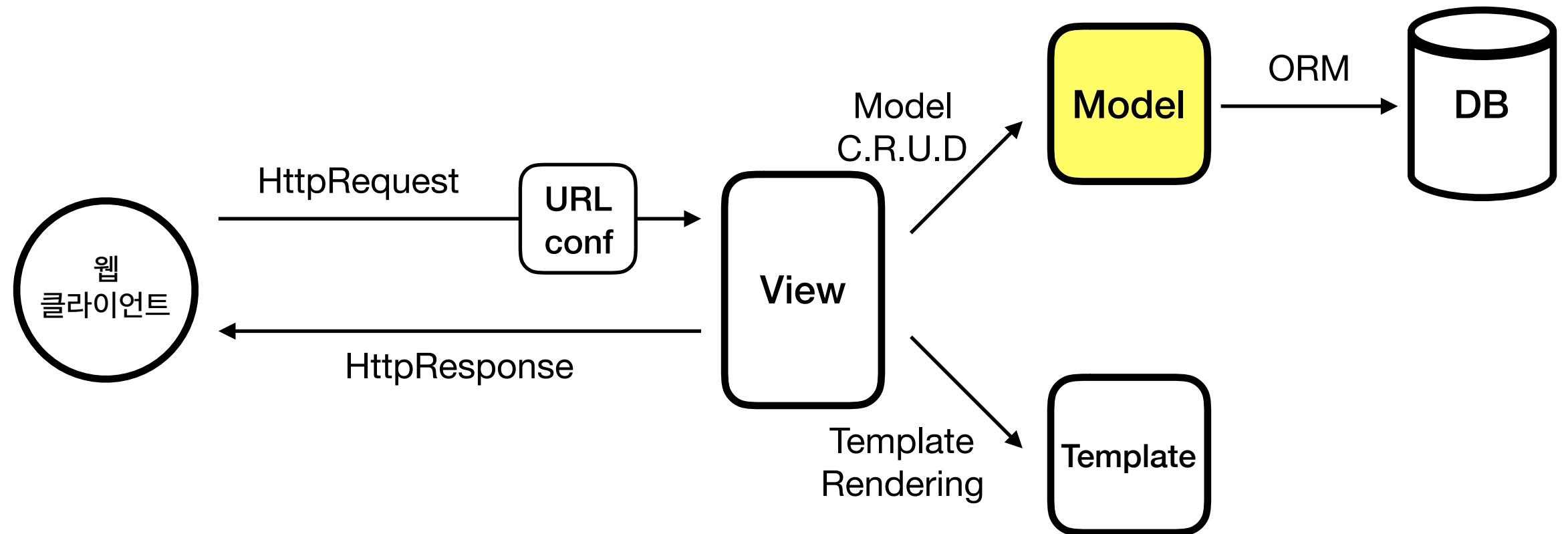


# 웹 프로그래밍 실전

# 3. Model 코딩

# MTV 계층



# 모델(Model)

- 데이터베이스에 저장되는 데이터를 의미한다.
- ORM(Object Relation Mapping) 기법 : 객체와 관계형 데이터베이스를 연결해주는 역할
- 테이블을 하나의 클래스로 정의한다.
- 테이블의 열은 클래스 변수(속성)으로 정의한다.
- 즉, 모델이란 데이터에 대한 정의를 담고 있는 장고의 클래스

# models.py

- 테이블 클래스는 `django.db.models.Model` 클래스를 상속받아 정의한다.
- 클래스 변수의 타입은 장고에서 미리 정의된 필드 클래스를 사용한다.
- PK는 클래스에 지정해주지 않아도 장고에서 `NotNull`, `Autoincrement`로 자동으로 만들어준다.
- FK는 항상 다른 테이블의 PK에 연결되므로 클래스만 지정하면 된다. 실제 테이블에서 FK로 지정된 열은 `_id`가 붙는다.

# models.py

```
class Question(models.Model):  
  
    question_text = models.CharField(max_length=200)  
  
    pub_date = models.DateTimeField('date published')
```

# \_\_str\_\_ 메서드 오버라이딩

- \_\_str\_\_() 메서드는 객체를 문자열로 표현할 때 사용하는 함수이다.

```
def __str__(self):  
  
    return self.question_text
```

# Model Fields Type

- <http://docs.djangoproject.com/ko/2.1/ref/models/fields#model-field-types>



# Foreign Key

- 1:N 관계
- `django.db.models.ForeignKey` 사용
- `ForeignKey(to, on_delete)`
  - `to` : 대상 모델
  - `on_delete` : 1측 열이 삭제될 경우 N측 열에 대한 동작을 지정
    - CASCADE
    - PROTECT
    - SET\_NULL/SET\_DEFAULT/SET
    - DO\_NOTHING

# models.py

```
class Choice(models.Model):  
  
    question = models.ForeignKey(Question, on_delete=models.CASCADE)  
  
    choice_text = models.CharField(max_length=200)  
  
    votes = models.IntegerField(default=0)  
  
    def __str__(self):  
  
        return self.choice_text
```

# 모델 활성화

- 모델 클래스로부터 마이그레이션 파일 생성
- `python manage.py makemigrations polls`
- 마이그레이션 파일로부터 데이터베이스 테이블 생성
- `python manage.py migrate polls`

# admin.py

- models.py 파일에서 정의한 테이블이 Admin 사이트에 보이도록 등록한다.
- 테이블을 새로 만들 때는 models.py와 admin.py 두 개의 파일을 함께 수정해야 한다.

```
from polls.models import Question, Choice
```

```
admin.site.register(Question)
```

```
admin.site.register(Choice)
```