC%B2%A0%ED%95%99/)

* 장점: <https://www.codedoks.com/ko/docs/1-2->

[django%ec%9d%98-%ec%a3%bc%ec%9a%94-%ea%b8%b0%eb%8a%a5/](https://wikidocs.net/197525)

* 해당 웹서버 프레임워크를 사용하는 웹사이트: <https://wikidocs.net/197525>

Express.js

- 사용 언어

JavaScript

- 등장 배경, 역사

2009 년에 **Node.js** 가 개발되어 서버에서도 **javascript** 를 사용할 수 있게 되었지만 복잡한 웹 애플리케이션을 개발하는데 한계가 있었다. 때문에 많은 개발자들이 복잡한 웹 애플리케이션을 쉽게 개발할 수 있는 프레임워크를 원했고

2010 년에 TJ Holowaychuk 가 **express.js** 를 개발했다. **Express.js** 는 처음부터 미들웨어 시스템과 라우팅을 제공하여 개발자들이 복잡한 **HTTP** 서버 애플리케이션을 쉽게 구축할 수 있도록 했고, 간단한 인터페이스와 기본적인 기능을 제공하여 개발자들에게 인기를 끌었다.

- 장점

Express.js 는 간단하고 명확한 코드를 제공함으로써 최소한의 코드로 복잡한 웹 애플리케이션을 구현할 수 있다. 그리고 미들웨어 시스템을 통해 요청과 응답 사이에 다양한 기능을 추가할 수 있는데 재사용 가능한 모듈을 만들고 관리하는데 유용하다. 또한 다양한 **HTTP** 메소드를 지원하고 **URL** 패턴 매칭을 통하여 복잡한 라우팅을 쉽게 관리할 수 있다.

- 간단한 프로그래밍 예제

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('20203109');
});

app.listen(port, () => {
  console.log(`Server is running at http://localhost:${port}`);
});
```

- 해당 웹서버 프레임워크를 사용하는 웹사이트

MySpace, LearnBoost, Storify ...

- 향후 전망 (어떤 프레임워크가 많이 사용될 지)

Express.js 는 현재도 활발하게 최신 웹 개발 트렌드와 요구사항을 반영하여 지속적으로 업데이트되고 있다.. **Node.js** 생태계 내에서 **Express.js** 는 여전히 가장 널리 사용되는 프레임워크 중 하나로, 많은 기업과 개발자들이 이를 사용하여 웹

애플리케이션을 개발하고 있다. 때문에 **express.js** 는 웹 애플리케이션 개발의 필수 도구로 자리 잡음으로써 앞으로도 웹 개발에 있어서 큰 영향을 미칠 것으로 예상된다.

[참고]

- * 등장 배경, 역사: <https://developers.redhat.com/articles/2021/12/03/introduction-nodejs-reference-architecture-part-6-choosing-web-frameworks#>
- * 장점: <https://blog.logrocket.com/node-js-alternative-frameworks-express-js/>
- * 해당 웹서버 프레임워크를 사용하는 웹사이트: <https://codecondo.com/15-websites-built-with-express-js/>

Ruby on Rails

- 사용언어

Ruby

- 등장 배경, 역사

2000년대 초 웹 애플리케이션 개발이 복잡했을 때 다양한 기능을 구현하는데 많은 부분이 수작업으로 이루어졌고 개발 속도가 느렸다. 1995년 Yukihiro Matsumoto가 Ruby를 개발하여 강력한 객체지향 프로그래밍 기능을 제공하였고, 2003년 David Heinemeier Hansson이 Basecamp를 개발하면서 Rails의 기본 개념을 정립했다. 2004년 7월 David Heinemeier Hansson은 Ruby on Rails의 첫 번째 버전의 공개를 시작으로 2024년 5월 Rails 7.1.3.3까지 업데이트 되면서 지금까지 꾸준히 사용되고 있다.

- 장점

Ruby 언어는 명확하고 쉬운 문법을 가지고 있어 직관적으로 코드를 해석할 수 있으며, 정리가 잘된 공식 문서와 블로그, 튜토리얼이 있어 다른 언어와 비교하여 낮은 진입장벽을 가지고 있다. 그리고 활발한 오픈소스 커뮤니티가 존재하여 많은 개발자들이 지속적으로 유지보수하고 있으며 프레임워크 또한 빠르게 발전하고 있다. 또한 내장된 테스트 프레임워크를 통하여 테스트 주도 개발을 쉽게 수행할 수 있어 코드의 품질과 안정성을 높일 수 있다.

- 간단한 프로그래밍 예제

```
class HelloWorldController < ApplicationController
  def hello
    render plain: "Hello, world!"
  end
end
```

- 해당 웹서버 프레임워크를 사용하는 웹사이트
Github, Airbnb, shopify ...

- 향후 전망 (어떤 프레임워크가 많이 사용될 지)

Ruby on rails 은 커뮤니티가 활발하여 지속적인 업데이트와 새로운 기능을 추가하고 있다. 그리고 여전히 많은 기업들이 **ruby on rails** 을 사용하여 웹 애플리케이션을 개발하고 유지보수하고 있는 만큼 **ruby on rails** 의 안정성과 신뢰성을 보장받을 수 있으며 이러한 장점들이 앞으로도 웹 개발 분야에서 경쟁력을 갖추고 중요한 위치를 유지할 수 있는 가능성을 보여준다.

[참고]

* 등장 배경, 역사: <https://guides.rubyonrails.org/> ,

https://ko.wikipedia.org/wiki/%EB%A3%A8%EB%B9%84_%EC%98%A8_%EB%A0%88%EC%9D%BC%EC%A6%88

* 장점: <https://blog.dramancompany.com/2020/11/java-spring-ruby-on-rails/>

* 해당 웹서버 프레임워크를 사용하는 웹사이트: <https://rubyonrails.org/>