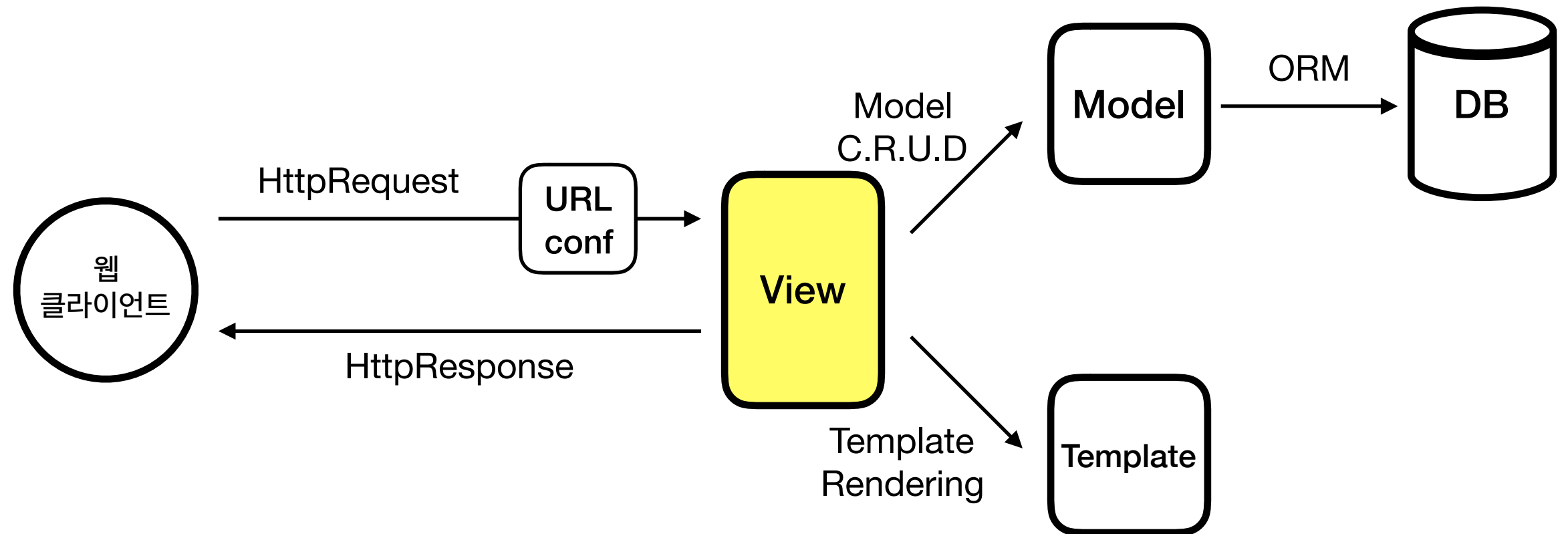


# 웹 프로그래밍 실전

# 5. View 코딩

# MTV 계층



# 뷰(View)

- 함수 또는 클래스의 메서드로 작성되며 웹 요청을 받고 응답을 반환하는 역할
- `views.py` : 다양한 형태의 응답 데이터를 만들어내기 위한 로직을 작성하는 파일
- 함수형 뷰(FBV)와 클래스형 뷰(CBV)가 있다.
- `HttpRequest` : View의 첫 번째 인자, 클라이언트의 요청 정보
- `HttpResponse` : View의 반환 객체, 클라이언트로 전달되는 응답 정보

# views.py

```
from django.shortcuts import render
```

```
from polls.models import Question
```

```
def index(request):
```

```
    latest_question_list = Question.objects.all().order_by('-pub_date')[:5]
```

```
    context = {'latest_question_list': latest_question_list}
```

```
    return render(request, 'polls/index.html', context)
```

# Rendering

- 템플릿 코드를 템플릿 파일로 해석하는 과정
- 템플릿 코드 : 렌더링 전 템플릿 문법에 따라 작성된 파일
- 템플릿 파일 : 렌더링 후 결과물 파일(HTML, XML, JSON 등 텍스트 파일)
- `render()` : 템플릿 코드를 로딩한 후 `context` 변수를 적용하여 템플릿 파일로 만든 결과를 `HttpResponse` 객체에 담아 반환하는 단축함수
- 단축함수 : 웹 프로그램 개발 시 자주 사용하는 기능 또는 공통적으로 사용하는 기능을 장고에서 개발하여 제공하는 내장함수

# Template Loader

- TEMPLATES 항목에 정의된 디렉터리를 찾는다.
- INSTALLED\_APPS 항목에 등록된 각 앱의 templates 디렉터리를 찾는다.
- 애플리케이션 이름은 대부분 중복되지 않지만 템플릿 파일 이름은 중복될 수 있다. 따라서 templates 디렉터리 하위에 애플리케이션 이름으로 디렉터리를 만들고 템플릿 파일을 위치한다.

article/templates/article/post\_list.html

shop/templates/shop/post\_list.html

render(request, 'article/post\_list.html')

render(request, 'shop/post\_list.html')

# views.py

```
from django.shortcuts import get_object_or_404, render
```

```
def detail(request, question_id):
```

```
    question = get_object_or_404(Question, pk=question_id)
```

```
    return render(request, 'polls/detail.html', {'question':question})
```



# get\_object\_or\_404

- 첫 번째 인자는 모델 클래스
- 두 번째 인자부터는 검색 조건
- `question = get_object_or_404(Question, pk=question_id)`
- Question 모델 클래스로부터 `pk=question_id` 검색 조건에 맞는 객체를 조회한다.
- 조건에 맞는 객체가 없으면 `Http404` 익셉션을 발생시킨다.