
django

Python
web framework

Django - Why? How?

Django - Why?

우리가 Django를 배우는 이유

1. Python 기반의 프레임워크

2. 검증된 프레임워크

Spotify

Instagram

Dropbox

Delivery Hero

2. 검증된 프레임워크

Version	Date[36]	Notes[37]
0.90[38]	16 Nov 2005	
0.91[39]	11 Jan 2006	"new-admin"
0.95[40]	29 Jul 2006	"magic removal"
0.96[41]	23 Mar 2007	"newforms", testing tools
1.0[42]	3 Sep 2008	API stability, decoupled admin, unicode
1.1[43]	29 Jul 2009	Aggregates, transaction based tests
1.2[44]	17 May 2010	Multiple db connections, CSRF, model validation
1.3[45]	23 Mar 2011	Class based views, staticfiles
1.4 LTS[46]	23 Mar 2012	Timezones, in browser testing, app templates.
1.5[47]	26 Feb 2013	Python 3 Support, configurable user model
1.6[48]	6 Nov 2013	Dedicated to Malcolm Tredinnick, db transaction management, connection pooling.
1.7[49]	2 Sep 2014	Migrations, application loading and configuration.
1.8 LTS[50]	1 Apr 2015	Native support for multiple template engines. *Supported until at least April 2018*
1.9[51]	1 Dec 2015	Automatic password validation. New styling for admin interface.
1.10[52]	1 Aug 2016	Full text search for PostgreSQL. New-style middleware.
1.11 LTS[53]	4 Apr 2017	Last version to support Python 2.7. *Supported until at least April 2020*
2.0[54]	2 Dec 2017	First Python 3-only release, Simplified URL routing syntax, Mobile friendly admin.
2.1[55]	1 Aug 2018	Model "view" permission.
2.2 LTS[56]	1 Apr 2019	Security release. *Supported until at least April 2022*
3.0[57]	2 Dec 2019	
3.1[58]	Aug 2020	
3.2 LTS[58]	Apr 2021	*Supported until at least April 2024*
Old version	Older version, still supported	**Latest version**	Latest preview version	Future release		

출처: 위키백과(django)

2005년 v1.0

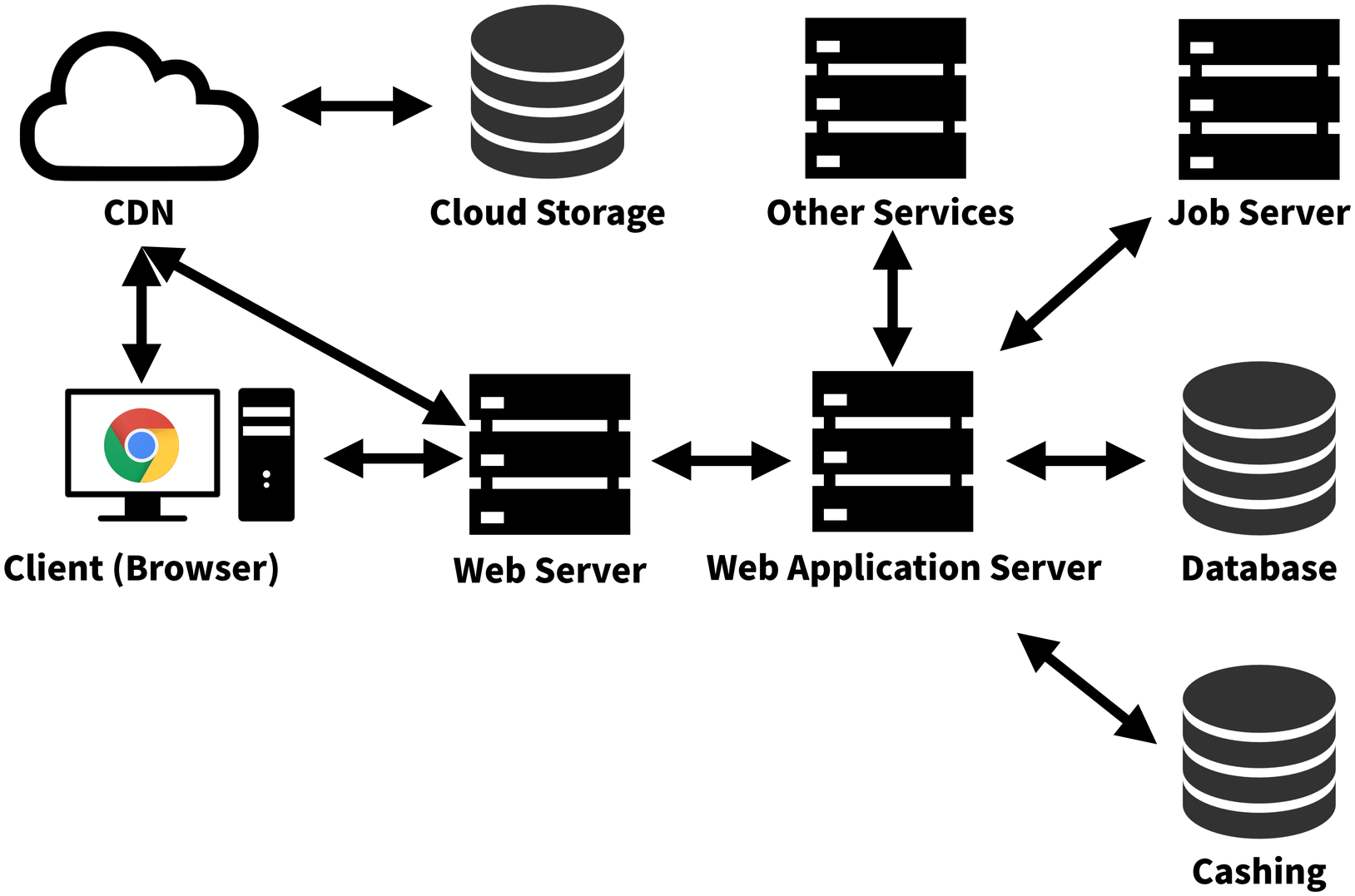
2017년 v2.0

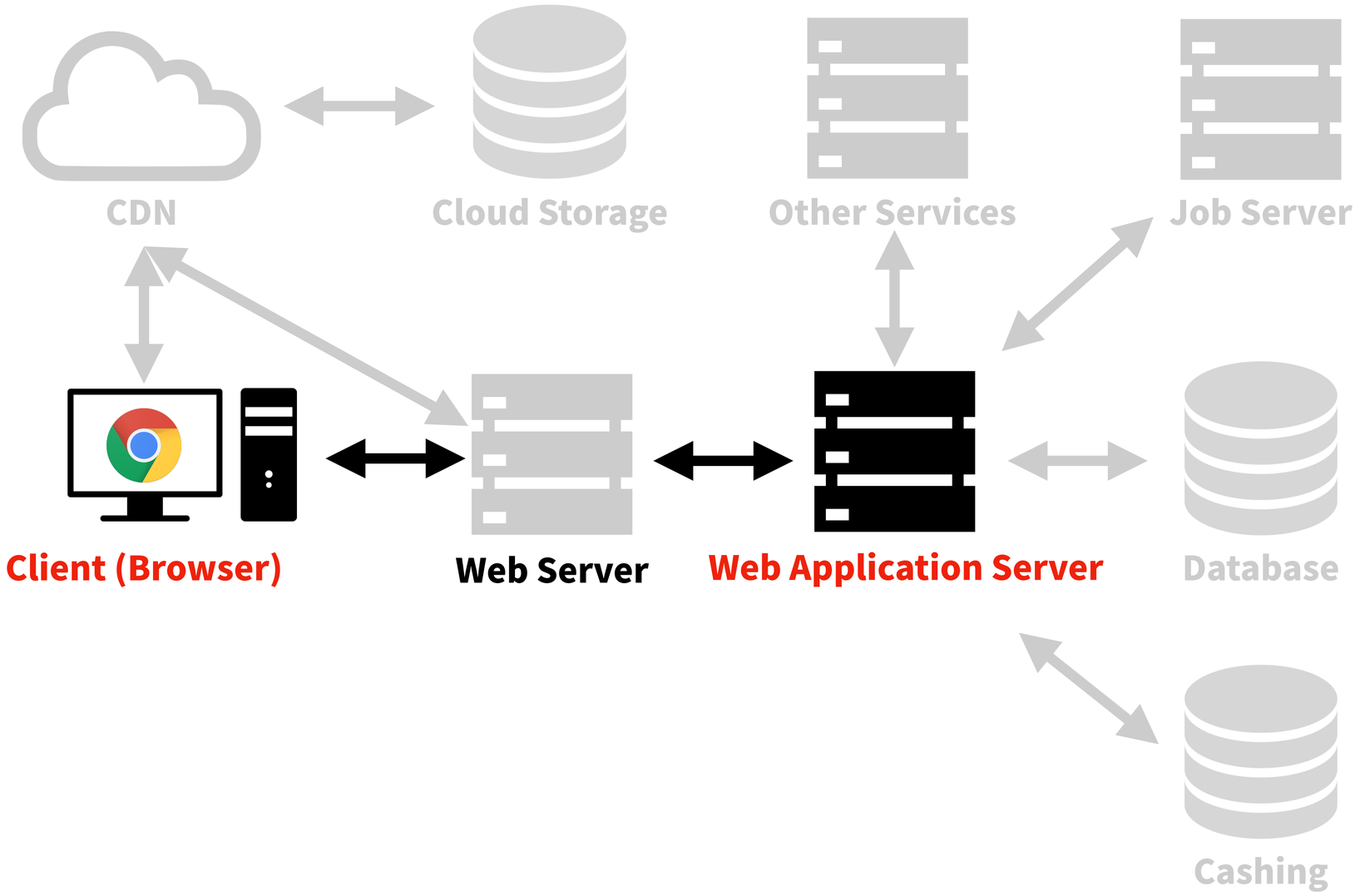
2019년 v3.0

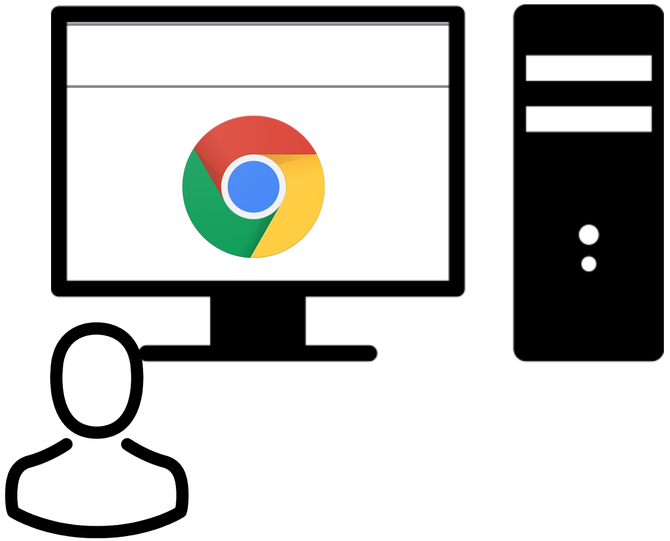
Django - How?

어떻게 동작 하는건가요?

Web Application

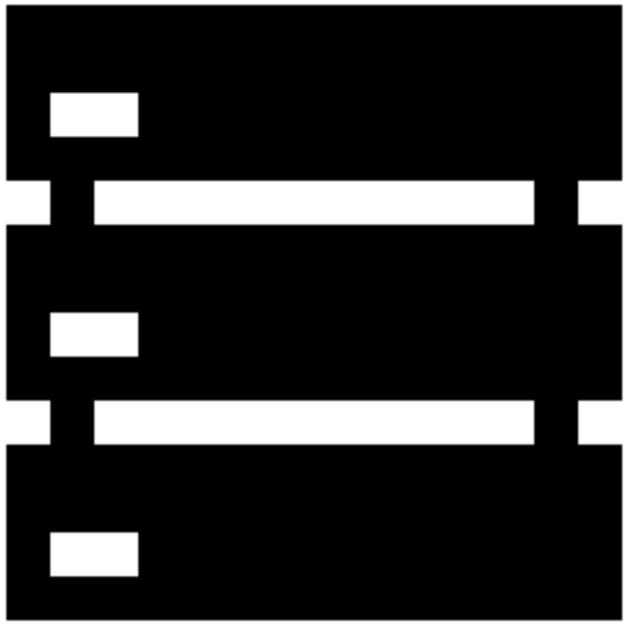

CDN
Cloud Storage
Other Services
Job Server
Client (Browser)
Web Server
Web Application Server
Database
Cashing


CDN
Cloud Storage
Other Services
Job Server
Client (Browser)
Web Server
Web Application Server
Database
Cashing



Request(요청)

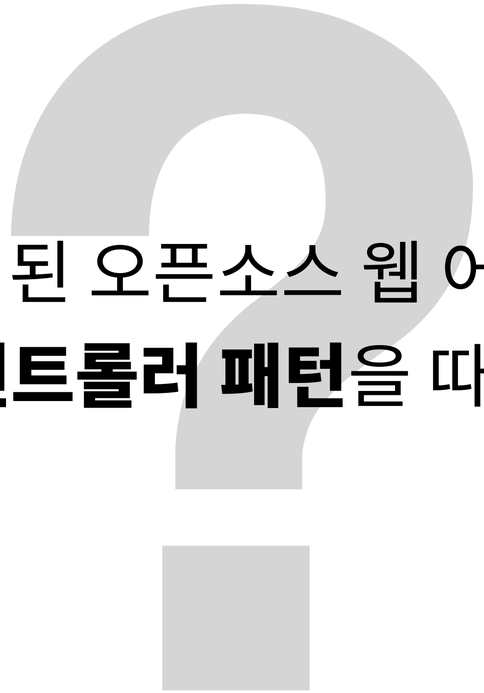
Respond(응답)



클라이언트 (Client)

서버 (Server)


django



django는 파이썬으로 작성된 오픈소스 웹 어플리케이션 프레임워크로,
모델-뷰-컨트롤러 패턴을 따르고 있다.

출처: 위키백과(django)

모델-뷰-컨트롤러 패턴

모델-뷰-컨트롤러(Model–View–Controller, MVC)는 소프트웨어 공학에서 사용되는 **소프트웨어 디자인 패턴**이다.

이 패턴을 성공적으로 사용하면, 사용자 인터페이스로부터 비즈니스 로직을 분리하여 애플리케이션의 시각적 요소나 그 이면에서 실행되는 비즈니스 로직을 서로 영향 없이 쉽게 고칠 수 있는 애플리케이션을 만들 수 있다.

MVC에서 모델은 애플리케이션의 정보(데이터)를 나타내며, 뷰는 텍스트, 체크박스 항목 등과 같은 사용자 인터페이스 요소를 나타내고, 컨트롤러는 데이터와 비즈니스 로직 사이의 상호동작을 관리한다.

출처: 위키백과(모델, 뷰, 컨트롤러)

소프트웨어 디자인 패턴
MVC

Model

View

Controller

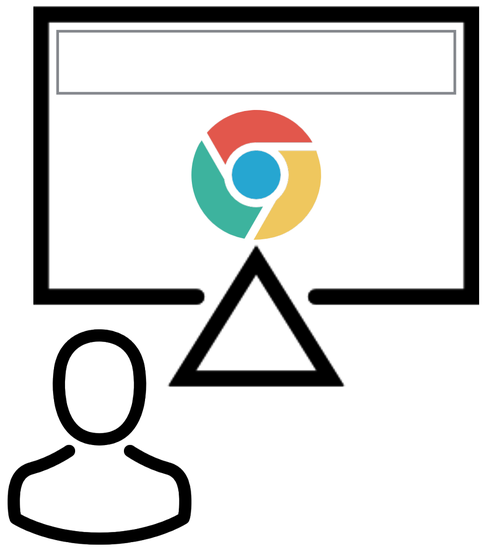
Model

Template

View

MTV

Model	Template	View
M	**T**	**V**
데이터 관리 (DB 동작)	사용자가 보는 화면 (인터페이스)	중간 관리자 (상호 동작)

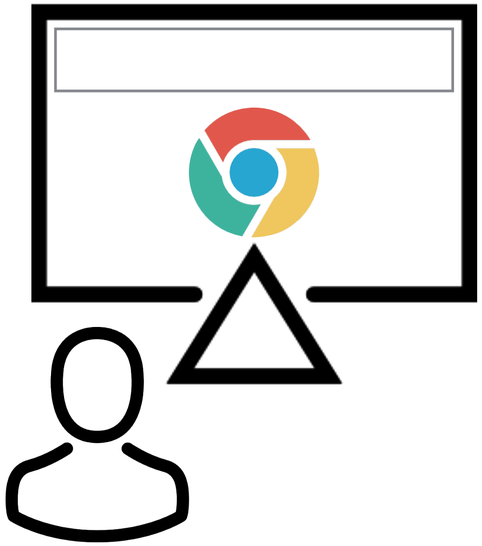


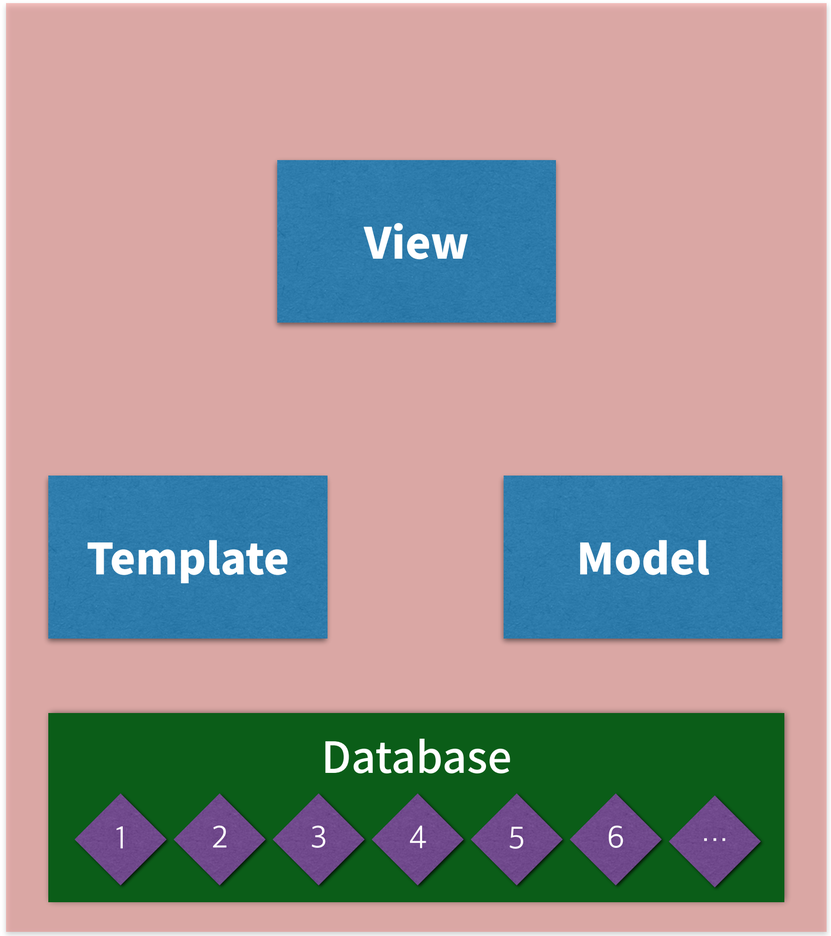
Request(요청)

Respond(응답)

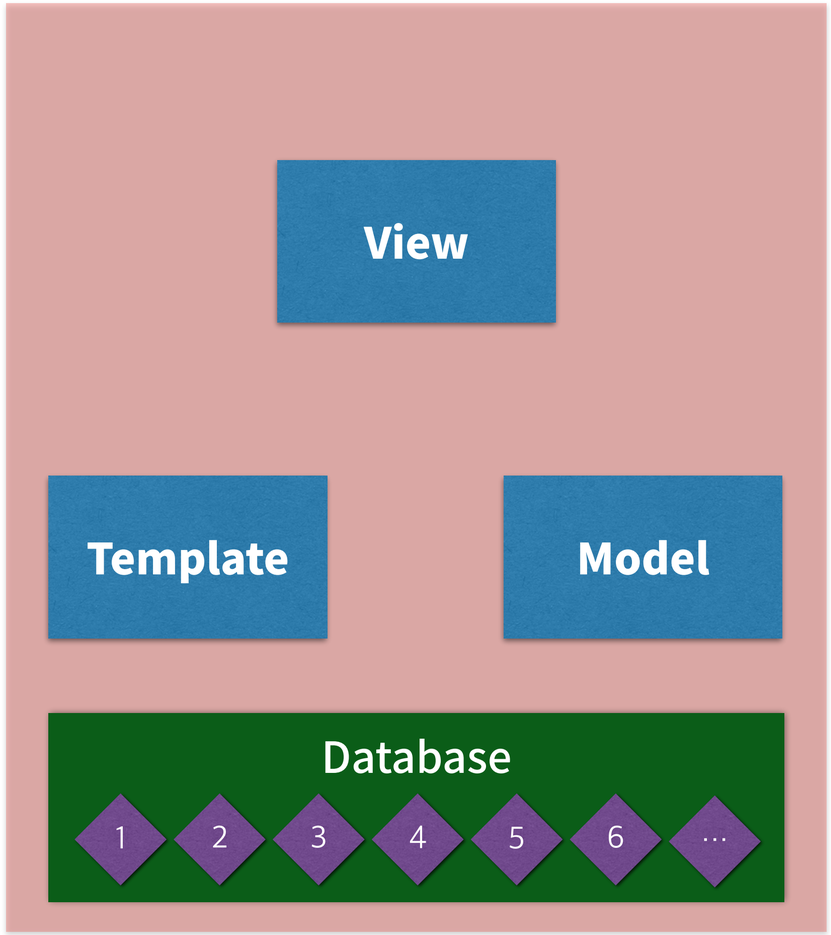
Web Application

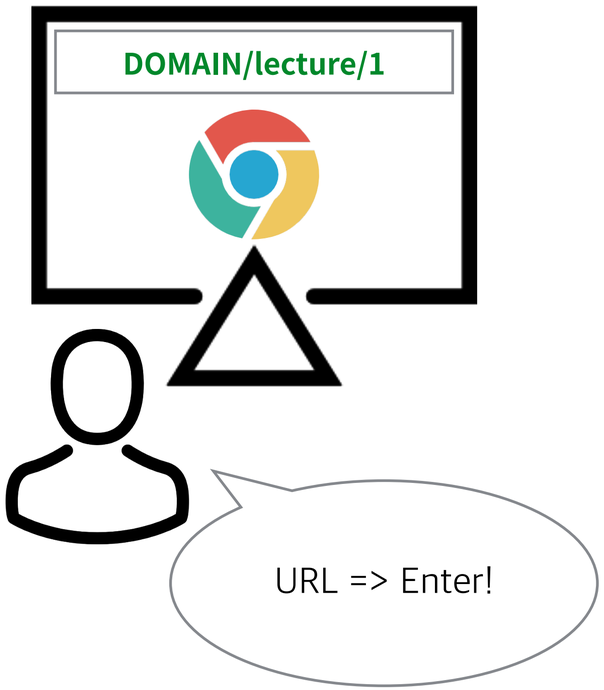
django

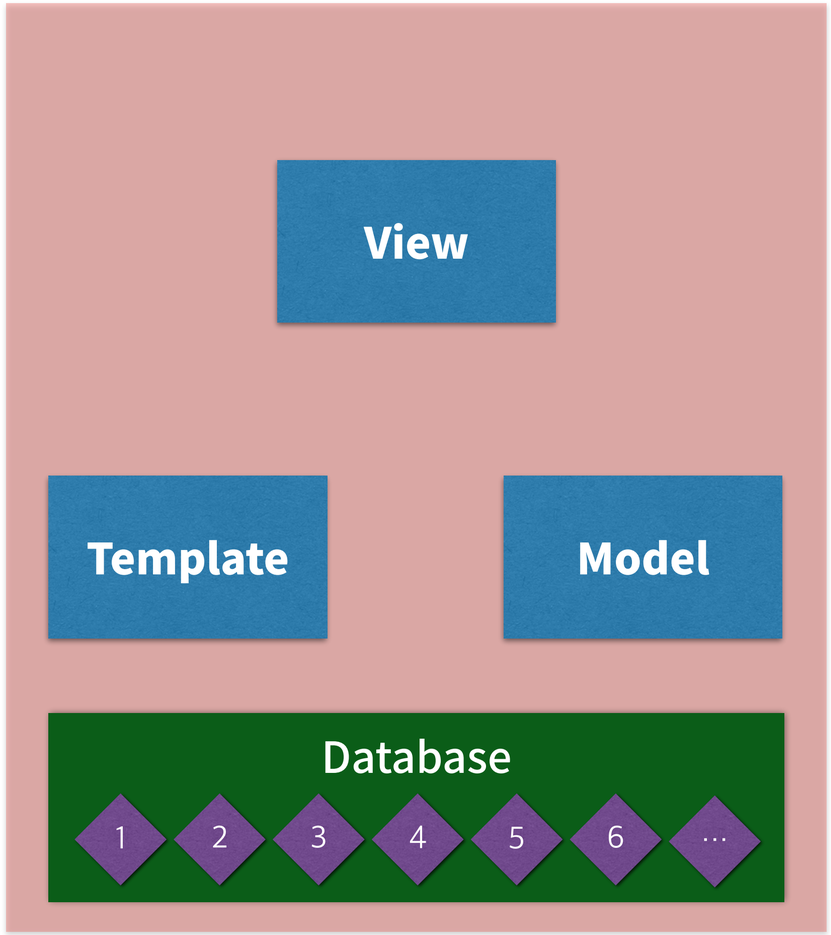


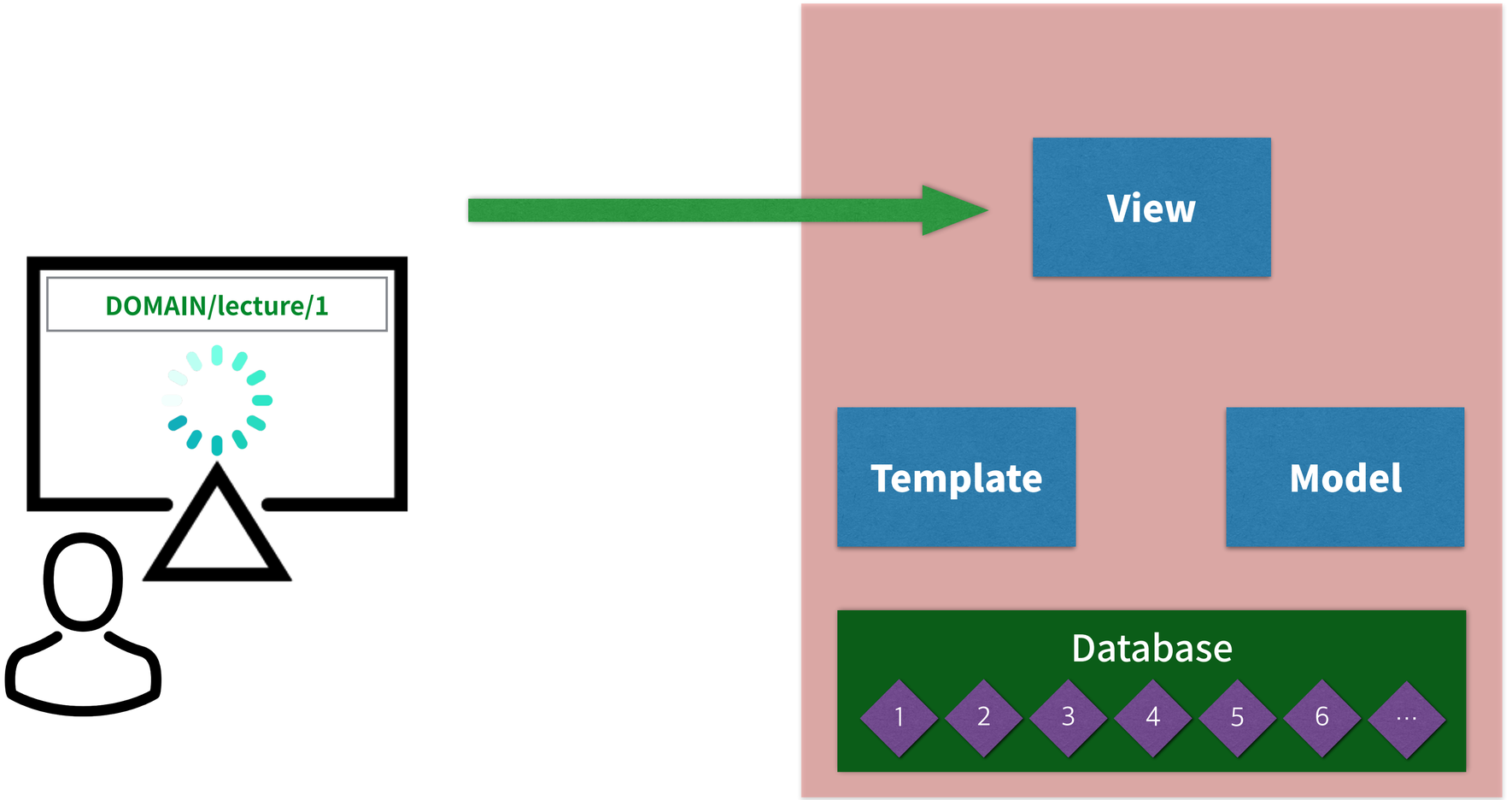

View
Template
Model
Database
1
2
3
4
5
6
...

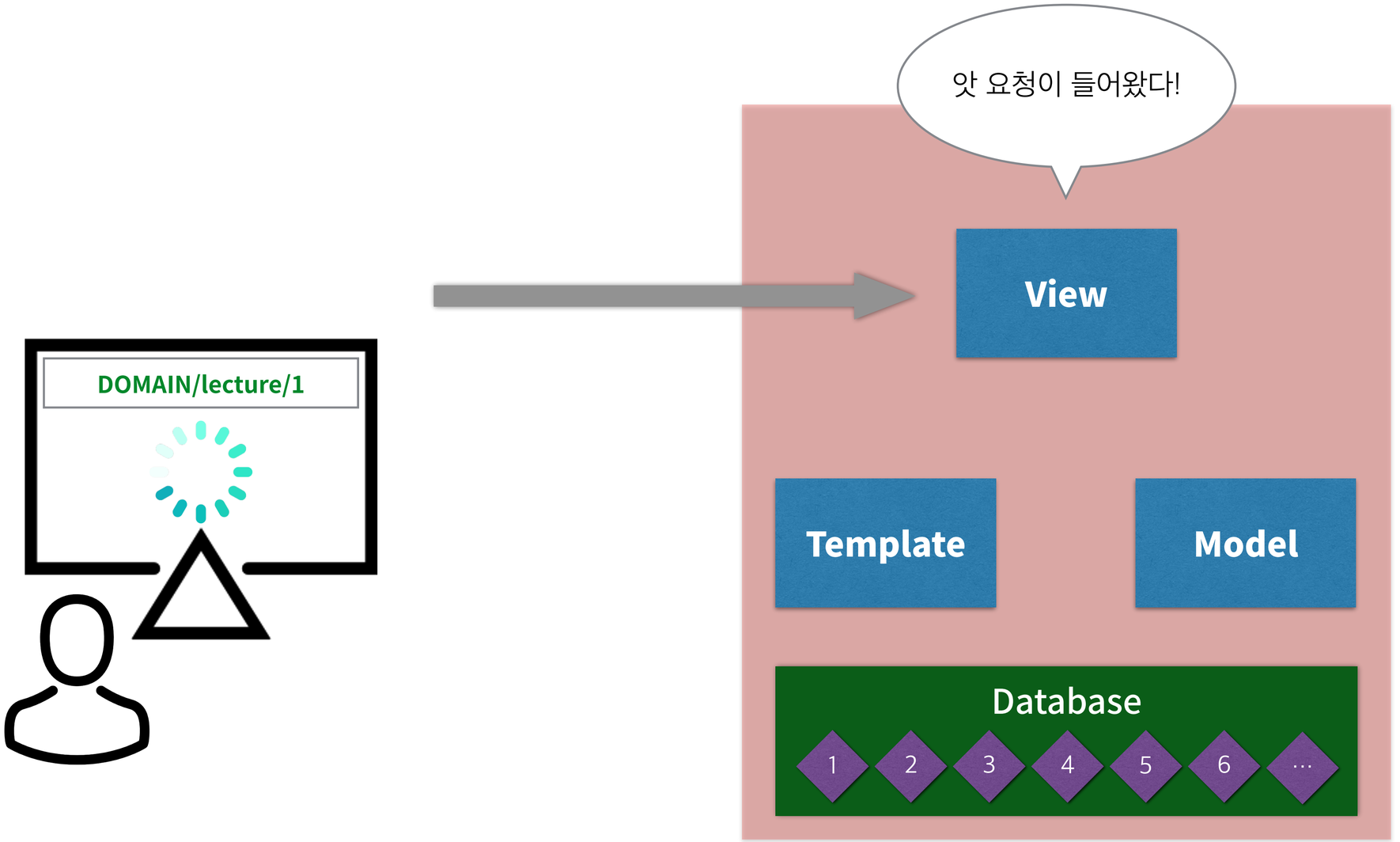

1번 강의를 봐야지!

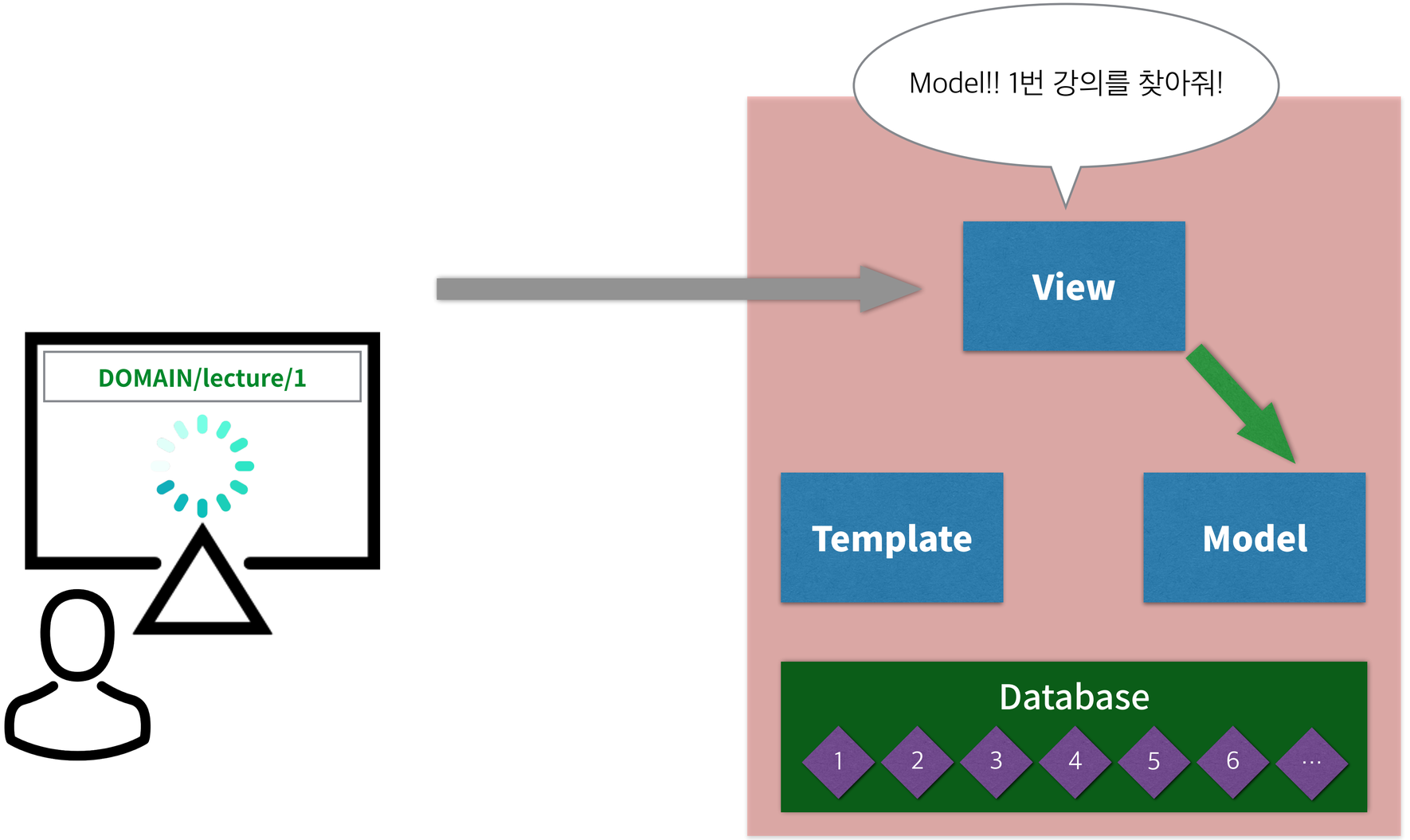

View
Template
Model
Database
1
2
3
4
5
6
…

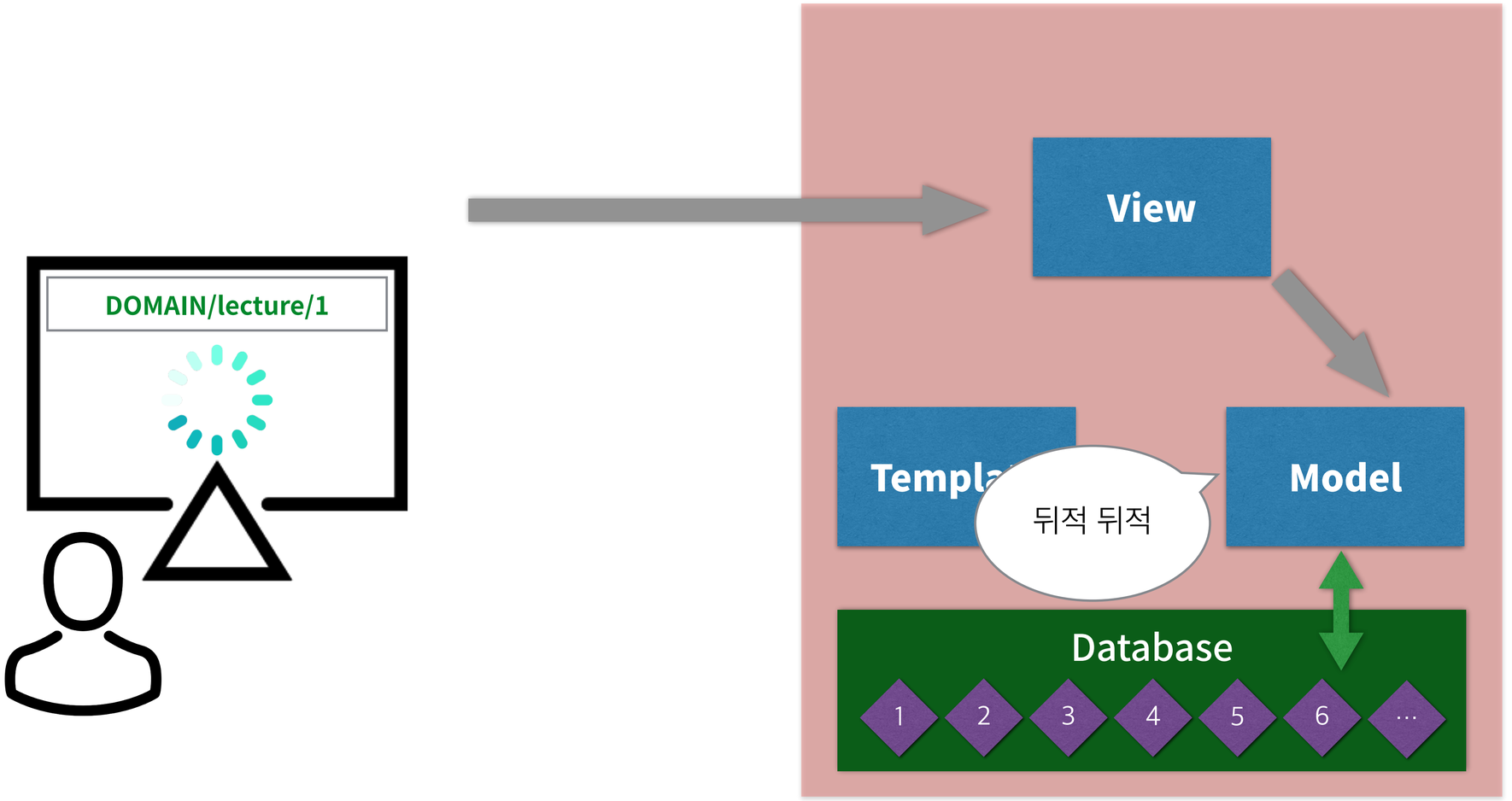

DOMAIN/lecture/1
URL => Enter!

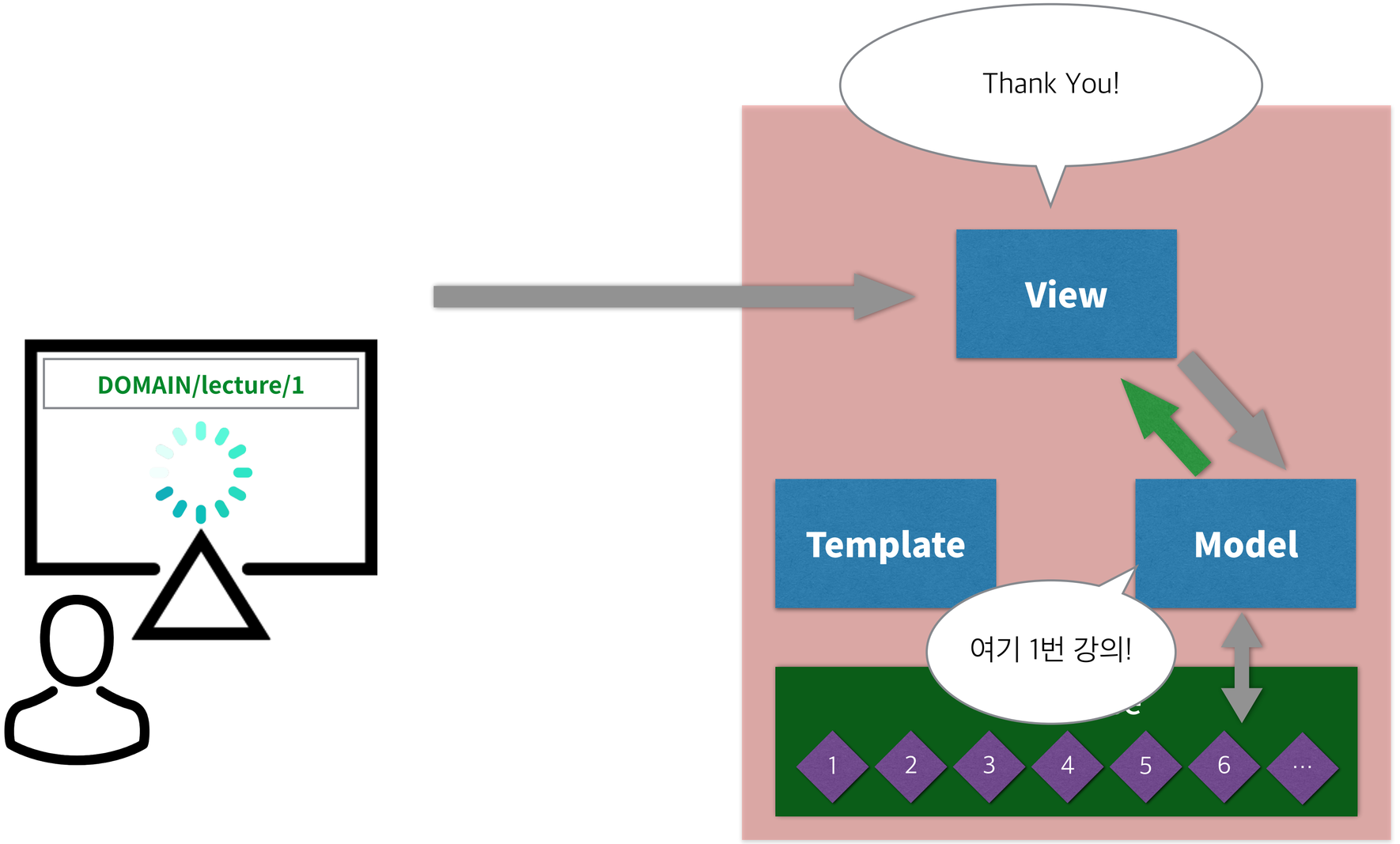

View
Template
Model
Database
1
2
3
4
5
6
…

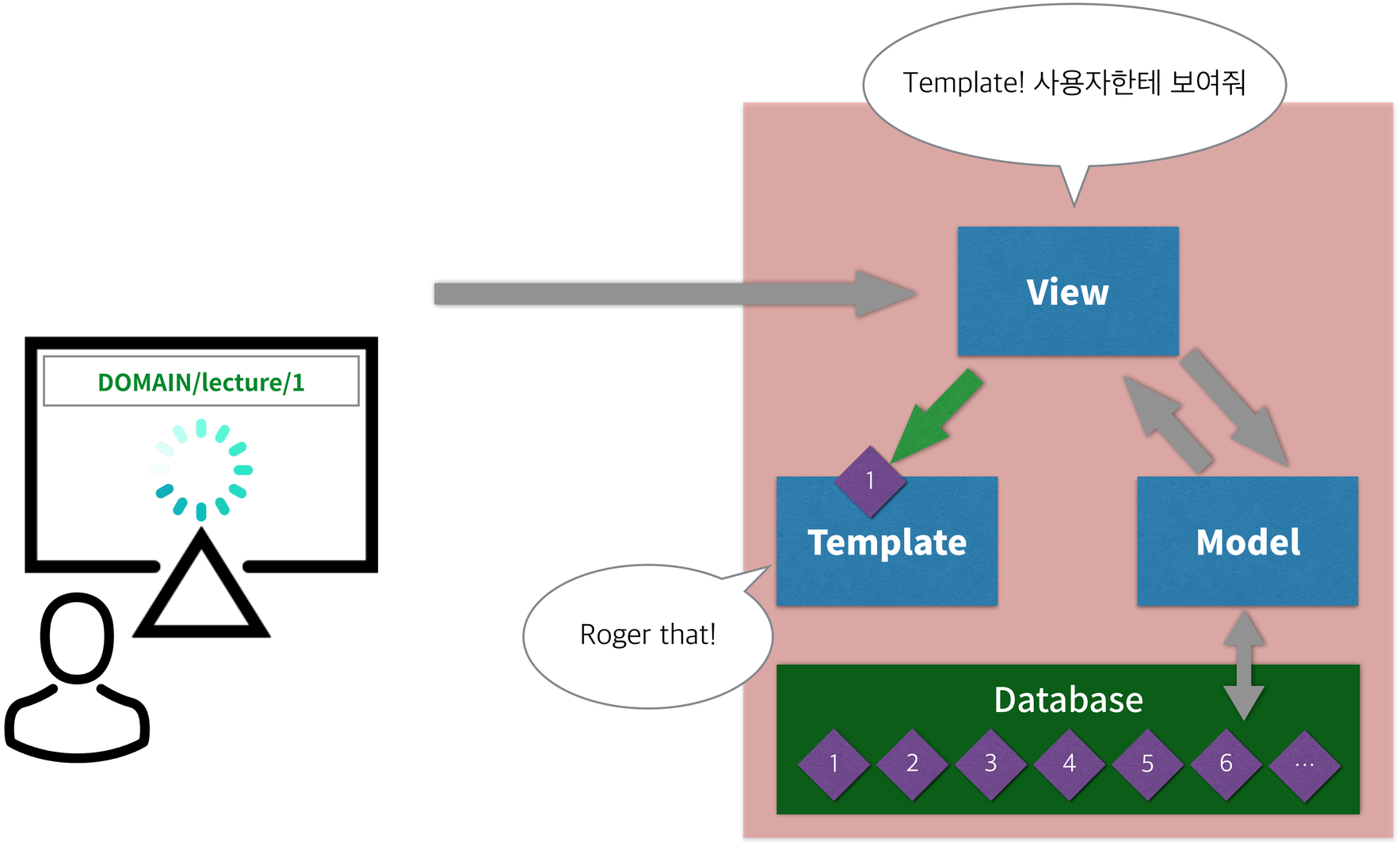

DOMAIN/lecture/1
View
Template
Model
Database
1
2
3
4
5
6
…

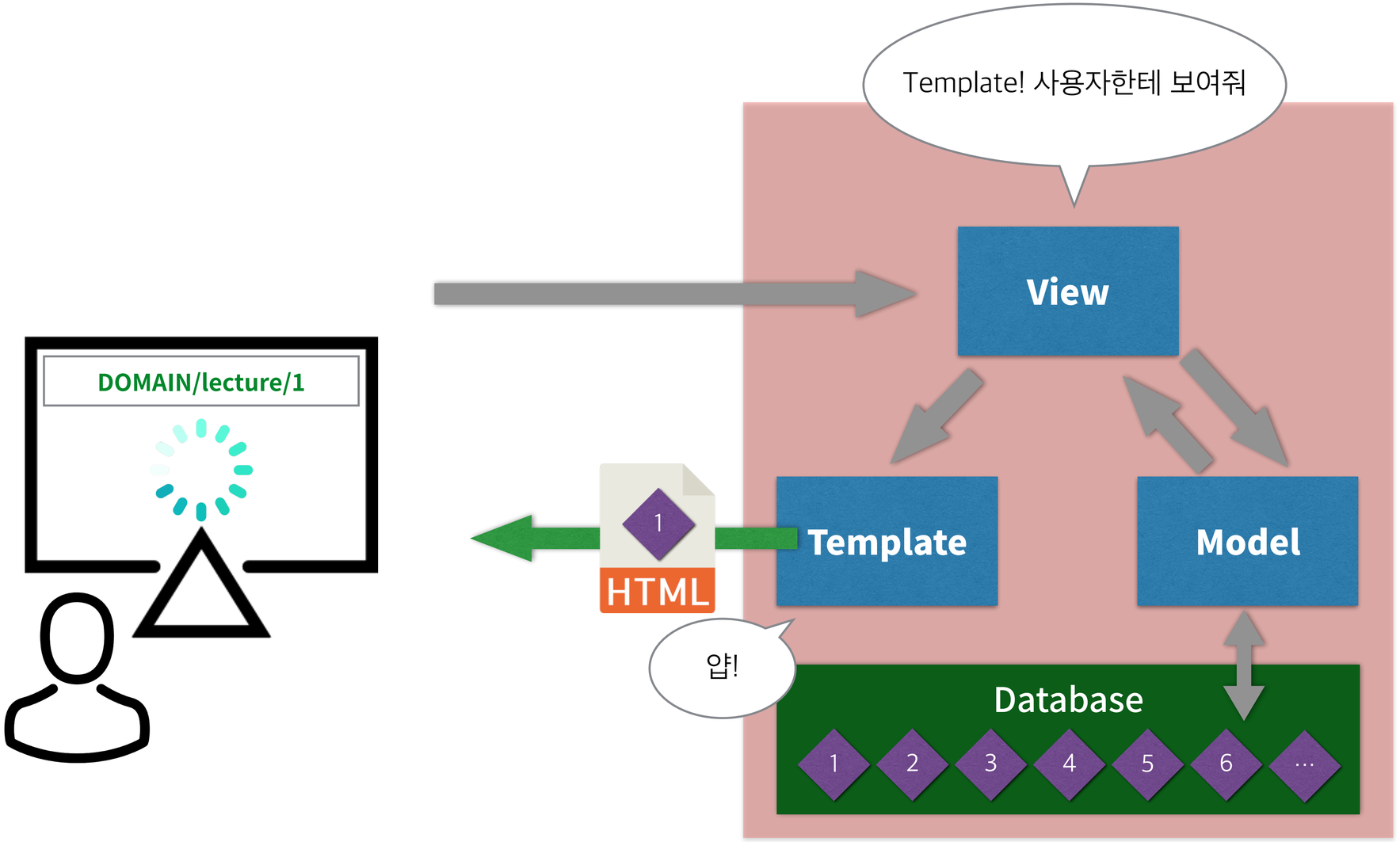

DOMAIN/lecture/1
앗 요청이 들어왔다!
View
Template
Model
Database
1
2
3
4
5
6
…

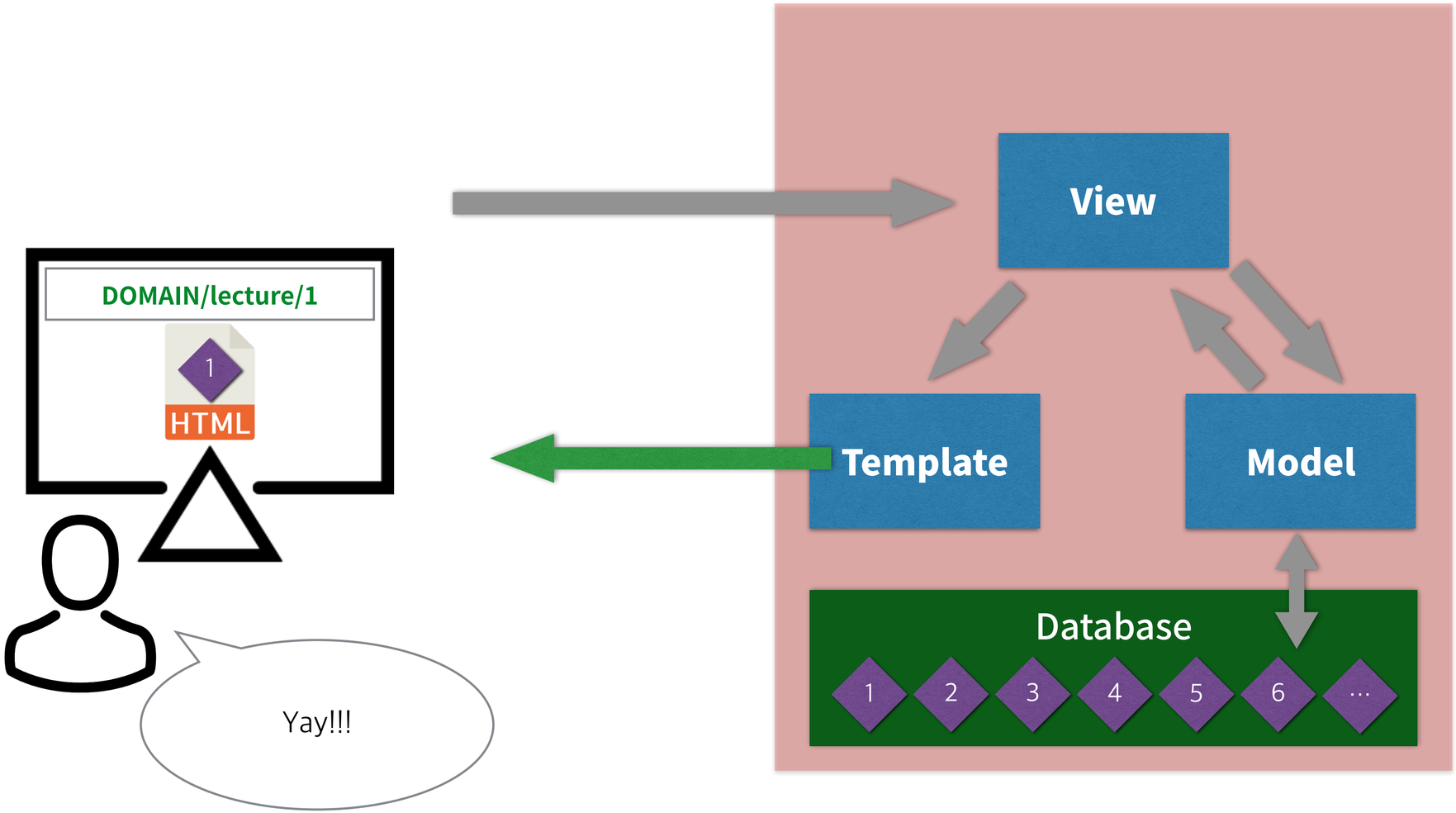

DOMAIN/lecture/1
Model!! 1번 강의를 찾아줘!
View
Template
Model
Database
1
2
3
4
5
6
…


DOMAIN/lecture/1
View
Templa
뒤적 뒤적
Model
Database
1
2
3
4
5
6
…


DOMAIN/lecture/1
Thank You!
View
Template
Model
여기 1번 강의!
1
2
3
4
5
6
…


DOMAIN/lecture/1
Template! 사용자한테 보여줘
View
1
Template
Model
Roger that!
Database
1
2
3
4
5
6
…

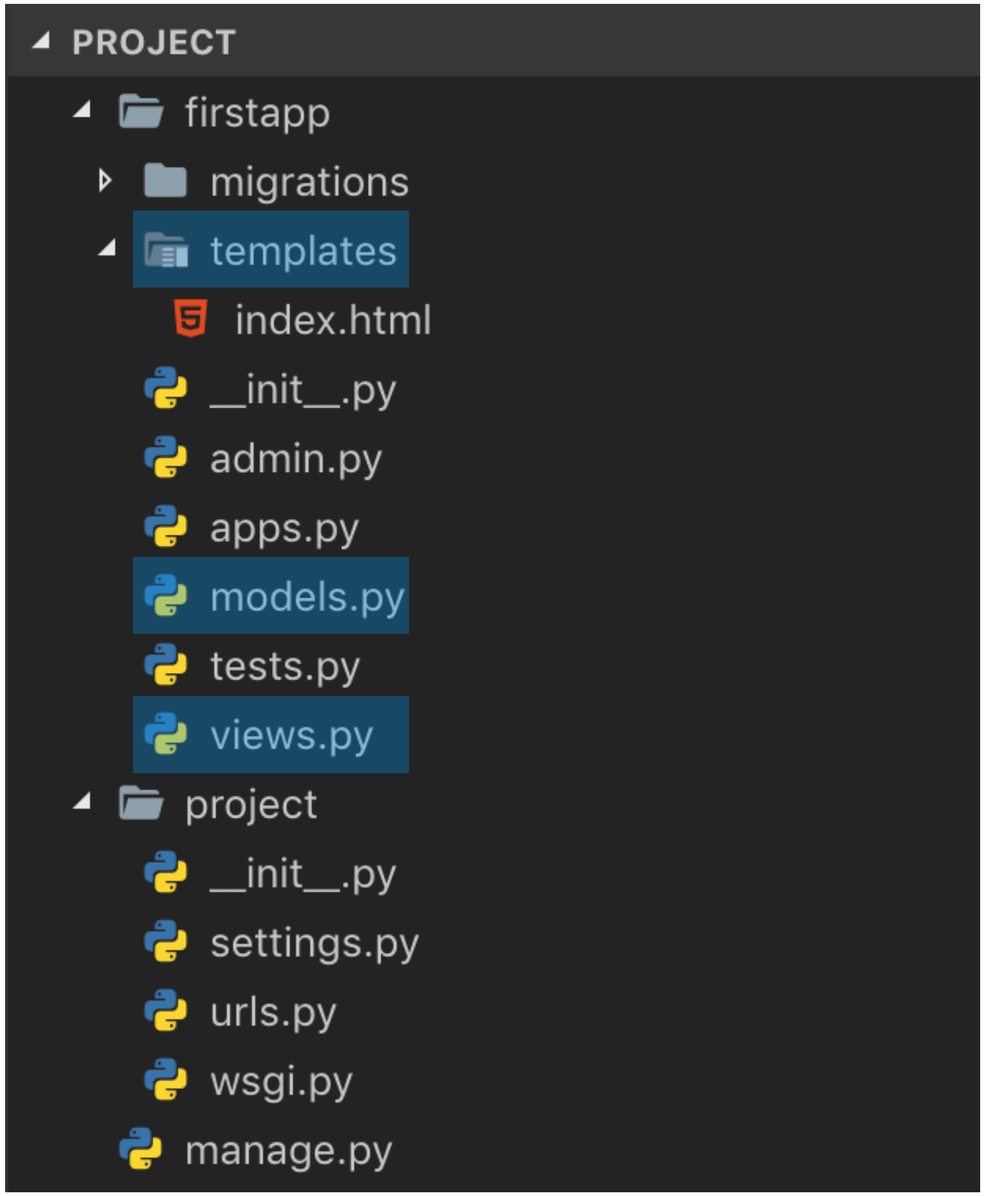

DOMAIN/lecture/1
Template! 사용자한테 보여줘
View
Template
Model
1
HTML
얍!
Database
1
2
3
4
5
6
…


View
Template
Model
Database
1
2
3
4
5
6
…
DOMAIN/lecture/1
1
HTML
Yay!!!

MTV

Model	Template	View
M	**T**	**V**
데이터 관리 (DB 동작)	사용자가 보는 화면 (인터페이스)	중간 관리자 (상호 동작)

First django App



Hello, Django!

1. 패키지 설치

```
$ pip install django
```

2. 프로젝트 생성

```
$ django-admin startproject project
```

3. app 생성

```
$ python manage.py startapp firstapp
```